تكنولوجيا المعلومات

INFORMATION TECHNOLOGY

كتاب الطالب

11

المسار التكنولوجي

الفصل الدراسي الأول
2020–2021

الطبعة الأولى

binarylogic

binarylogic

تكنولوجيا المعلومات المستوى الحادي عشر

المسار التكنولوجي

كتاب الطالب / الفصل الدراسي الأول 2021 - 2020

ISBN: 978-618-05-4672-9

PUBLISHED BY **MM PUBLICATIONS**

تكنولوجيا المعلومات

INFORMATION TECHNOLOGY

كتاب الطالب

الاسم .....

الشعبة .....

حضرة صاحب السمو الشيخ تميم بن حمد آل ثاني
أمير دولة قطر

النشيد الوطني

قَسَمًا بِمَنْ رَفَعَ السَّمَاءْ قَسَمًا بِمَنْ نَشَرَ الضِّيَاءْ
قَطَرٌ سَتَبْقَى حُرَّةً تَسْمُو بِرُوحِ الأَوْفِيَاءْ
سِيرُوا عَلَى نَهْجِ الأُلَى وَعَلَى ضِيَاءِ الأَنْبِيَاءْ
قَطَرٌ بِقَلْبِي سِيرَةٌ عِزٌّ وَأَمْجَادُ الإِبَاءْ
قَطَرُ الرِّجَالِ الأَوَّلِين حُمَاتُنَا يَوْمَ النِّدَاءْ
وَحَمَائِمٌ يَوْمَ السَّلاَمْ جَوَارِحٌ يَوْمَ الفِدَاءْ

مفاتيح رموز الكتاب

برامج أخرى:
قسم في نهاية الوحدة يعرض بعض الأدوات والبرامج البديلة.

المصطلحات:
قسم يوضح ما تعلمته والمفردات الجديدة التي يحتويها الدرس.

مشروع الوحدة:
نشاط في نهاية كل وحدة يدمج المهارات التي يتم تدريسها في الوحدة.

ماذا تعلمت:
قسم يركز على النقاط المهمة التي يحتاج الطلاب إلى مراجعتها.

تمرين عملي

تمرين نظري

نصيحة ذكية:
معلومات مفيدة.

كن آمنًا:
معلومات لحماية نفسك.

لمحة تاريخية:
أحداث حقيقية في الماضي.

أهلا بك!

تعال معي لنستكشف عالم تكنولوجيا المعلومات

انتقل إلى حاسوبك واتبعني!

وزارة التعليم والتعليم العالي
إدارة المناهج الدراسية ومصادر التعلم

المراجعة والتدقيق
فِرَق من:
كلية الهندسة - جامعة قطر
إدارة التوجيه التربوي
الميدان التربوي

الإشراف العلمي والتربوي
إدارة المناهج الدراسية ومصادر التعلم
قسم المواد الدراسية

1. تطوير مواقع الويب I		6
إنشاء موقع ويب		10
هيكلة المحتوى		26
تطبيق قواعد CSS		60
تصميم موقع الويب		81
2. تطوير مواقع الويب II		112
التصميم المتجاوب		116
موقع الويب التفاعلي		132
المعالجة المعتمدة على الخادم		152

الكفايات الأساسية للمنهج التعليمي الوطني لدولة قطر

- التفكير الإبداعي والتفكير الناقد
- التعاون والمشاركة
- الكفاية اللغوية
- التقصي والبحث
- الكفاية العددية
- حل المشكلات
- التواصل

1. تطوير مواقع الويب I

في هذه الوحدة سنستخدم أدوات تطوير الويب لإنشاء موقع ويب وسنتعلم كيفية استخدام علامات HTML و CSS لتنسيق صفحة الويب. أخيرًا، سنتعلم كيفية تصميم نموذج HTML وإنشاء شريط تنقل على صفحة الويب.

ماذا سنتعلمْ؟

في هذه الوحدة سنتعلم:

> مراحل إنشاء موقع الويب.

> استخدام محرر Visual Studio Code.

> إضافة وسوم HTML والفقرات والصور إلى موقع الويب.

> كيفية إنشاء القوائم والارتباطات التشعبية.

> إنشاء نموذج (form).

> تقسيم صفحة الويب إلى عدة أجزاء.

> استخدام قواعد CSS (أوراق الأنماط المتتالية) في تصميم النصوص والصور في موقع الويب.

> إنشاء وتصميم شريط تصفح Navigation Bar.

مواضيع الوحدة

> إنشاء موقع ويب

> هيكلة المحتوى

> تطبيق قواعد CSS

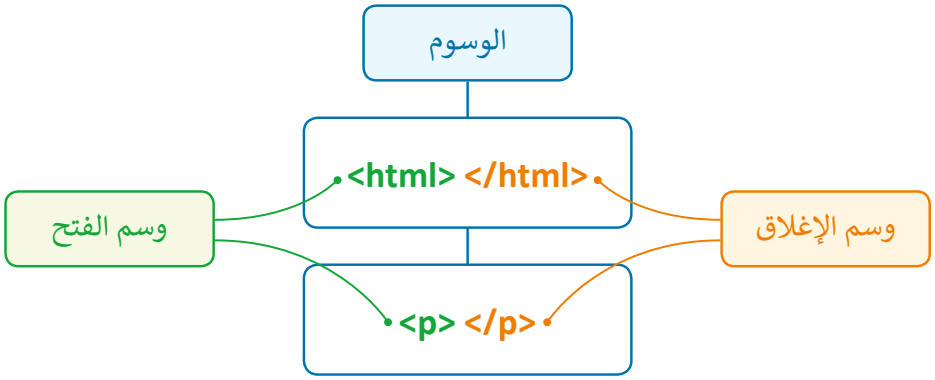
> تصميم موقع الويب

الأدوات

> Visual Studio Code

هل تذكر؟

تعني HTML (Hypertext Markup Language) لغة ترميز النص التشعبي وهي الأكثر استخدامًا في إنشاء صفحات الويب.



يجب أن يكون للصفحة بنية وهيكلية، تحتوى على مجموعة من العناصر، أهمها:

1. العنوان Title، باستخدام الوسمين <title> </title>
2. محتوى الصفحة داخل الوسمين <body> </body>
3. فقرات لتنظيم المحتوى، توضع كل منها داخل الوسمين

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Qatar page</title>
  </head>
  <body>
    <h1> Welcome to Qatar! </h1>
    <p>Doha is the capital of Qatar.</p>
  </body>
</html>
```

يتم تعريف الصور في HTML بالوسم <img> و الذي يحتوي فقط على سمات دون الحاجة إلى وسمٍ للإغلاق.

```
<p align="center"><img src="healthy_app.jpg" height="200" width="100"> </p>
<hr>
<p align="right">MIT App Inventor استخدمنا Healthy Food تطبيق لإنشاء </p>
<p align="right"> <u></u> <a href="http://www.appinventor.mit.edu/explore/"
```

كيفية إضافة لون للخلفية:

<body style="background-color:#FFEBD8;">

لكل لون رمز خاص به

	#FFFFFF
	#000000
	#FF0000
	#1D761B

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title> تطبيق الغذاء الصحي </title>
  </head>
  <body style="background-color:#FFEBD8">
```

لمحاذاة الفقرة، نستخدم خاصية Align مع الوسم <P align="....." كما في الأمثلة الآتية:

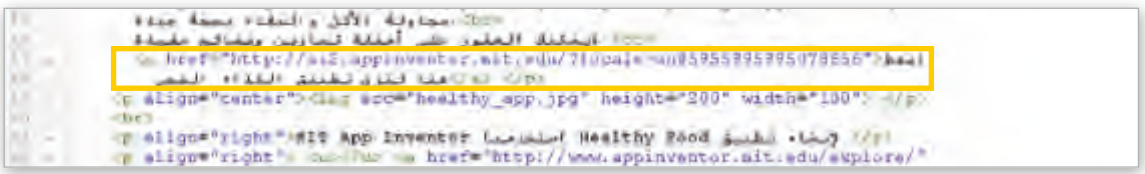
يتم محاذاة هذه الفقرة إلى اليمين

 يتم محاذاة هذه الفقرة إلى اليسار

 يتم محاذاة هذه الفقرة إلى الوسط

يطلق على روابط HTML الارتباطات التشعبية، ويمكن الملاحظة أنه عند تحريك الفأرة فوق رابط، سيتحول شكل مؤشر الفأرة إلى شكل يد صغيرة.

يتم تمييز الروابط في HTML باستخدام الوسمين <a> .



الدرس الأول

إنشاء موقع ويب

إنشاء موقع ويب إلكتروني

كما تعرفنا في السابق فإن موقع الويب هو مجموعة من الصفحات الإلكترونية المترابطة ذات المحتوى المشترك، والتي يتم تخزينها بخوادم إنترنت **Servers** ويمكن الوصول إليها من خلال حاسوب العميل **Client**.

بعض محتويات صفحة الويب:

- النصوص.
- ملفات وسائط متعددة وتشمل:
  - الصور.
  - ملفات فيديو.
  - ملفات صوتية.
- روابط إلى صفحات أو مواقع ويب أخرى.

يجب اتباع خطواتٍ محددة وعملية معينة عند إنشاء موقع ويب. تتكون هذه العملية من خطوات متسلسلة تبدأ من تصميم الموقع الإلكتروني وصولًا إلى بنائه ونشره عبر شبكة الإنترنت.

مراحل إنشاء موقع الويب:

1 **التخطيط:**

يجب تحديد الهدف والغرض من الموقع ورسم هيكليته العامة و ذلك قبل البدء بتصميم الموقع.

2 **التصميم:**

بعد تكوين صورة واضحة عن الموقع ومحتوياته، نقوم برسم المخطط العام للصفحة الرئيسة والصفحات الفرعية للموقع على الورق.

3 **التنفيذ:**

بعد تصميم المخطط العام، تأتي عملية تصميم الصفحات وتطوير المحتوى باستخدام أحد برامج تصميم مواقع الويب، مثل Visual Studio Code.

4 **اختبار الموقع ونشره عبر شبكة الإنترنت:**

في هذه المرحلة تتم عملية اختبار كل صفحة بعناية والتحقق من عمل جميع الروابط بين الصفحات بشكل صحيح ثم نشر الموقع على شبكة الإنترنت.

التخطيط

يجب أولًا تحديد الهدف النهائي من إنشائنا لموقع الويب الخاص بنا.

الهدف من مشروعنا هو تصميم و إنشاء موقع الويب لمشجعي منتخب قطر الوطني لكرة القدم. سنقدم في هذا الموقع بعض المعلومات عن المنتخب وإنجازاته، وسنمكّن زوار الموقع من ترك رسالة باستخدام نموذج تواصل.

قبل البدء بتصميم موقع الويب علينا أن نتعرف مبادئ التصميم التي ستساعدنا في الحصول على النتيجة المرجوة كما خططنا لها.

الخصائص التي ينبغي توافرها في موقع الويب:

- محتوى واضح.
- صور و رسومات.
- نسق لوني مناسب.
- تنسيق مناسب للعرض على الهواتف النقالة.

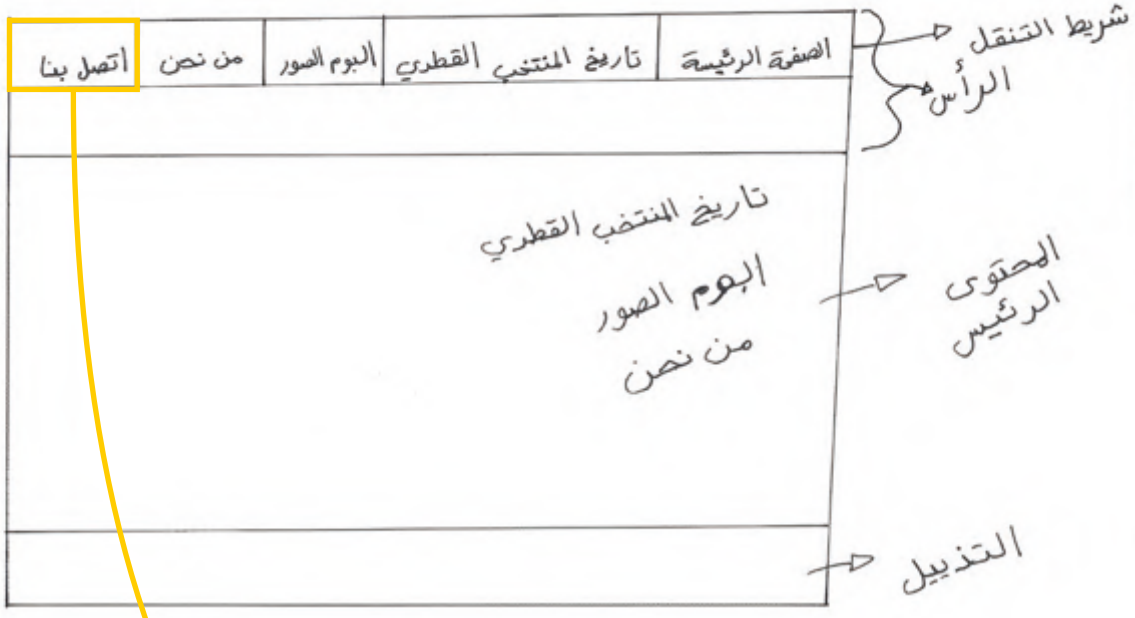
التصميم

في هذه المرحلة سنضع تصميمًا لمخطط صفحة الويب، والتي ستحتوي على ثلاثة أجزاء مختلفة.

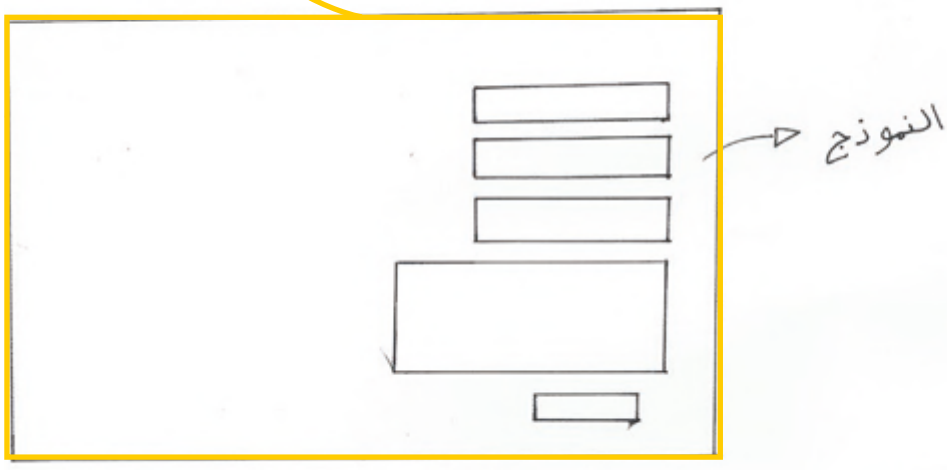
الأجزاء الثلاثة ستكون:

- رأس الصفحة "`Header`". وهذا يشمل ترويسة رسومية وشريط التنقل.
- جسم الصفحة "`Body`". ويشمل المحتوى المتمثل في النصوص والصور وغيرها ...
- التذييل "`Footer`". ويحتوي على روابط مفيدة.

الصفحة الرئيسة



صفحة "اتصل بنا"



التنفيذ

الآن وبعد أن صممنا تخطيط الموقع، سنبدأ بإنشائه وذلك باستخدام محرر **Visual Studio Code**.

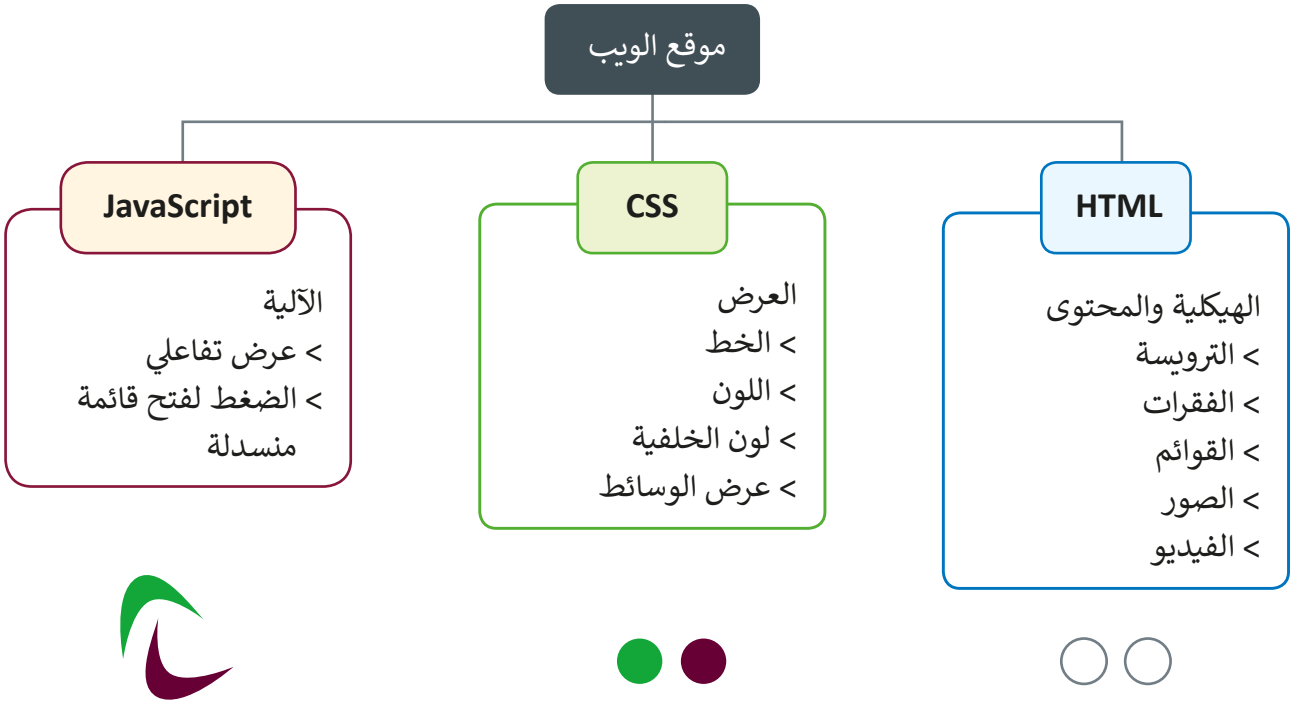
سنقوم أولًا بتجهيز بُنية الموقع باستخدام لغة HTML وإضافة المحتوى، وفيما بعد سنطبق نمط تنسيق مخصص سيمنح موقعنا جاذبية وتفاعلًا باستخدام CSS، وأخيرًا سنستخدم Javascript من أجل إنشاء واجهات تفاعلية ستحسن من تجربة المستخدم والوظيفة المطلوبة من الموقع.

- سيتم استخدام **HTML**، و **CSS**، و **Javascript** لإنشاء الموقع.

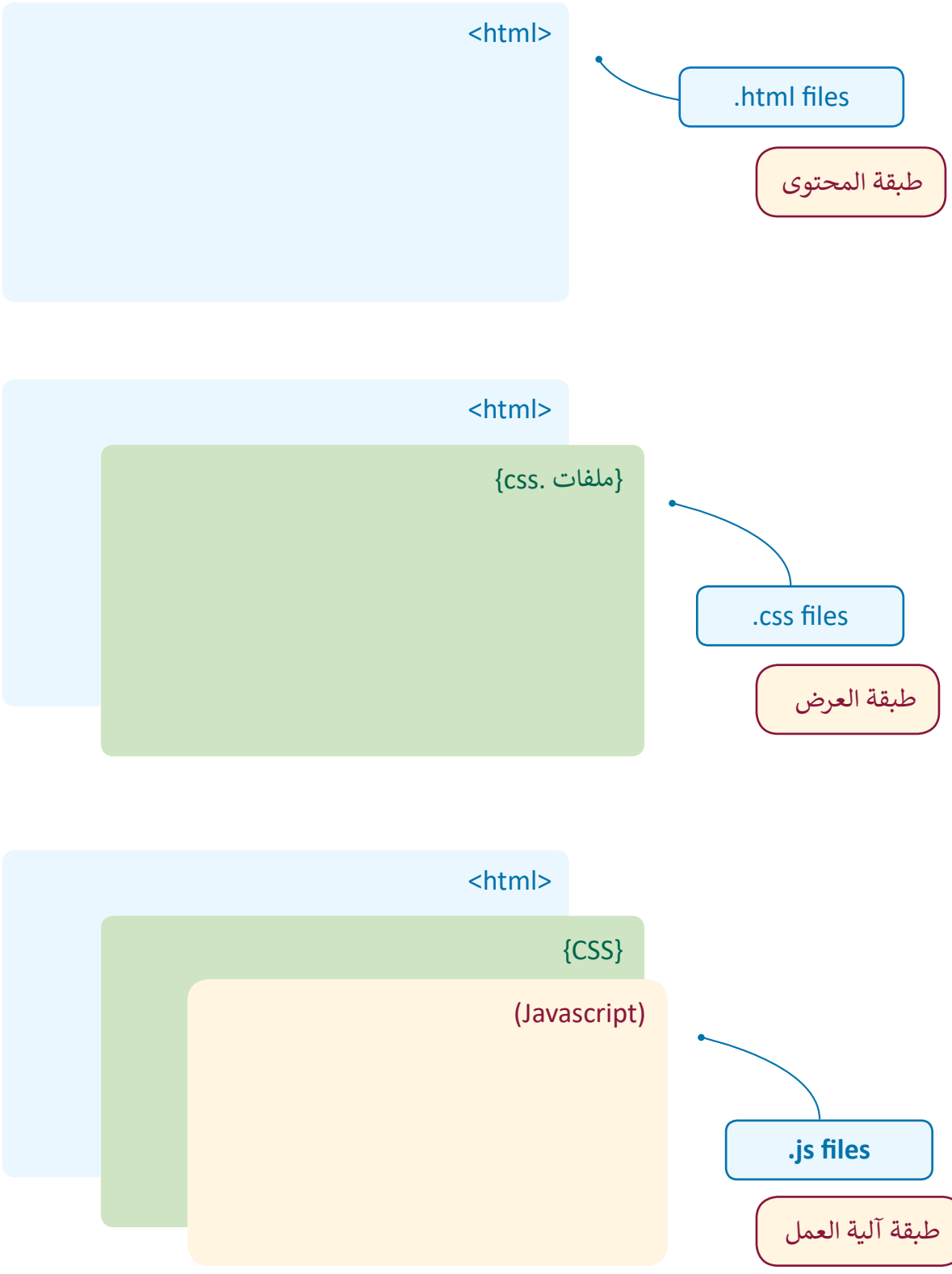
HTML: اختصار لـ Hyper Text Markup Language (لغة ترميز النص التشعبي)، و تستخدم لإعداد الهيكلية العامة للموقع، و يُمكن اعتبارها العمود الفقري لمواقع الويب.

CSS: Cascading Style Sheets (أوراق التنسيق المتتالية): تُحسِّن من تخطيط موقع الويب لعرضه بطريقةٍ أفضل.

JS: Javascript هي لغة برمجة نصية تستخدم لإضافة محتوىً تفاعلي للموقع الإلكتروني، لتحسين وظائف الموقع والتحكم في عمل العناصر الأخرى.



يفضل العمل باللغات الثلاث المستخدمة في موقع الويب بشكلٍ منفصل، وحفظ المقطع البرمجي على شكل ملفاتٍ مستقلة، ثم الدمج بين هذه اللغات باستخدام ربط الصفحات في **HTML** مع **CSS** و **JavaScript**. إن كل لغة برمجة تعتبر طبقةً منفصلة في الموقع تؤدي غرضًا مختلفًا.

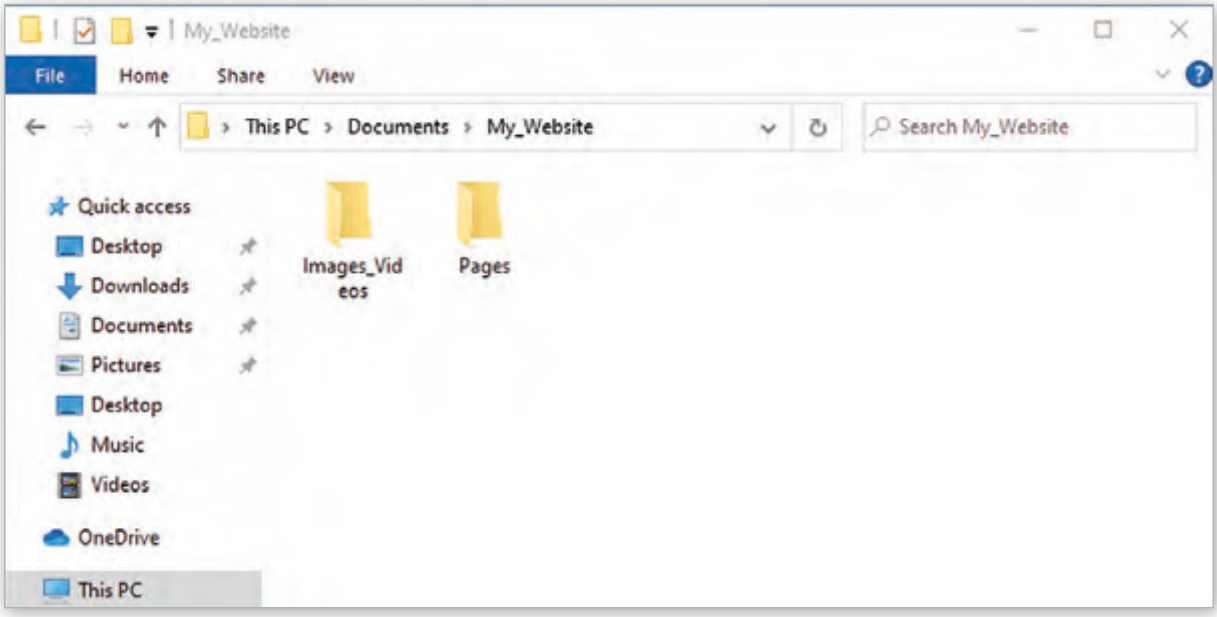


بدء إنشاء موقع الويب

لإنشاء موقع الويب يجب اتباع الخطوات التالية:

أولاً: إنشاء مجلد رئيس في حاسوبنا، وبداخل هذا المجلد سننشئ مجلدات فرعية منها مجلد **Images_Videos**: (الصور والفيديو) حيث سيتم حفظ صور الموقع داخل هذا المجلد، ومجلد **Pages** حيث سيتم حفظ صفحات الويب التي سنقوم بإنشائها.

يفضل أن يكون اسم المجلد الرئيس والمجلدات الفرعية باللغة الانجليزية، تماشيًا مع اللغة المستخدمة في البرمجة.

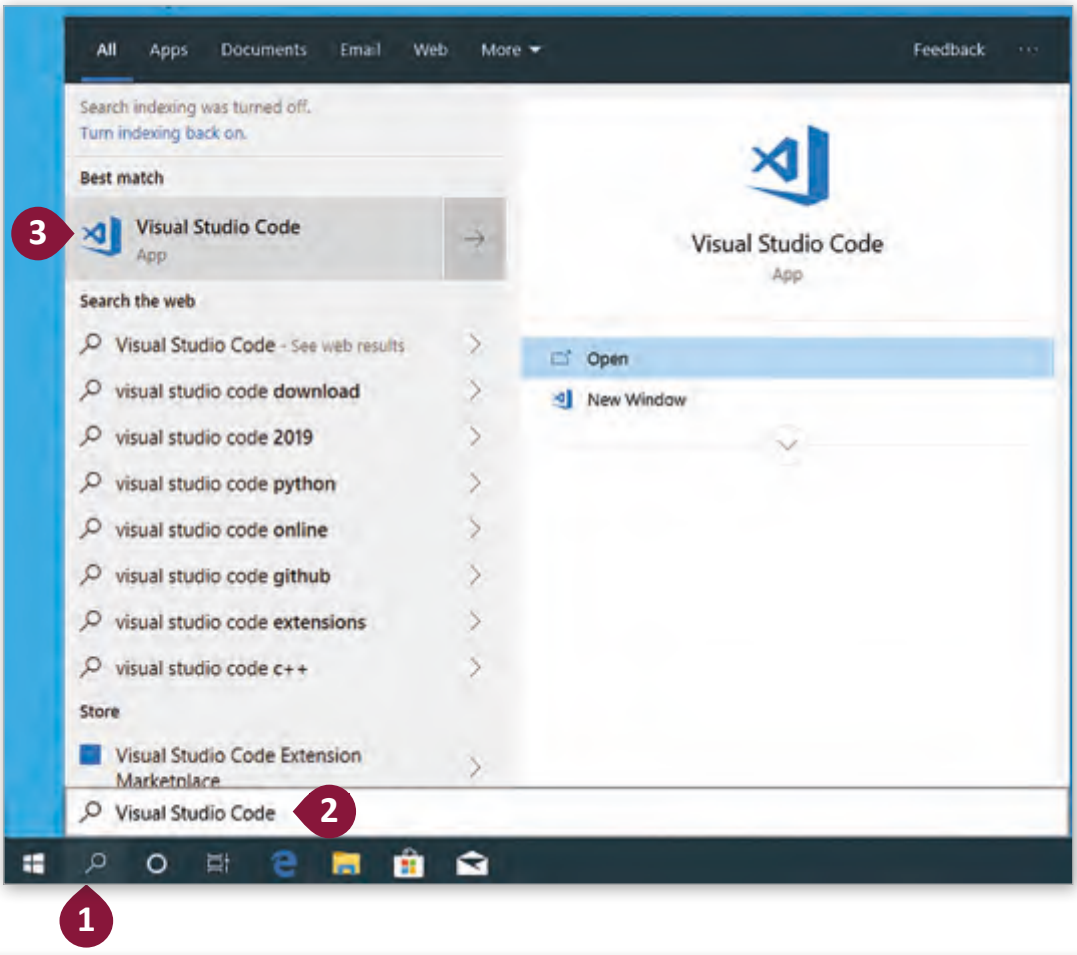


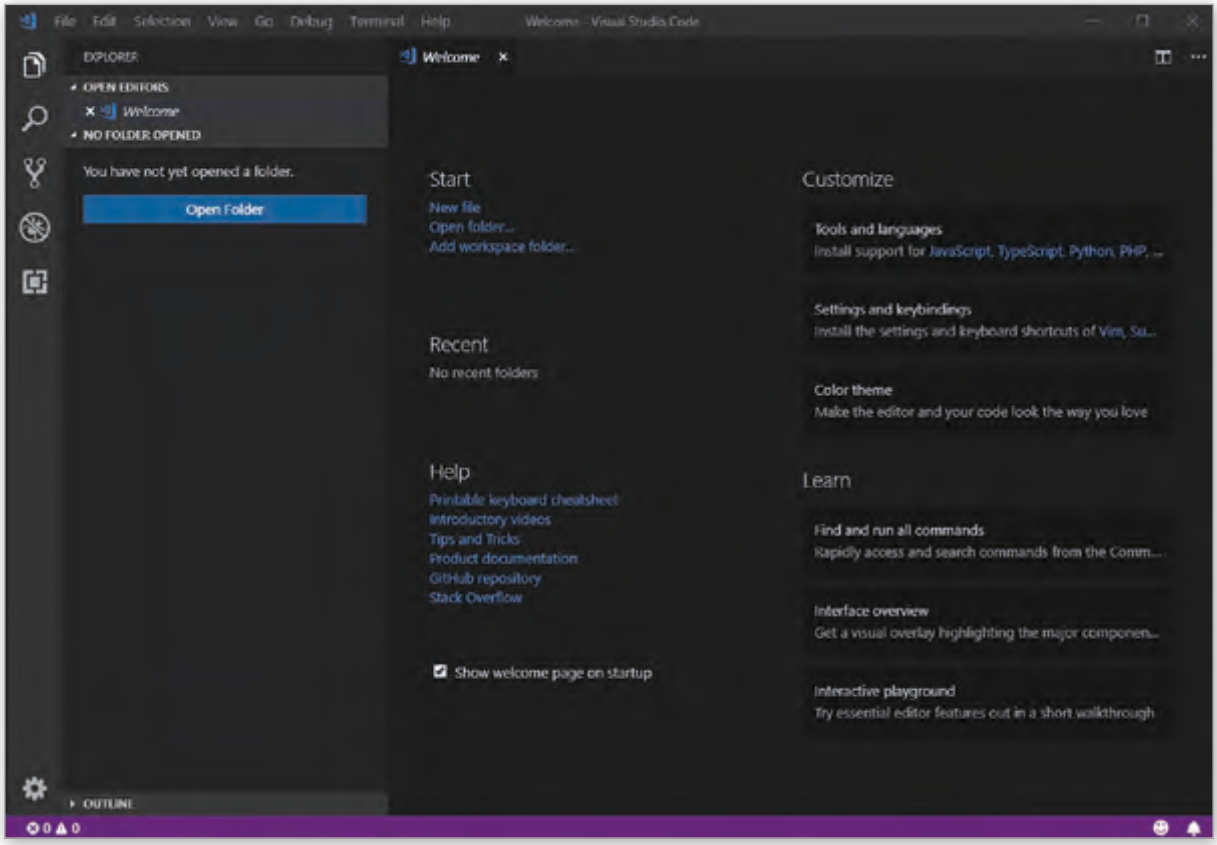
ثانيًا: نقوم بفتح محرر **Visual Studio Code** وبدء إنشاء ملف **HTML** الخاص بالموقع.

لفتح محرر Visual Studio Code:

> اضغط زر البحث. ❶

> في شريط البحث اكتب **Visual Studio Code** ❷ اضغط لفتح البرنامج. ❸


All
Apps
Documents
Email
Web
More
Feedback
Search indexing was turned off.
Turn indexing back on.
Best match
Visual Studio Code
App
Search the web
Visual Studio Code - See web results
visual studio code download
visual studio code 2019
visual studio code python
visual studio code online
visual studio code github
visual studio code extensions
visual studio code c++
Store
Visual Studio Code Extension Marketplace
Visual Studio Code
Open
New Window
1
2
3


EXPLORER
OPEN EDITORS
Welcome
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Show welcome page on startup
Customize
Tools and languages
Settings and keybindings
Color theme
Learn
Find and run all commands
Interface overview
Interactive playground
OUTLINE

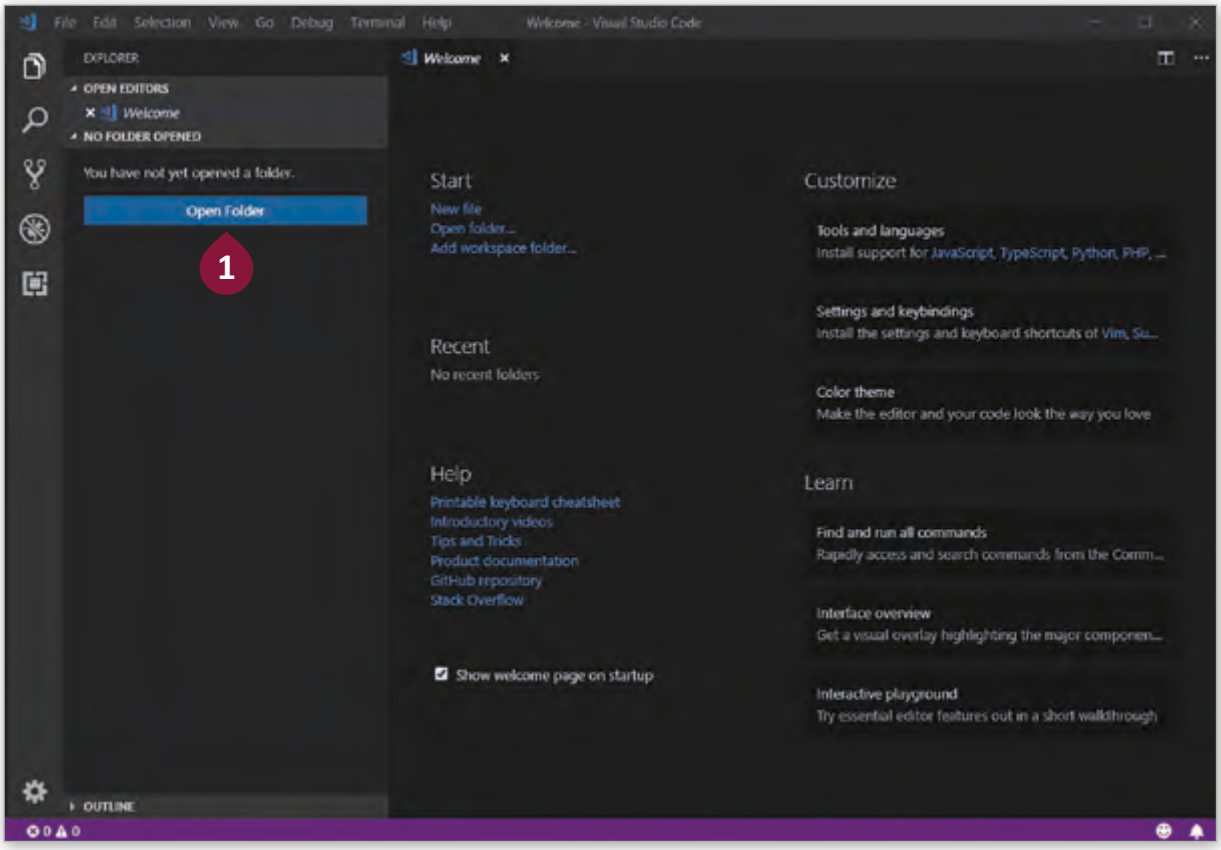
ثالثًا: سنقوم بفتح المجلد الذي أنشأناه مسبقًا.

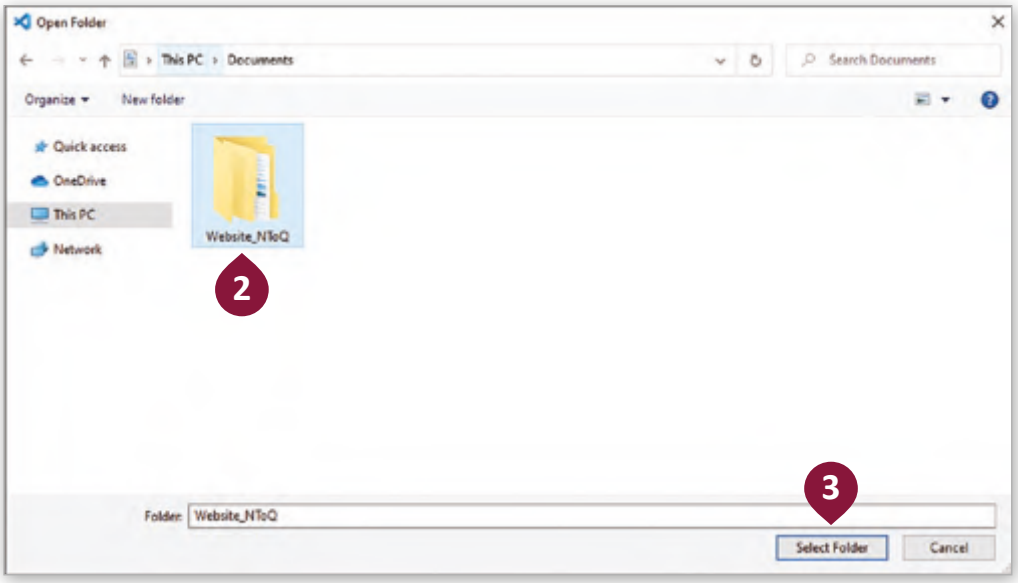
لفتح المجلد:

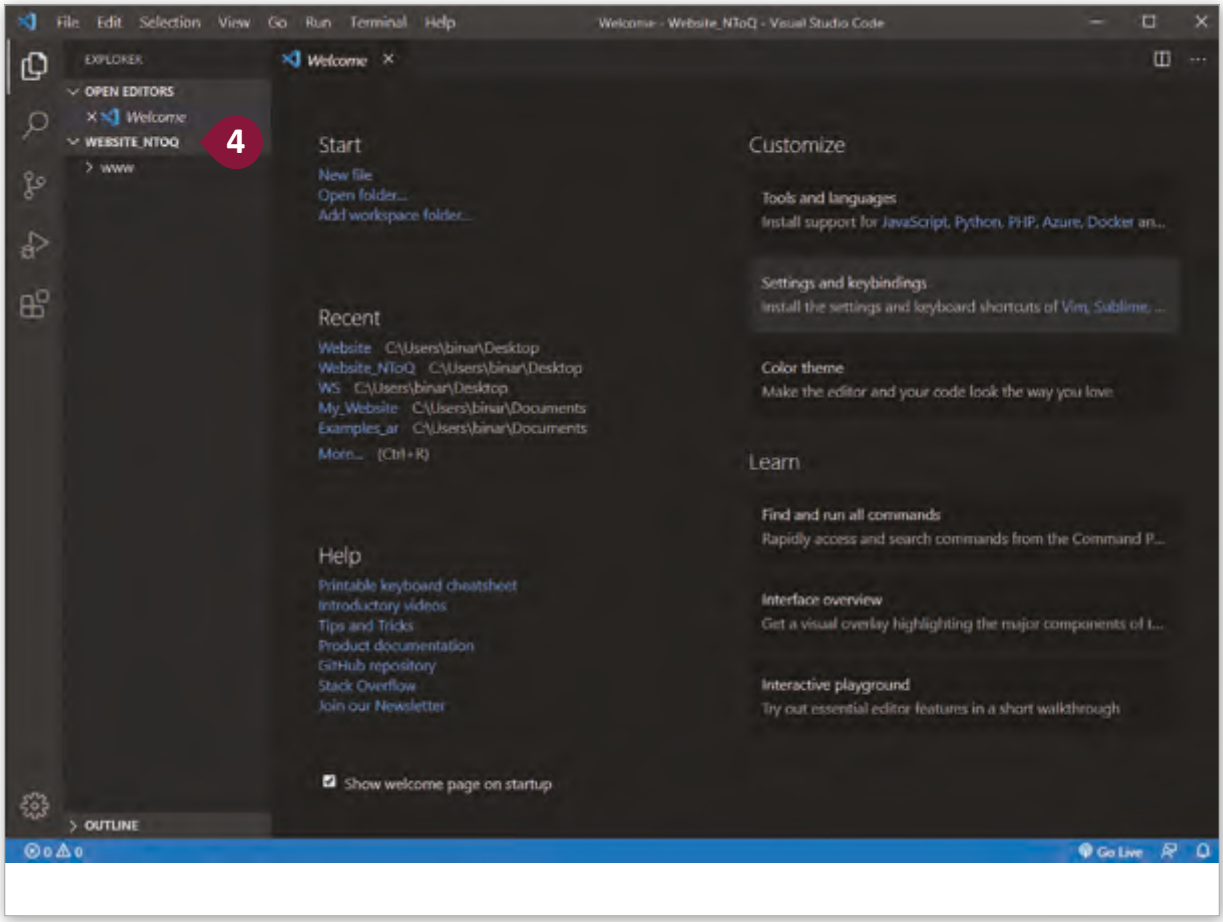
> من قسم المستكشف **Explorer**، اضغط **Open Folder** (فتح مجلد). ❶

> من نافذة فتح المجلد اختر المجلد الذي تريده ❷ ثم اضغط **Select Folder** (اختر المجلد). ❸

> سيظهر مجلد العمل الخاص بك والمجلدات الفرعية في الشريط الجانبي. ❹




Open Folder
This PC › Documents
Search Documents
Organize
New folder
Quick access
OneDrive
This PC
Network
Website_NToQ
2
3
Folder: Website_NToQ
Select Folder
Cancel


File Edit Selection View Go Run Terminal Help
Welcome - Website_NToQ - Visual Studio Code
EXPLORER
OPEN EDITORS
Welcome
WEBSITE_NTOQ
www
4
Welcome
Start
New file
Open folder...
Add workspace folder...
Recent
Website C:\Users\binar\Desktop
Website_NToQ C:\Users\binar\Desktop
WS C:\Users\binar\Desktop
My_Website C:\Users\binar\Documents
Examples_ar C:\Users\binar\Documents
More... (Ctrl+R)
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Join our Newsletter
Customize
Tools and languages
Install support for JavaScript, Python, PHP, Azure, Docker an...
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, ...
Color theme
Make the editor and your code look the way you love
Learn
Find and run all commands
Rapidly access and search commands from the Command P...
Interface overview
Get a visual overlay highlighting the major components of t...
Interactive playground
Try out essential editor features in a short walkthrough
Show welcome page on startup
OUTLINE
Go Live

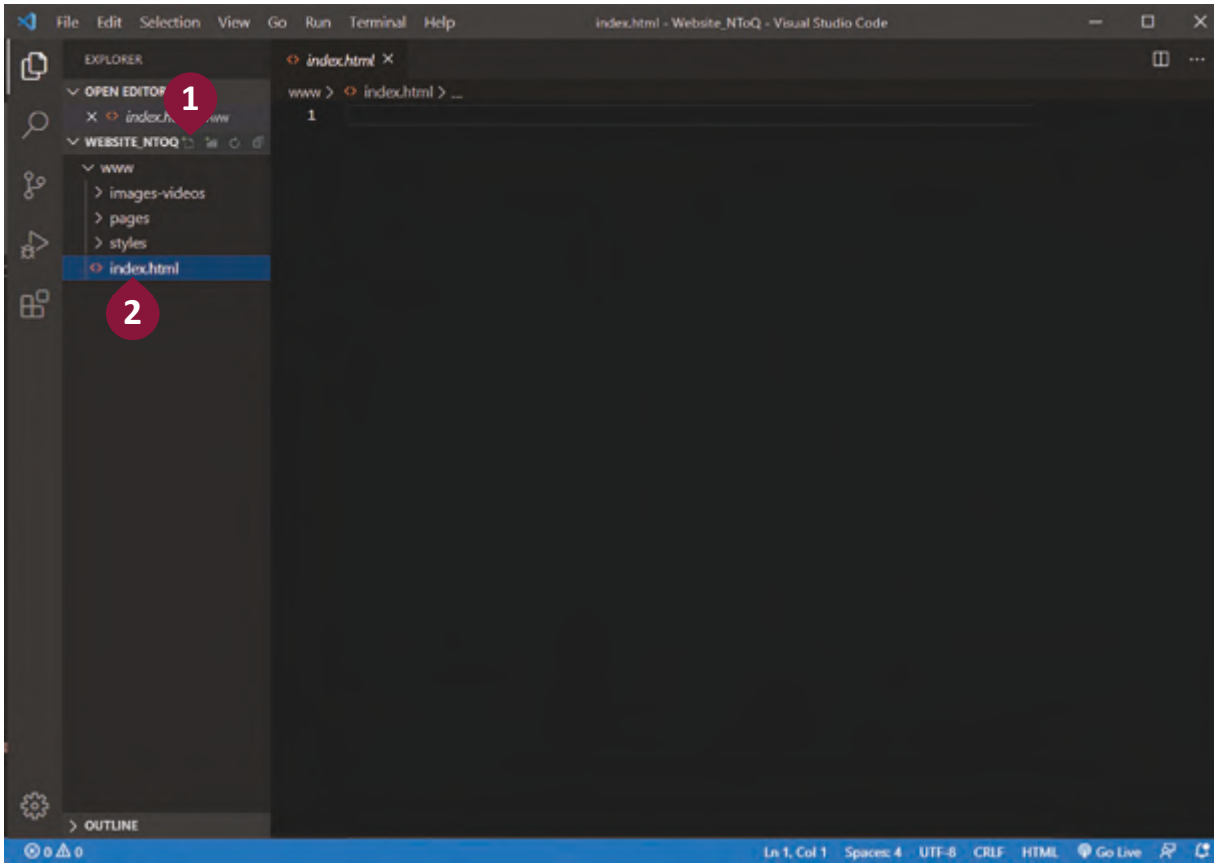
رابعًا: لإعداد بُنية الموقع يجب أن ننشئ ملف **HTML**.

لإنشاء ملف HTML:

> من قسم **Explorer** (المستكشف)، وفي المجلد الذي قمت بفتحه اضغط زر **New File** (ملف جديد). ❶

> اكتب الاسم بالامتداد **html**. ❷

> يمكنك الآن إضافة المحتوى للمستند في محرر لغة **HTML**. ❸

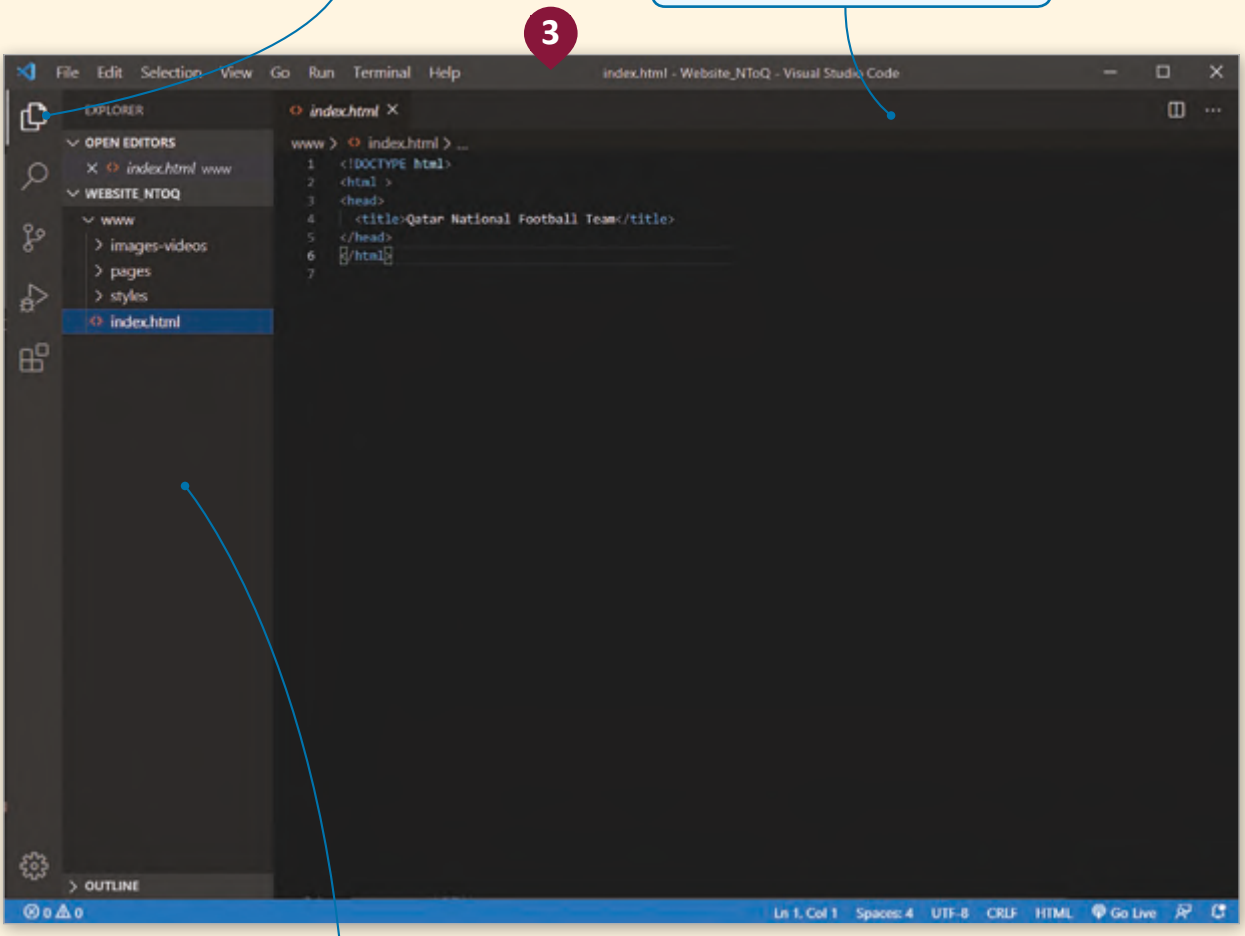


نصيحة ذكية

لا تنسَ أن تحفظ مستندك بعد كل عملية تعديل تقوم بها. يمكنك أيضًا استخدام اختصارات لوحة المفاتيح Ctrl + S.

تظهر هنا الإشعارات بعدد المستندات التي لم يتم حفظها.

منطقة التحرير: وهي المساحة الرئيسة لتحرير ملفاتك. يمكنك فتح الملفات لتحريرها كما تريد.



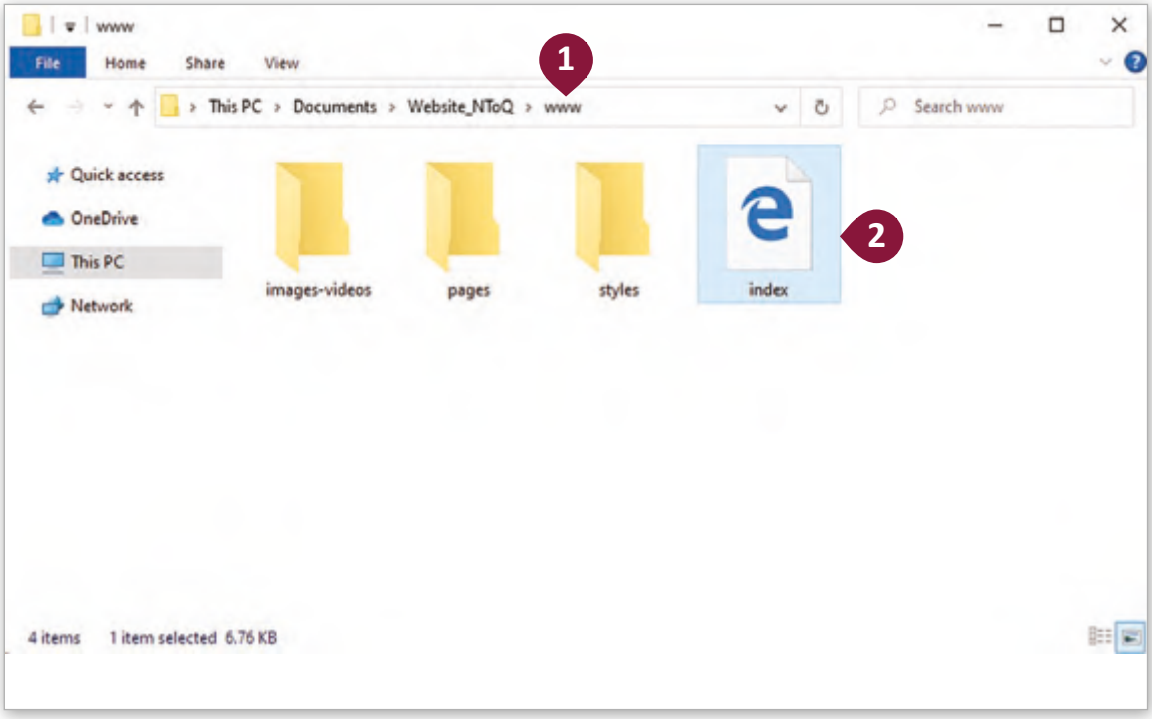
الشريط الجانبي: يحتوي على طرق عرض مختلفة مشابهة للمستكشف تساعدك أثناء العمل في مشروعك.

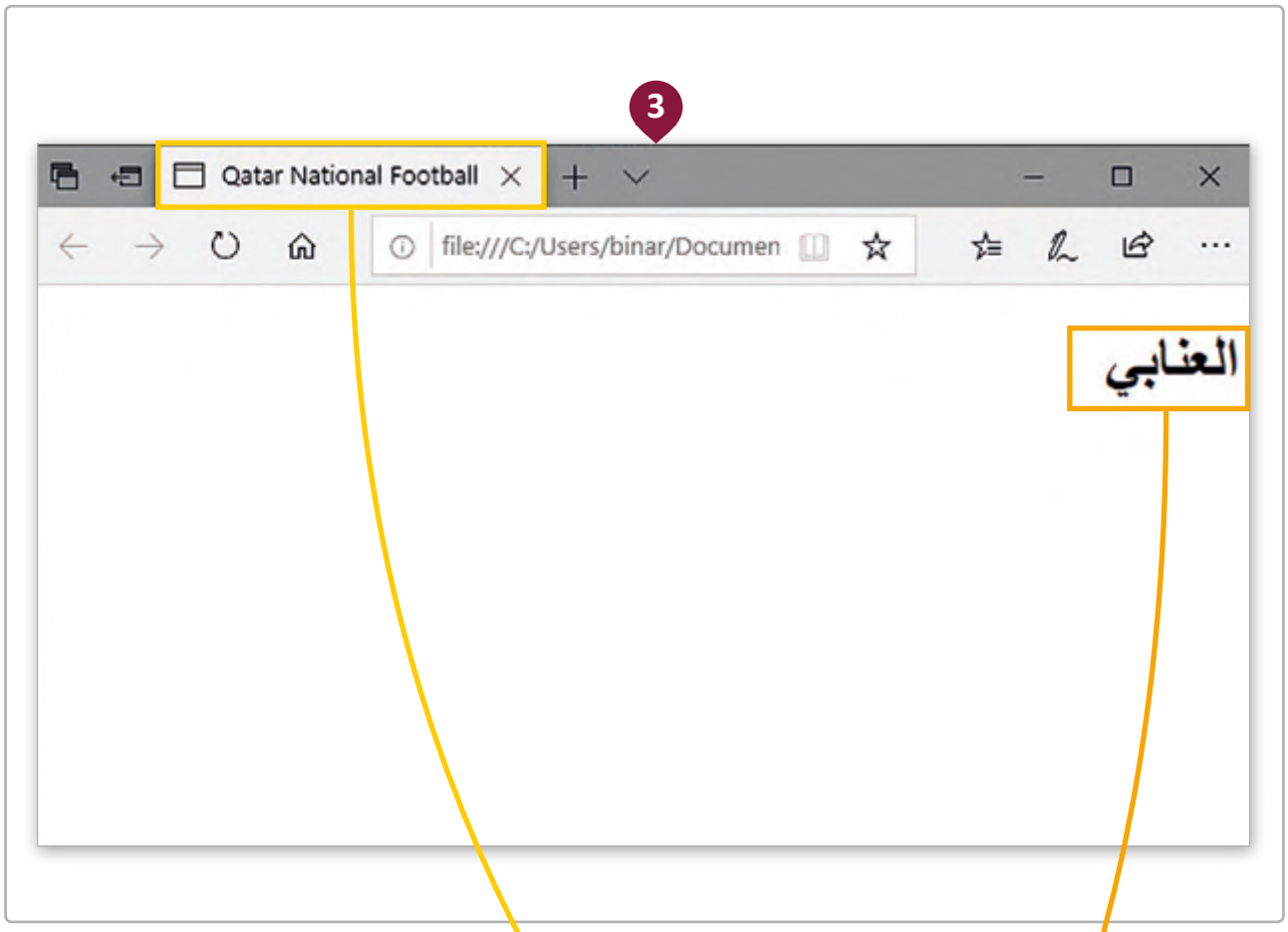
خامسًا: لعرض صفحة الويب نقوم بفتح ملف **HTML** الذي أنشأناه باستخدام المتصفح.

لفتح ملف HTML في المتصفح:

> افتح المجلد الرئيس في حاسوبك واعثر على المجلد الفرعي الذي حفظت فيه ملف **HTML** الذي أنشأته مسبقًا. ❶

> اضغط ضغطا مزدوجا على الملف ❷ سيتم فتح الملف وعرض نتيجة تنفيذه. ❸





خاصية dir تستخدم لضبط اتجاه النص الأساسي الخاص بالعرض، من الضروري ضبط اللغة في HTML لتكون من اليمين إلى اليسار (right-to-left) عند الكتابة باللغة العربية.

خاصية charset تحدد ترميز الحروف الخاص في مستند .html

1

قم بتوضيح العلاقة بين كل من HTML و CSS وJavaScript داخل صفحة موقع الويب.

2

ضع علامة ✓ أمام العبارة الصحيحة وعلامة ✗ أمام العبارة الخطأ.

.1	تستخدم HTML في إنشاء هيكلية موقع الويب.	○
.2	تمكننا CSS (أوراق التنسيق المتتالية) من إضافة المحتوى التفاعلي الخاص بالموقع لتحسين وظائفه.	○
.3	باستخدام قواعد CSS يمكن تصميم موقع الويب لجعله أكثر جاذبية.	○
.4	تستخدم أوامر JavaScript للتحكم وإدارة العناصر المختلفة في الصفحة.	○

3

خطط لإنشاء موقع ويب يتضمن معلومات سياحية للمسافرين إلى بلدك أو إلى دولة أخرى من اختيارك، بحيث يتضمن التالي:

- صورًا عالية الجودة.

- وصفًا مختصرًا للدولة مع تمييز الأماكن المهمة.

- دليل للفنون والثقافة بما فيها المتاحف والمسارح والأماكن السياحية.

- بعض المعلومات عن الفعاليات الخارجية.

1. أنشيء مخططًا توضيحيًا يساعدك في إنشاء الموقع لاحقًا.

2. أنشئ مجلدًا باسم "Adventure_website".

3. في هذا المجلد قم بإنشاء مجلدين فرعيين: الأول باسم "Pages" والثاني باسم "images"، والذي تتم فيه إضافة جميع الصور التي ستستخدمها في الموقع.

4. افتح محرر Visual Studio Code ثم قم بفتح المجلد الذي أنشأته، وداخل هذا المجلد أنشئ ملف HTML للبدء بإنشاء موقع الويب الخاص بك.

5. امنح موقعك اسمًا ثم قم بحفظ التغييرات.

هيكلة المحتوى

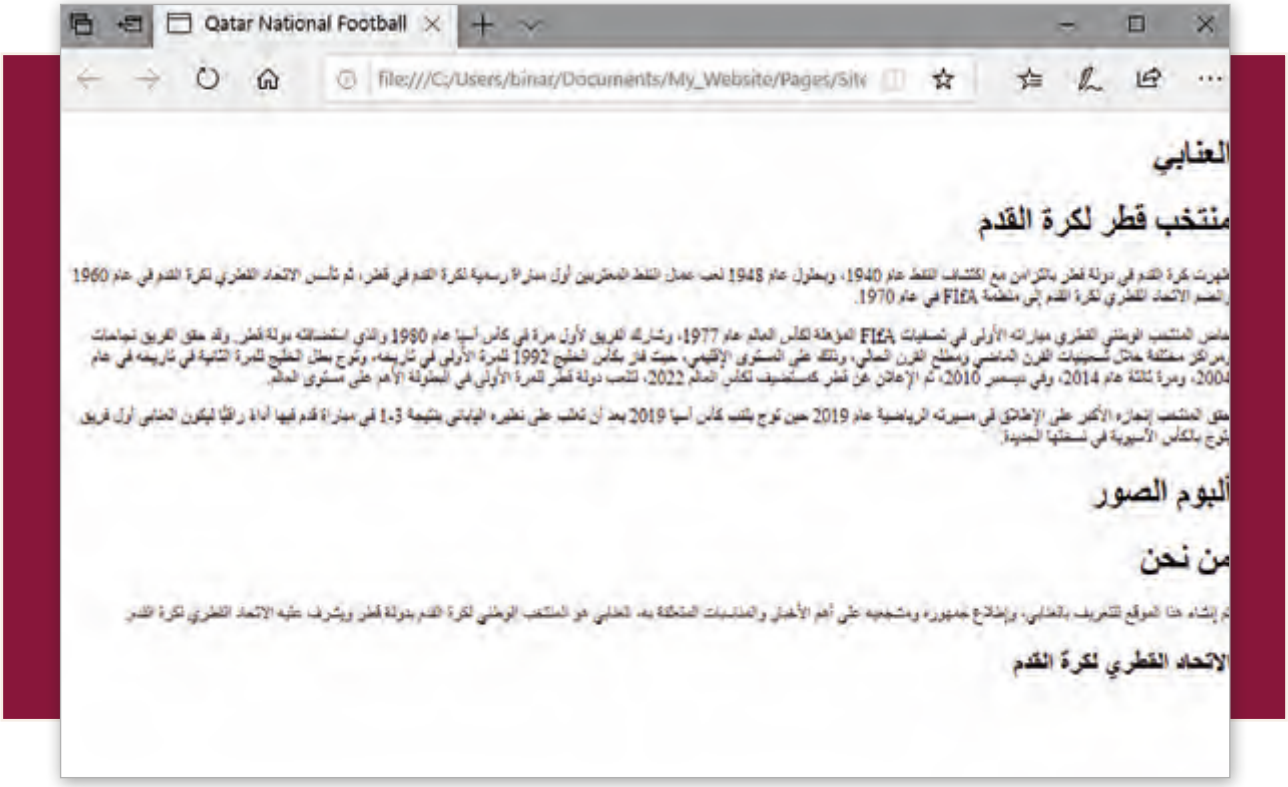
إضافة فقرة Adding a Paragraph

كما تعلمنا سابقًا فإن إحدى عناصر مستند **HTML** المهمة هي الفقرة، والتي يتم تعريفها داخل وسم <p>.

```
<p></p>
<p></p>
<p></p>
```

تأتي أهمية استخدام الفقرات أنها تتيح فرصة تقسيم النص الخاص بنا إلى أجزاء أصغر، وهذا سيسهل على زوار الموقع مهمة تصفحه وقراءة نصوصه بسهولة...

في مستندنا سنضيف فقرات متعلقة بمنتخب قطر لكرة القدم.



.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Qatar National Football Team</title>
   <meta charset="UTF-8" />
</head>
<h1>العنابي</h1>
<h1>منتخب قطر لكرة القدم </h1>
<p>ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، وبحلول عام 1948 لعب عمال النفط المغتربين أول مباراة رسمية لكرة القدم في قطر، ثم تأسس الاتحاد القطري لكرة القدم في عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIfA عام 1970.</p>
<p>خاض المنتخب الوطني القطري مباراته الأولى في تصفيات FIfA، 1977 المؤهلة لكأس العالم وشارك الفريق لأول مرة في كأس آسيا عام 1980 والذي استضافته دولة قطر. وقد حقق الفريق نجاحات ومراكز مختلفة خلال تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على المستوى الإقليمي، حيث فاز بكأس الخليج 1992 للمرة الأولى في تاريخه، وتوج بطل الخليج للمرة الثانية في تاريخه في عام 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر كمستضيف لكأس العالم 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى العالم.</p>
<p>حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب كأس آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداءً راقيًا ليكون العنابي أول فريق يتوج بالكأس الآسيوية في نسختها الجديدة.</p>
<h1>ألبوم الصور</h1>
<h1>من نحن</h1>
 <p>تم إنشاء هذا الموقع للتعريف بالعنابي، وإطلاع جمهوره ومشجعيه على أهم الأخبار والمناسبات المتعلقة به، العنابي هو المنتخب الوطني لكرة القدم بدولة قطر ويشرف عليه الاتحاد القطري لكرة القدم. </p>
<h2>الاتحاد القطري لكرة القدم</h2>
```

يتم استخدام الوسم <h1> لإنشاء العنوان وتمييزه بخط كبير.

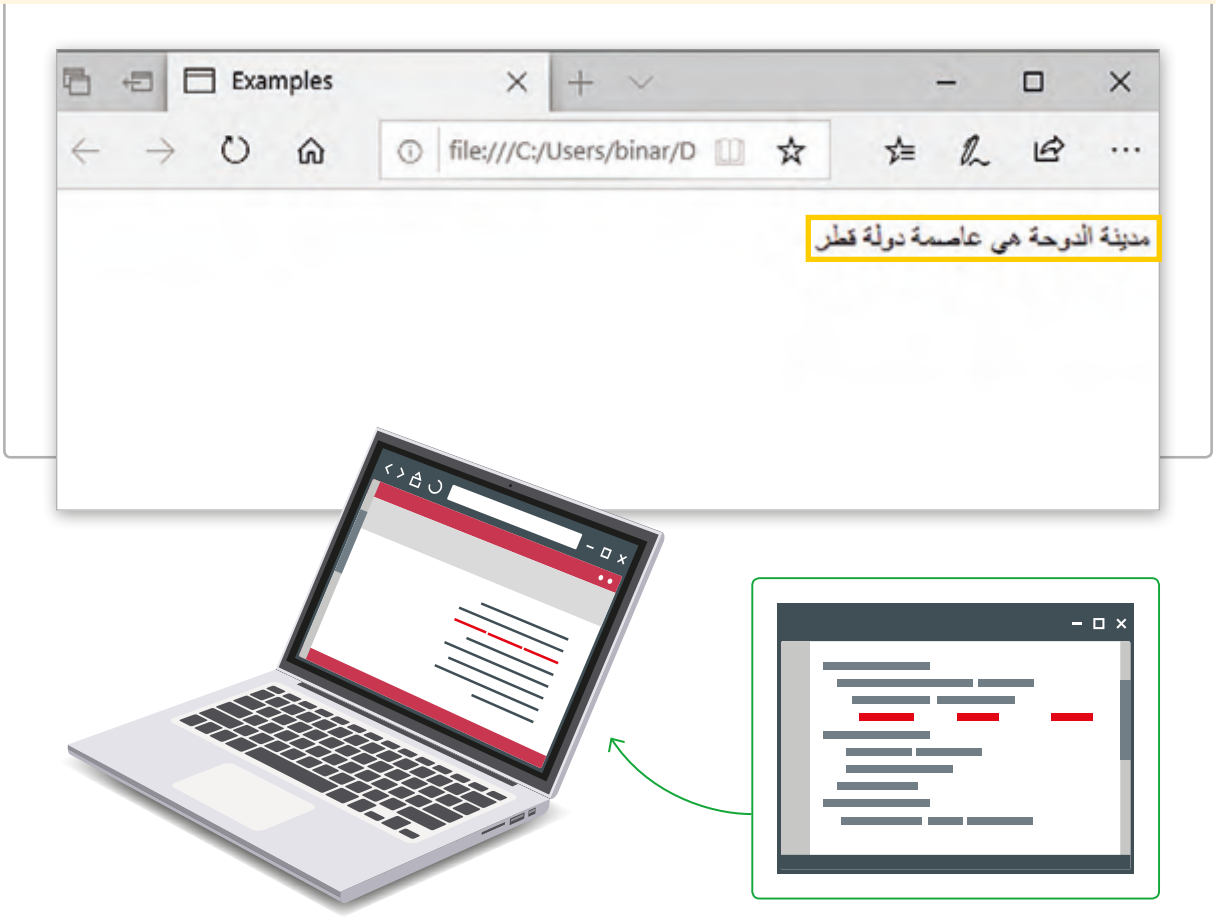
المساحة الفارغة White Space

في **HTML**، يتم دمج المساحات الفارغة معًا حيث تظهر كمساحة بيضاء واحدة. يستغل مبرمجو صفحات الويب المساحات البيضاء لجعل المقطع البرمجي أسهل للقراءة .هذا لا يؤثر على مظهر الفقرة بأي شكل من الأشكال.

على سبيل المثال، لدينا الفقرة التالية:

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8" />
</head>
<body>
   <p>مدينة الدوحة هي عاصمة دولة قطر</p>
</body>
</html>
```



القوائم Lists

يوجد نوعان من القوائم التي يتم استخدامها في مواقع الويب:

- القائمة المرتبة: في هذه القائمة يتم اتباع تسلسل رقمي بحيث يتم ترقيم كل عنصر في القائمة.
- القائمة غير المرتبة: يتم تمييز عناصر هذه القائمة بواسطة نقاط التعداد النقطي.

القائمة المرتبة Ordered List

يتم إنشاء القائمة المرتبة **Ordered List** بواسطة الوسم ، ويوضع كل عنصر **List Item** في هذه القائمة بين وسم الفتح **
** ووسم الإغلاق **
**.

لنستعرض مثالًا على ذلك.

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
</head>
<body>
   <h1>متاحف قطر</h1>
      <ol>
         <li>متحف قطر الوطني</li>
         <li>متحف الفن الإسلامي</li>
         <li>متحف: المتحف العربي للفن الحديث</li>
         <li>متحف قطر الأولمبي والرياضي</li>
         <li>متحف المستشرقين</li>
      </ol>
</body>
</html>
```

القائمة غير المرتبة Unordered List

يتم إنشاء القائمة غير المرتبة **Unordered List** بوسم ، وكل عنصر **List Item** يليه في القائمة يتم وضعه بين وسم الفتح **
** ووسم الإغلاق **.**

لنستعرض مثالًا على ذلك.

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
</head>
<body>
   <h1>متاحف قطر</h1>
   <ul>
      <li>متحف قطر الوطني</li>
      <li>متحف الفن الإسلامي</li>
      <li>متحف: المتحف العربي للفن الحديث</li>
      <li>متحف قطر الأولمبي والرياضي</li>
      <li>متحف المستشرقين</li>
   </ul>
</body>
</html>
```

لنضيف الآن القوائم الخاصة بالمنتخب الوطني في مشروعنا.

في البداية سننشئ قائمة بمثابة شريط تصفح الموقع، وستتضمن العناصر التالية:

- Home < (الصفحة الرئيسة)
- History < (تاريخ المنتخب القطري)
- Gallery < (ألبوم الصور)
- About < (من نحن)
- Contact Us < (اتصل بنا)

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
</head >
<body>
   <ul>
      <li>الصفحة الرئيسة</li>
      <li> تاريخ المنتخب القطري</li>
      <li> ألبوم الصور</li>
      <li>من نحن</li>
      <li>اتصل بنا</li>
   </ul>
</body>
</html>
```

الروابط Hyperlinks

من المفيد جدًا استخدام الروابط في موقع الويب فهي تسمح لنا بالتنقل من صفحة ويب إلى أخرى.

من الأمثلة على الروابط:

- روابط من صفحة إلى صفحةٍ أخرى في نفس الموقع.
- روابط من جزء في صفحة الويب إلى جزء آخر في نفس الصفحة.
- روابط من موقع ويب إلى آخر.
- روابط تفتح في نافذة متصفح جديدة.
- روابط تبدأ تطبيق البريد الإلكتروني الخاص بك، لإنشاء رسالة بريد الكتروني جديدة.

يتم إنشاء الروابط باستخدام الوسم <a>. كل ما يقع بين وسم الفتح <a> ووسم الإغلاق يصبح قابلًا للضغط عليه، باستخدام خاصية href يمكننا تحديد عنوان الهدف (الصفحة التي ستفتح عند الضغط على الرابط).

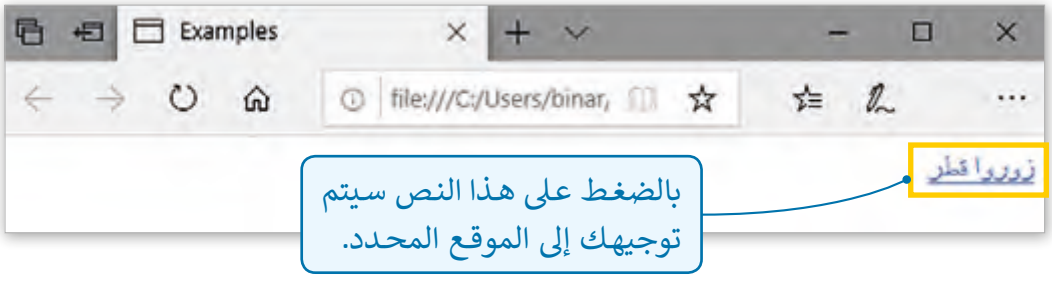
```
<a href="https://www.visitqatar.qa/">
```

هذا عبارة عن رابط لموقع ويب (URL). إن قيمة خاصية href هنا هي عبارة عن اسم لموقع ويب كامل، تمامًا كالعنوان الذي تقوم بكتابته في المتصفح لزيارة تلك الصفحة.

لنستعرض مثالًا على الروابط إلى مواقع ويب أخرى.

.html

```
<!DOCType html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
</head>
<body>
   <a href="https://www.visitqatar.qa/ar">زوروا قطر</a>
</body>
</html>
```



نصيحة ذكية

قم بإنشاء الروابط إلى مواقع الويب الموثوقة. يجب علينا مشاركة محتوى الآخرين من خلال موقعنا، لكي يتم الربط إلى موقعنا من قبل الآخرين.

حالة الرابط

لربما لاحظت عند تصفح الإنترنت أن مؤشر الفأرة يتغير عند مروره فوق رابط. يتمثل هذا التغيير في ظهور يد صغيرة وتغير لون الرابط.

تظهر الروابط بشكل افتراضي كما يلي:

- الروابط التي لم نقم باستخدامها مسبقًا تظهر مسطرة وباللون الأزرق.

- الروابط التي التي قمنا باستخدامها مسبقًا تظهر مسطرة وباللون البنفسجي.

خاصية target

عند استخدام خاصية **target** في معلومات الرابط، فإننا نحدد أين سيتم فتح الصفحة المرتبطة بهذا الرابط. يُمكن لهذه الخاصية أن تأخذ القيم التالية:

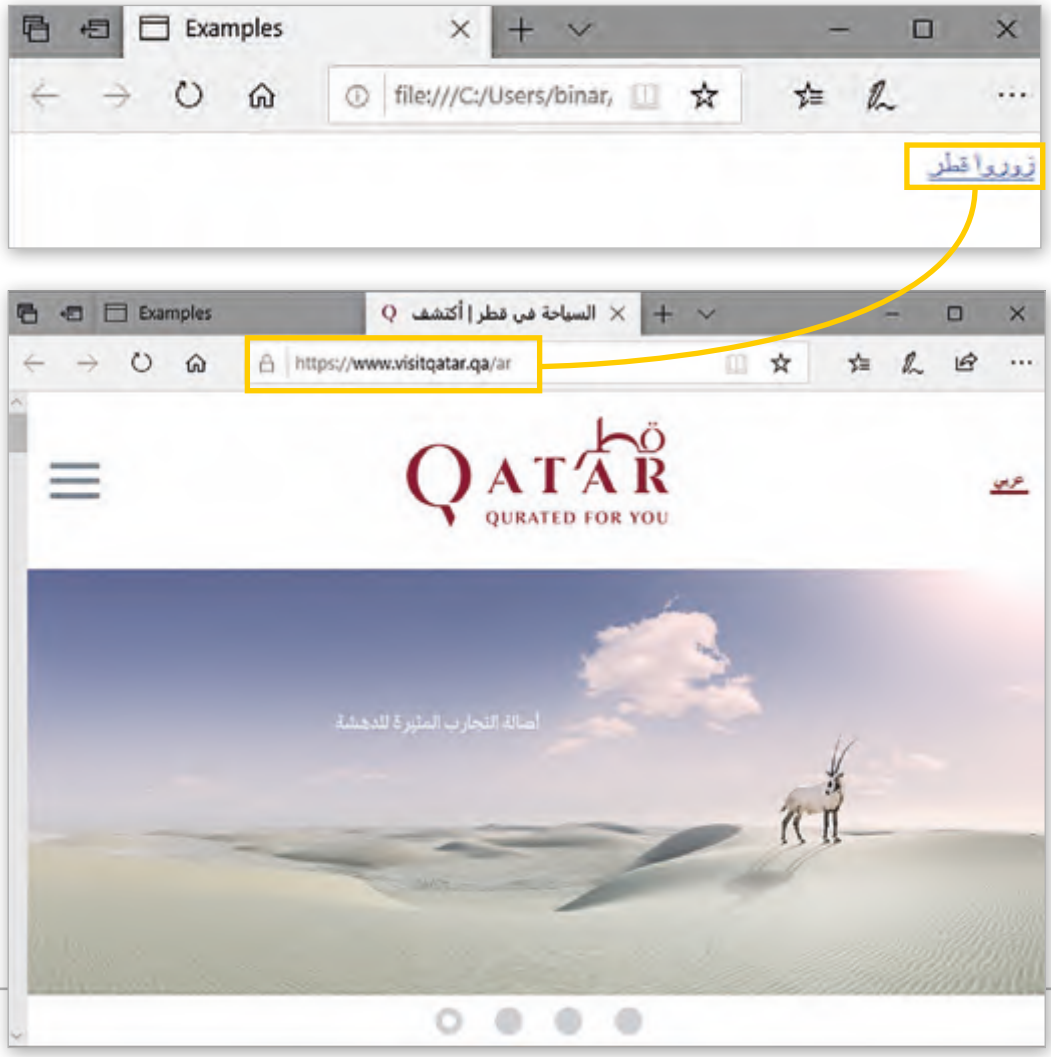
خاصية target	
الخاصية	**الوظيفة**
_blank	سيتم فتح الصفحة في علامة تبويب جديدة.
_self	سيتم فتح الصفحة في نفس علامة التبويب.
_parent	سيتم فتح الصفحة في الإطار الرئيس.
_top	سيتم فتح الصفحة في الإطار الداخلي من النافذة.

على سبيل المثال، إذا أردنا فتح الرابط في نافذةٍ جديدة، فسنقوم بالخطوات التالية.

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
</head>
<body>
   <a href="https://www.visitqatar.qa/ar"target="_blank">زوروا قطر</a>
</body>
</html>
```

تتم إضافة خاصية target هنا في داخل وسم <a> المفتوح.



إنشاء شريط التصفح

في مشروعنا قمنا بإدراج قائمة مرتبة كشريط للتصفح. تتكون هذه القائمة من مجموعة من الروابط، وعمومًا بعض عناصر هذه القائمة يجب أن يتم ربطه بجزء محدد من الصفحة، بينما يتم ربط العنصر "**Contact Us**" مع صفحةٍ أخرى في نفس الموقع.

إنشاء ارتباط بجزء محدد في نفس الصفحة

قبل البدء بربط الجزء المحدد من الصفحة، علينا تمييز الجزء من الصفحة الذي سيتم الإشارة إليه عبر الرابط. لهذا الغرض سنستخدم خاصية المُعرف "**id**".

تستخدم خاصية "id" مع كل عنصر HTML لتمييز العنصر عن باقي العناصر في صفحة الويب.

```
<h1>العنابي</h1>
<h1 id="history">منتخب قطر لكرة القدم </h1>
<p>ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، وبحلول عام 1948 لعب
عمال النفط المغتربين أول مباراة رسمية لكرة القدم في قطر، ثم تأسس الاتحاد القطري لكرة القدم في
عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIfA عام 1970.</p>
<p>خاض المنتخب الوطني القطري مباراته الأولى في تصفيات FIfA، 1977
المؤهلة لكأس العالم عام 1980 والذي استضافته دولة قطر. وقد حقق الفريق
وشارك الفريق لأول مرة في كأس آسيا عام 1992 للمرة الأولى في تاريخه، وتوج بطل الخليج للمرة الثانية في
نجاحات ومراكز مختلفة خلال تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على المستوى
الإقليمي، حيث فاز بكأس الخليج 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر كمستضيف
تاريخه في عام 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى العالم
لكأس العالم.</p>
<p>حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب كأس
آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداءً راقيًا ليكون العنابي أول
فريق يتوج بالكأس الآسيوية في نسختها الجديدة.</p>
<h1 id="gallery">ألبوم الصور</h1>
<h1 id="about">من نحن</h1>
<p>تم إنشاء هذا الموقع للتعريف بالعنابي، وإطلاع جمهوره ومشجعيه على أهم الأخبار والمناسبات
المتعلقة به، العنابي هو المنتخب الوطني لكرة القدم بدولة قطر ويشرف عليه الاتحاد القطري لكرة
القدم. </p>
<h2>الاتحاد القطري لكرة القدم</h2>
</body>
```

يمكن تعيين الـ id بكلمة تبدأ بحرف أو بإشارة (_)، ولا يمكن تعيين نفس الاسم لعنصرين مختلفين في نفس الصفحة.

لربط عنصر بمحتوى داخل الصفحة، نستخدم الـ id الخاص بهذا المحتوى مسبوقًا بعلامة (#).

لنطبق هذا في مشروعنا.

هذا يقوم بالانتقال إلى أعلى الصفحة الحالية بشكل تلقائي.

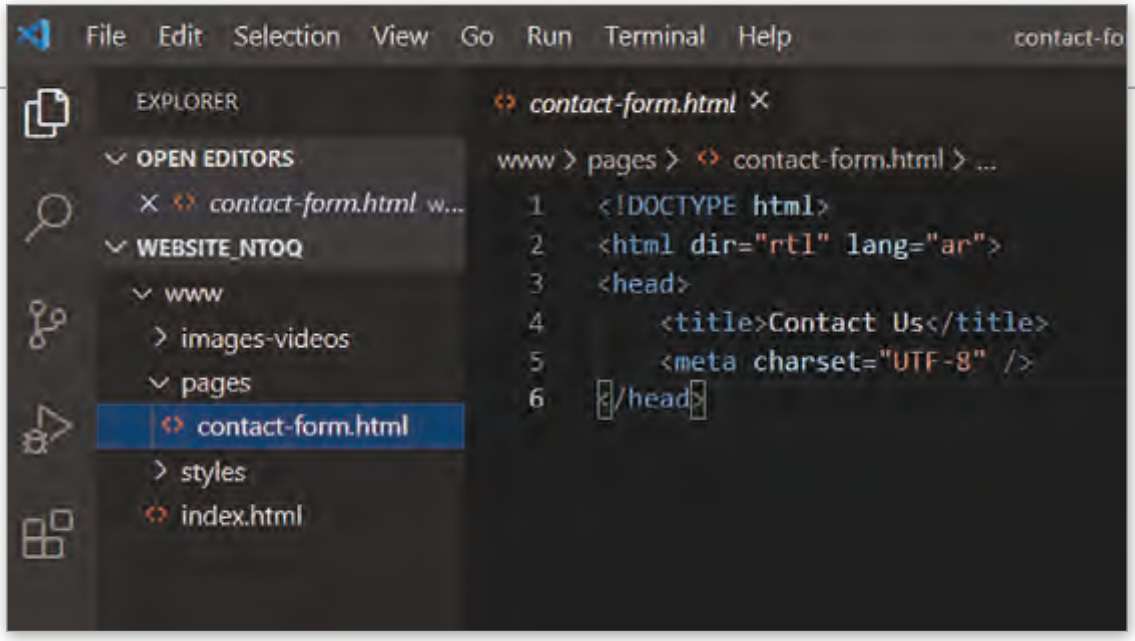
.html

```
<ul>
   <li><a href="#top">الصفحة الرئيسة</a></li>
   <li><a href="#history"> تاريخ المنتخب القطري</a></li>
   <li><a href="#gallery"> ألبوم الصور</a></li>
   <li><a href="#about">من نحن</a></li>
   <li>اتصل بنا</li>
</ul>
```

الارتباط إلى صفحة أخرى في نفس موقع الويب

لقد قمنا بربط 3 علامات تبويب في شريط التصفح إلى أجزاء محددة في نفس الصفحة. ستشير علامة تبويب **"Contact Us"** إلى صفحة ويب أخرى من موقعنا.

في البداية فلننشئ ملف **HTML** كما تعلمنا في الدرس السابق، وسنطلق على هذا الملف اسمًا، وليكن: **"contact-form.html"**.



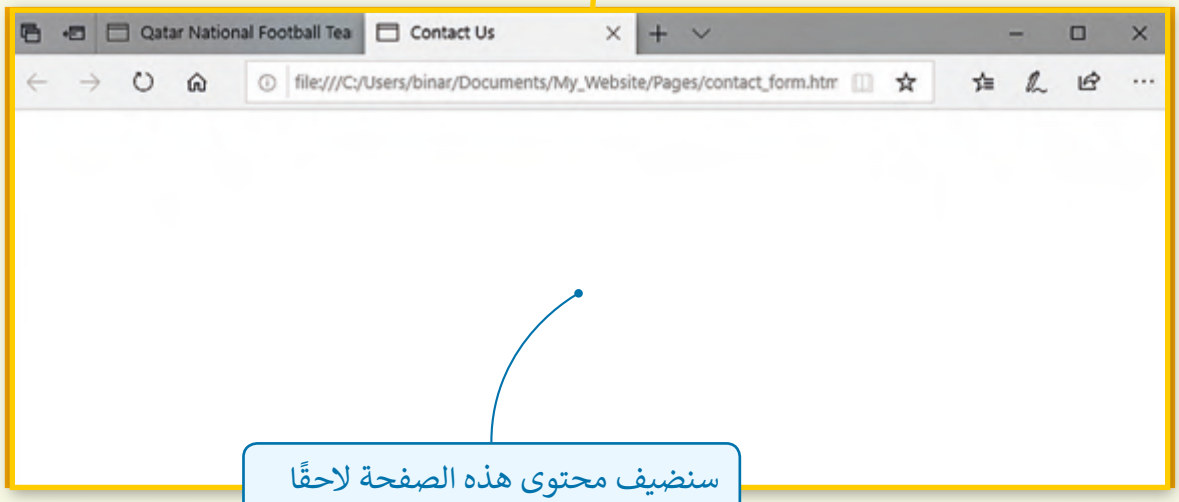
عند الربط بصفحات أخرى في نفس الموقع، فإننا نستخدم عنوان **URL** النسبي.

إذا كانت صفحات الموقع في نفس المجلد، فإن قيمة خاصية **href** تتكون من اسم الملف الذي يتم الربط به.

.html

```
<body>
  <ul>
    <li><a href="#top">الصفحة الرئيسة</a></li>
    <li><a href="#history"> تاريخ المنتخب القطري</a></li>
    <li><a href="#gallery"> ألبوم الصور</a></li>
    <li><a href="#about">من نحن</a></li>
    <li><a href="contact-form.html" target="_blank">اتصل بنا<a></li>
  </ul>
</body>
```

Qatar National Football
file:///C:/Users/binar/Documents/My_Website/Pages/Site.html
الصفحة الرئيسة
تاريخ المنتخب القطري
ألبوم الصور
من نحن
اتصل بنا
العنابي
منتخب قطر لكرة القدم
ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، ويحلول عام 1948 لعب عمال النفط المحترفين أول مباراة رسمية لكرة القدم في قطر. ثم تأسس الاتحاد القطري لكرة القدم في عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIFA في عام 1970.
خاض المنتخب الوطني القطري مبارياته الأولى في تصفيات FIFA المؤهلة لكأس العالم عام 1977، وشارك الفريق لأول مرة في كأس آسيا عام 1980 والذي استضافته دولة قطر. وقد حقق الفريق نجاحات ومراكز مختلفة خلال
تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على المستوى الإقليمي، حيث فاز بكأس الخليج 1992 للمرة الأولى في تاريخه، وتوج ببطل الخليج للمرة الثانية في تاريخه في عام 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر كمستضيف لكأس العالم 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى العالم.
حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب كأس آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداءً رائعًا ليكون العنابي أول فريق يتوج بالكأس الآسيوية في نسختها الجديدة.
ألبوم الصور
من نحن


Qatar National Football Tea
Contact Us
file:///C:/Users/binar/Documents/My_Website/Pages/contact_form.htm
سنضيف محتوى هذه الصفحة لاحقًا

روابط البريد الإلكتروني

هناك نوع من الروابط يختص بتشغيل تطبيق البريد الإلكتروني الخاص بالمستخدم عند الضغط عليه. يتم ذلك عبر تحديد قيمة خاصية **href** لتبدأ بـ **mailto:** متبوعة بعنوان البريد الإلكتروني الذي سيتم الإرسال إليه.

.html

```
<a href="mailto:emailaddress@example.com"> Clickable text</a>
```

.html

```
<h2>الاتحاد القطري لكرة القدم</h2>
<ul>
   <li><a href="mailto:info@example.com">info@example.com</a></li>
</ul>
```

نصيحة ذكية

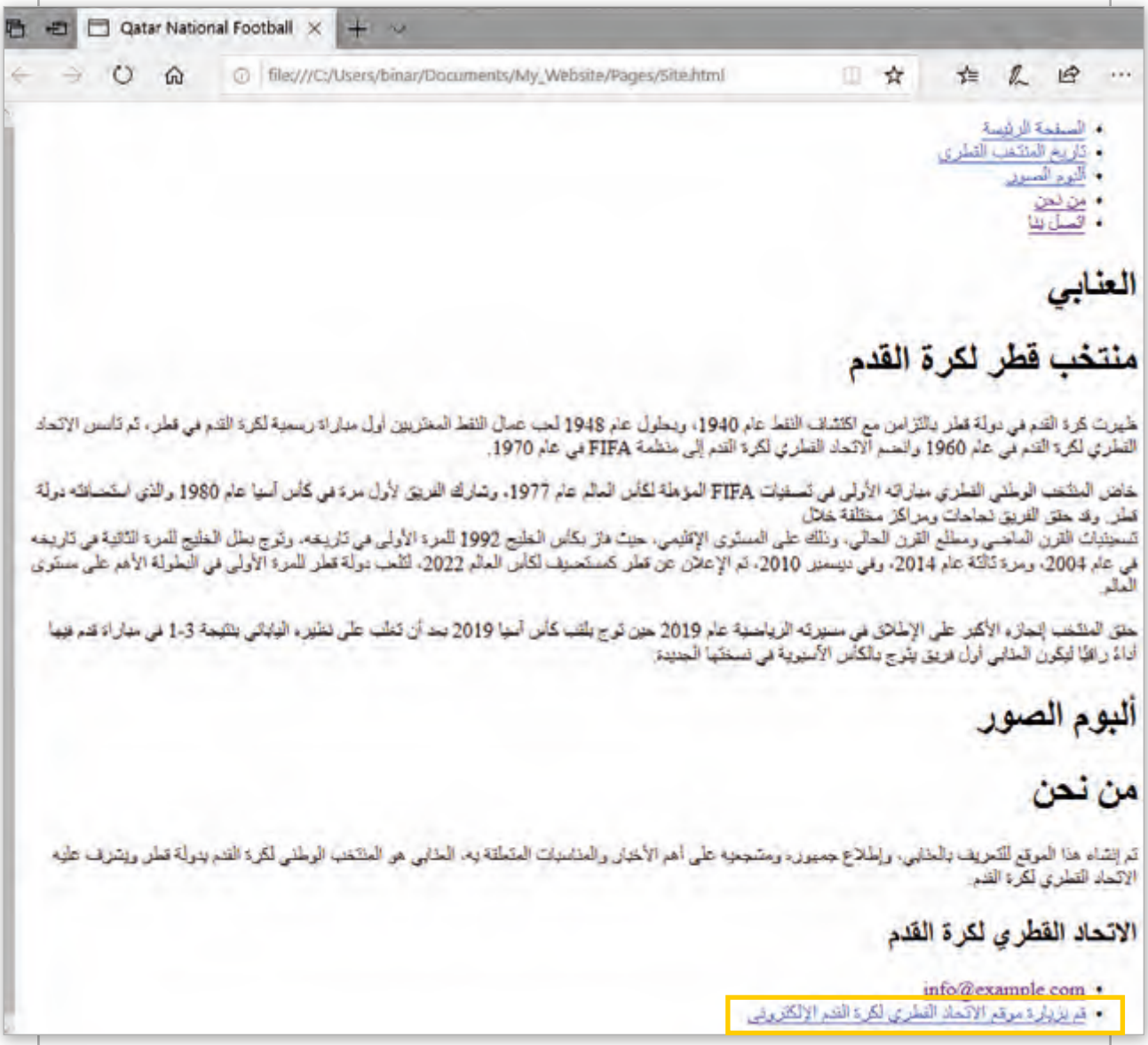
يجب وضع الوسوم الرئيسة داخل قسم (<body>....</body>) في مستند Html. إن وضعها في أي مكان آخر سيتسبب في حدوث أخطاء.

الارتباطات إلى مواقع أخرى

لقد رأينا أنه يمكن استخدام عناوين **URL** التي تأخذنا إلى موقع ويب مختلف. فلنطبق هذا في موقعنا.

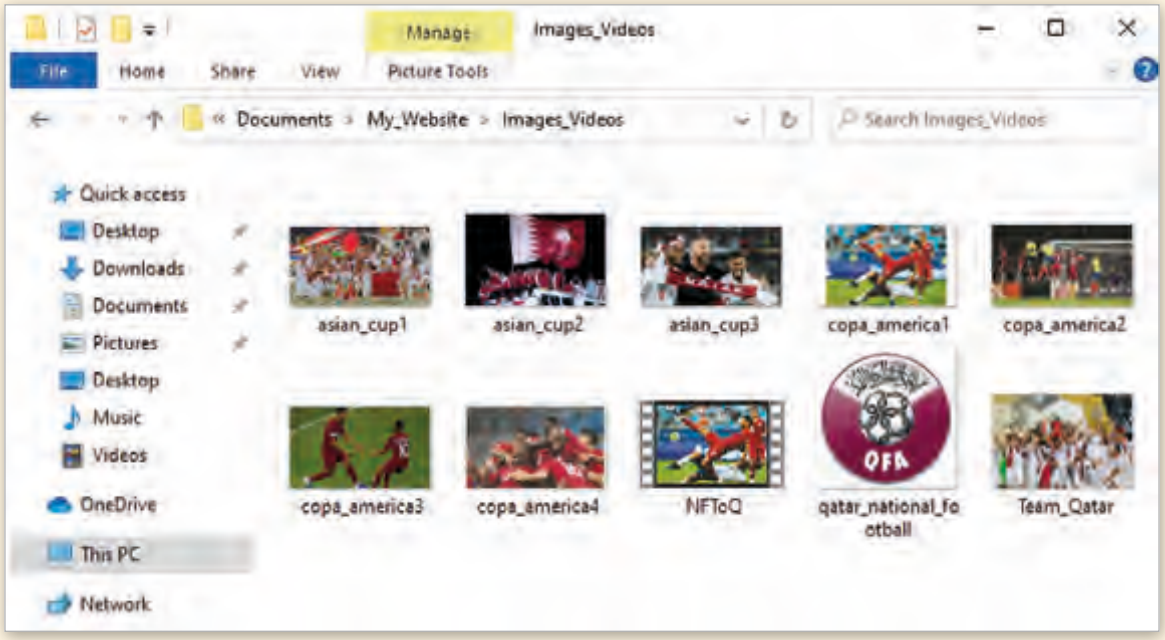
.html

```
<h2>الاتحاد القطري لكرة القدم</h2>
<ul>
   <li><a href="mailto:info@example.com">info@example.com</a></li>
   <li><a href="https://www.qfa.qa/first-national-team/">قم بزيارة موقع
   الاتحاد القطري لكرة القدم الإلكتروني </a></li>
</ul>
```

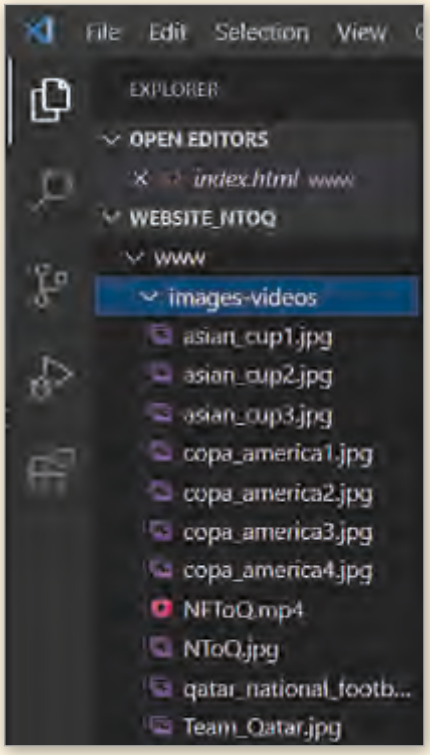


إضافة الصور والفيديو

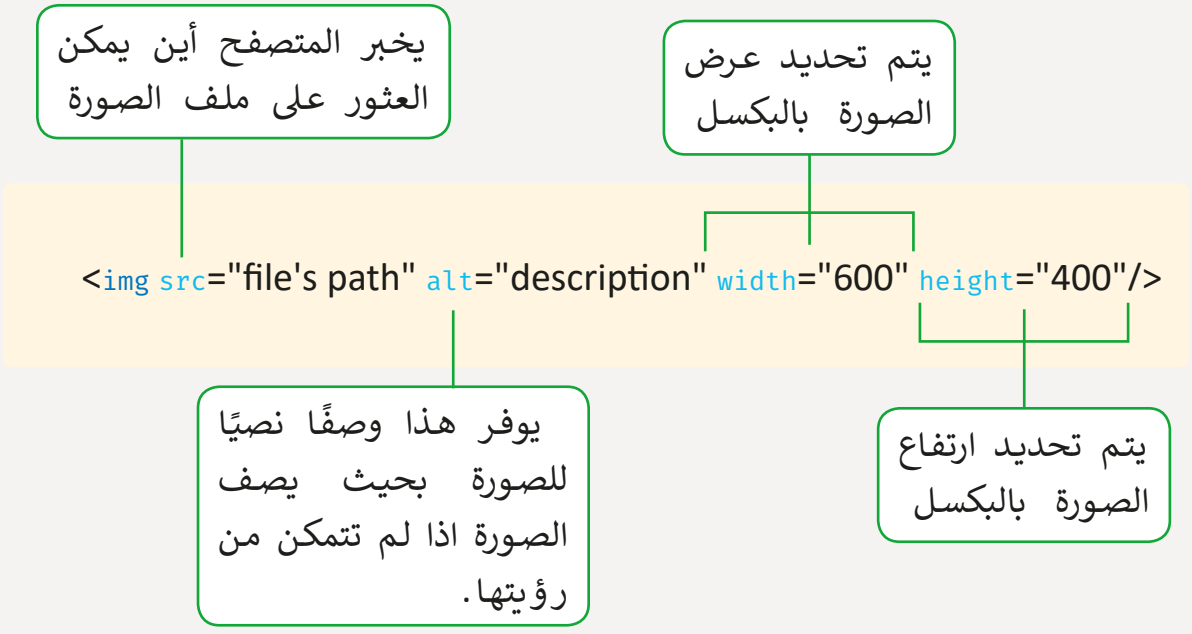
تعلمنا في السابق كيفية إضافة الصور إلى الصفحات، وأهميتها في إظهار الموقع بشكلٍ احترافي و جذاب. من الممارسات الجيدة أن نقوم بالاحتفاظ بالصور في مجلدٍ منفصل عن باقي ملفات الموقع، لذا كنا قد أنشأنا مجلد فرعي باسم **"Images_Videos"** ستتم فيه عملية إضافة الصور التي سنستخدمها في موقع الويب الخاص بنا.



عند استخدام Visual Studio Code، وفي قسم Explorer، يوجد مجلد فرعي باسم Images_Videos يمكننا من رؤية جميع ملفات الصور المتوفرة للاستخدام في الموقع.



يُستخدم الوسم <img> لإضافة الصور إلى صفحة الويب، و يتميز هذا الوسم بعدم وجود وسم إغلاق.



مسارات ملف HTML	
المسار	الوصف
<img src="picture.jpg">	الصورة picture.jpg توجد في المجلد نفسه الذي توجد به الصفحة الحالية.
<img src="images/picture.jpg">	الصورة picture.jpg توجد في مجلد images داخل المجلد الحالي.
<img src="/images/picture.jpg">	picture.jpg توجد في مجلد images الموجود في المجلد الرئيس للصفحة الحالية.
<img src="../picture.jpg">	picture.jpg توجد في مجلد أعلى بمستوى واحد من المجلد الحالي.

لنضيف الصور إلى موقعنا:

```
<h1 id="gallery">ألبوم الصور</h1>
<img src="images-videos/asian_cup1.jpg" width="600" height="400">
<img src="images-videos/asian_cup2.jpg" width="600" height="400">
<img src="images-videos/copa_america2.jpg" width="600" height="400">
```

بالإضافة إلى الصور، يمكننا أيضًا إضافة فيديو إلى مستندنا باستخدام الوسم <video>.
هذا الوسم له بعض الخصائص التي تسمح لنا بالتحكم بالفيديو.

يحدد ارتفاع وعرض الفيديو بالبكسل.

الصورة التي ستظهر أثناء تنزيل الفيديو أو حتى يضغط المستخدم زر التشغيل.

```
<video src="file's path" poster=" file's path" width="600" height="400"
preload controls loop>
```

يشير إلى أن الفيديو يعاد تشغيله تلقائيًا بعد انتهائه.

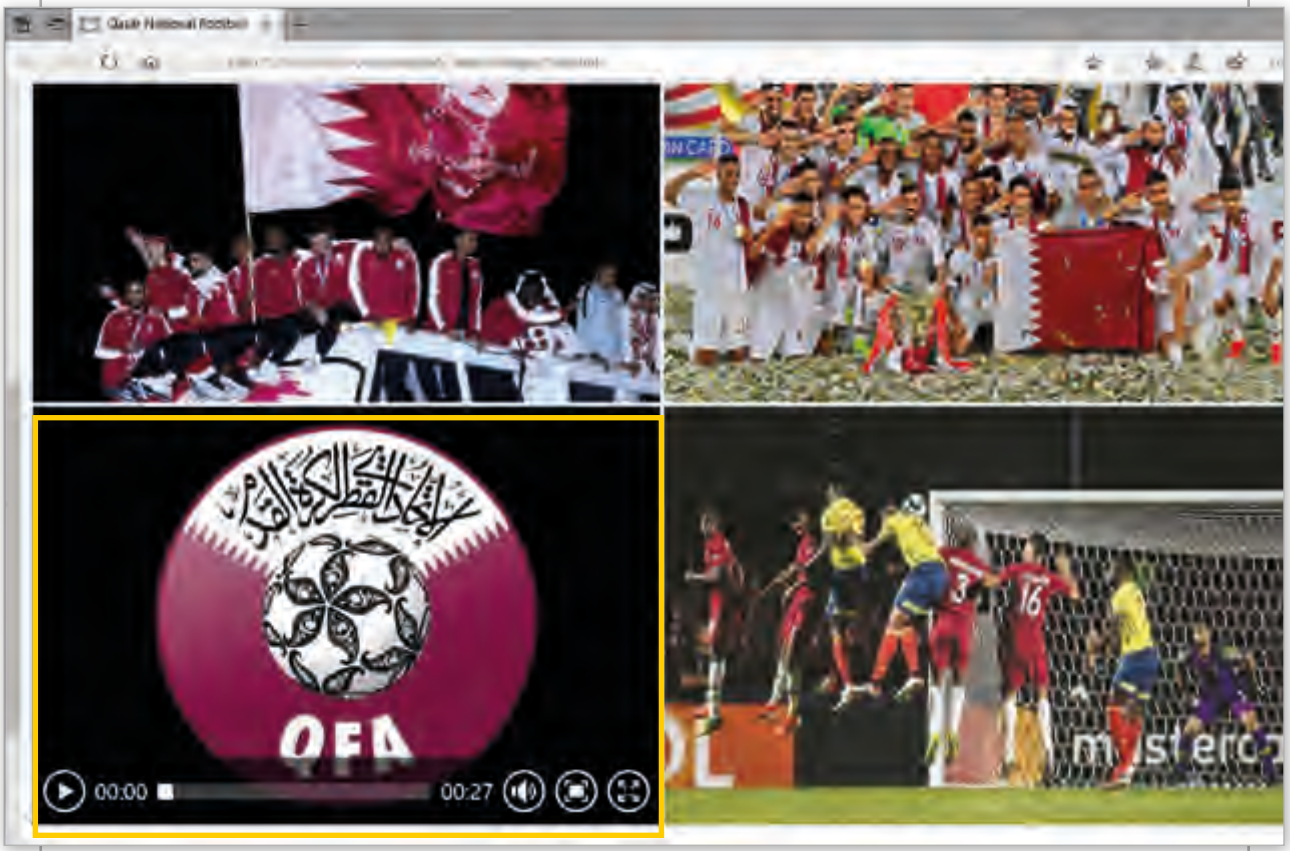
يشير إلى أن المتصفح يجب أن تكون لديه أزراره الخاصة للتشغيل.

يخبر المتصفح بما يجب فعله عند تحميل الصفحة.

```
<h1 id="gallery">ألبوم الصور</h1>
<img src="images-videos/asian_cup1.jpg" width="600" height="400">
<img src="images-videos/asian_cup2.jpg" width="600" height="400">
<img src="images-videos/copa_america2.jpg" width="600" height="400">
<!--Let's add the video to our site-->
<video src="images-videos/NFToQ.mp4" poster="../Images_Videos/qatar_
national_football.png" width="600" height="400" preload controls
loop>
   <p>لا يمكن عرض الفيديو</p>
</video>
```

هذا تعليق. لا يتم عرض التعليقات من قبل المتصفح.

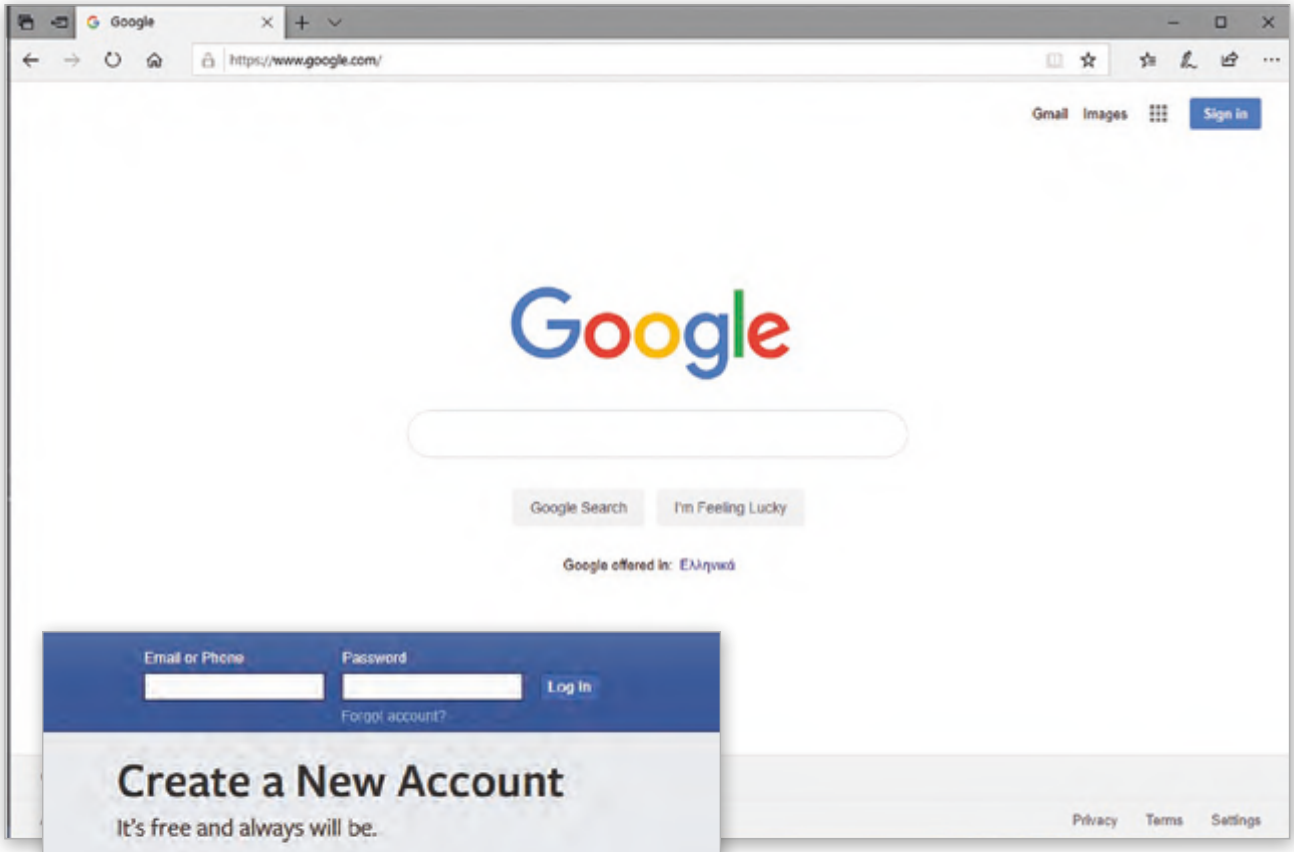
في حالة كان المتصفح لا يدعم عنصر الفيديو <video> أو صيغته، سيتم عرض ما هو موجود بين وسم الفتح ووسم الإغلاق .

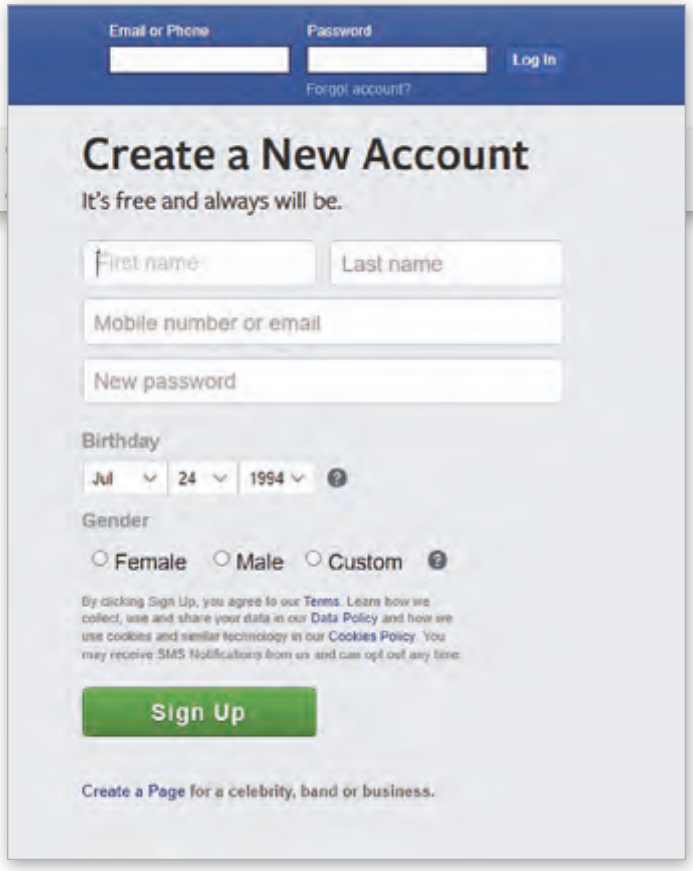


إنشاء نموذج اتصال Form

لإكمال بُنية موقعنا الإلكتروني، سنقوم بإنشاء نموذج اتصال.

النموذج هو مستند يحتوي على حقول لإدخال البيانات. لكل حقل في النموذج تسمية محددة تُمكِّن المستخدم الذي يقوم باستعراض النموذج من الحصول على فكرة عن عناصره المختلفة.





من النماذج الشائعة الاستخدام نموذج صندوق البحث الخاص بـ Google و نموذج التسجيل في Facebook.

آلية عمل النموذج

عندما نرى نموذجًا في صفحة الويب، فإن صناديق الإدخال والخيارات في هذا النموذج يتم إنشاؤها باستخدام لغة **HTML**.

عندما يقوم المستخدم بتعبئة النموذج ويضغط زر إرسال النموذج، فإن المعلومات تُرسل إلى الخادم لمعالجتها أو لحفظها.

بُنية النموذج

يحتوي النموذج على عدة عناصر تحكم يجمع كل منها معلومات مختلفة. توجد عناصر التحكم بالنموذج داخل الوسم **<form>**، ويتطلب كل وسم **<form>** خاصية **action**. تستقبل هذه الخاصية قيمة هي عبارة عن عنوان الرابط التشعبي (URL) الخاص بالصفحة الموجودة على الخادم، والتي ستتلقى المعلومات من النموذج عند إرسالها.

على سبيل المثال:

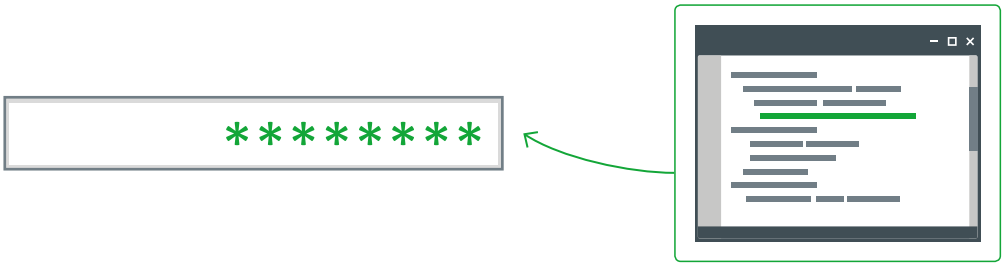
```
<form action="http://www.myserver.qa/comments.php>
```

وسم صندوق الإدخال <input>

يُستخدم الوسم **<input>** لإنشاء طرق تحكم مختلفة بعناصر النموذج، يمكن عرضها بطرق متعددة وفقًا لنوع الخاصية. في الجدول التالي نستعرض بعضًا منها:

الوسم input ليس له وسم إغلاق.

<input>	
النوع	الوصف
<input type="text">	ينشئ سطرًا واحدًا مخصصًا للنص.
<input type="password">	ينشئ سطرًا واحدًا مخصصًا للنص مع خاصية إخفاء النص وتستخدم مع كلمة المرور Password
<input type="email">	ينشئ سطرًا واحدًا مخصصًا للبريد الإلكتروني بالإضافة الى خاصية التأكد من كتابة البريد الالكتروني بشكل صحيح و كامل.
<input type="submit">	ينشئ زر اعتماد خاص بالنموذج لإرساله إلى الخادم.

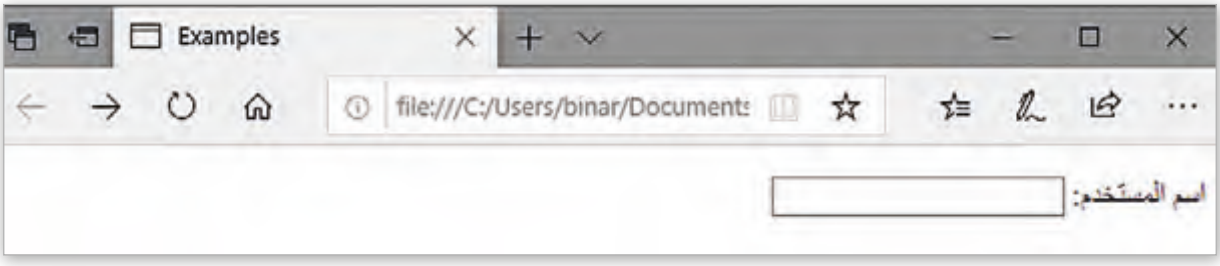


لنستعرض مثالًا على كل نوع

<input type="text">

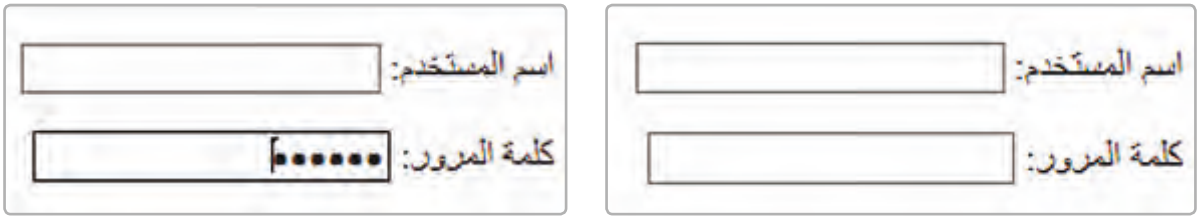
```
<form action="http://www.myserver.qa/comments.php">
   <p>اسم المستخدم:
      <input type="text" name="username">
   </p>
</form>
```

خاصية name تشير إلى اسم عنصر الإدخال.



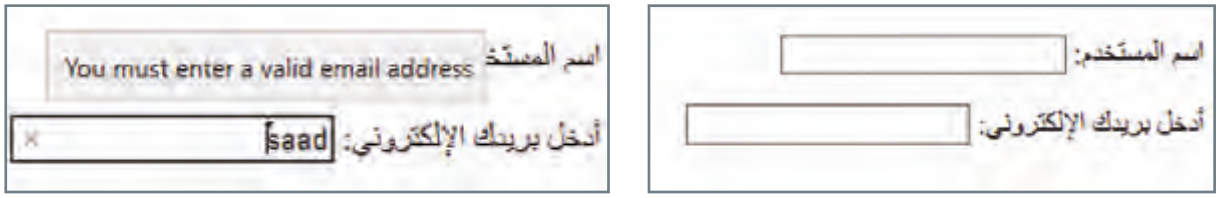
<input type="password">

```
<form action="http://www.myserver.qa/comments.php">
   <p>اسم المستخدم:
      <input type="text" name="username" >
   </p>
   <p>كلمة المرور:
      <input type="password" name="password">
   </p>
</form>
```



<input type="email">

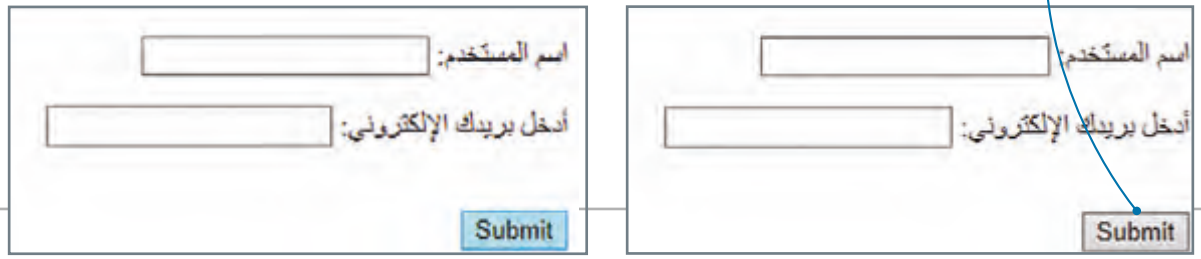
```
<form action="http://www.myserver.qa/comments.php">
   <label for="username">اسم المستخدم:</label>
   <input type="text" name="username">
   <label for="email">أدخل بريدك الإلكتروني:</label>
   <input type="email" name="email">
   <input type="submit" value="Submit">
</form>
```



<input type="submit">

```
<form action="http://www.myserver.qa/comments.php">
   <fieldset>
      <legend>تواصل</legend>
      <label for="username">اسم المستخدم:</label>
      <input type="text" name="username">
      <label for="email">أدخل بريدك الإلكتروني:</label>
      <input type="email" name="email">
      <input type="submit" value="Submit">
   </fieldset>
</form>
```

تستخدم خاصية value لتعيين النص الذي سيظهر على زر الأمر.



وسم أداة التسمية <label>

يستخدم الوسم **<label>** كأداة لتسمية الوسم **<input>**، ويستعمل مع خاصية **for** التي تحدد نموذج التحكم الخاص بأداة التسمية.

على سبيل المثال:

<label>

```
<form action="http://www.myserver.qa/comments.php">
   <label for="username">اسم المستخدم:</label>
   <input type="text" name="username">
   <label for="email">أدخل بريدك الإلكتروني:</label>
   <input type="email" name="email">
   <input type="submit" value="Submit">
</form>
```

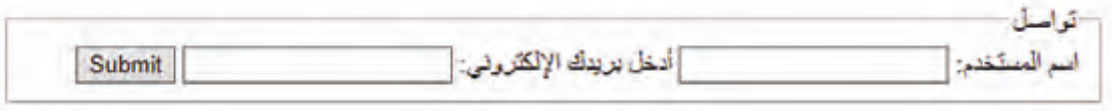
اسم المستخدم: أدخل بريدك الإلكتروني: Submit

وسم تجميع عناصر التحكم <fieldset>

يستخدم الوسم **<fieldset>** لتجميع عناصر تحكم النموذج معًا داخل إطار، مع إمكانية تسمية مجموعة الحقول باستخدام الوسم **<legend>**.

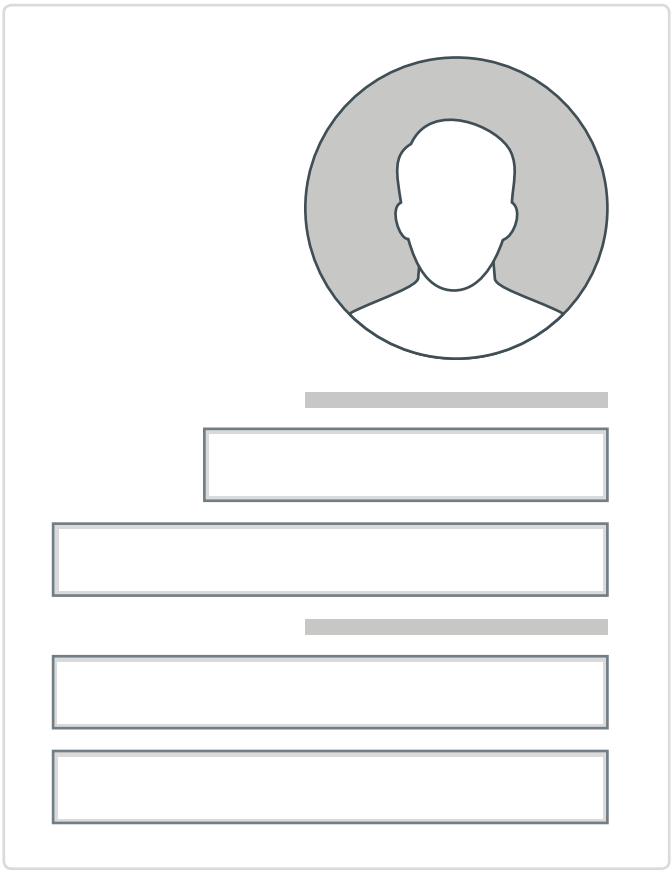
<fieldset>

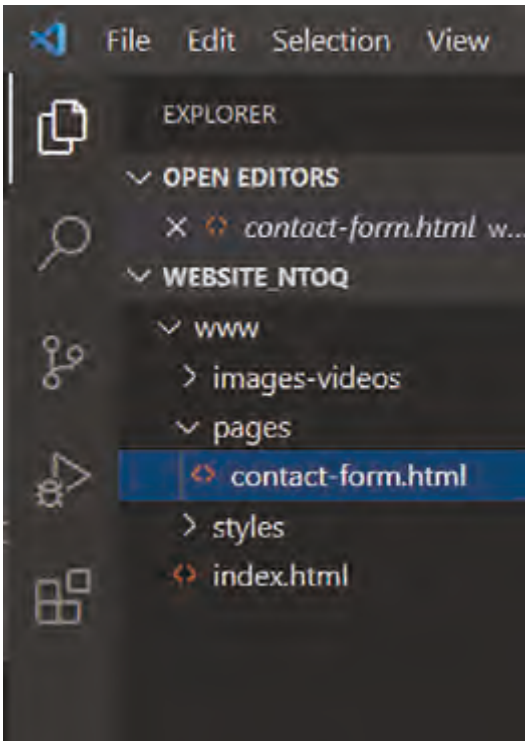
```
<form action="http://www.myserver.qa/comments.php">
  <fieldset>
    <legend>تواصل</legend>
    <label for="username">اسم المستخدم:</label>
    <input type="text" name="username">
    <label for="email">أدخل بريدك الإلكتروني:</label>
    <input type="email" name="email">
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



لقد حان الوقت لكي نطبق ما تعلمناه عن النماذج في موقع الويب الخاص بنا.

في المجلد الرئيس سننشئ صفحة باسم "**contact-form.html**".





سننشئ نموذجًا يحتوي على أربعة حقول لتعبئتها بالمعلومات التالية:

- < الاسم الأول.
- < اسم العائلة.
- < البريد الإلكتروني.
- < الرسالة.

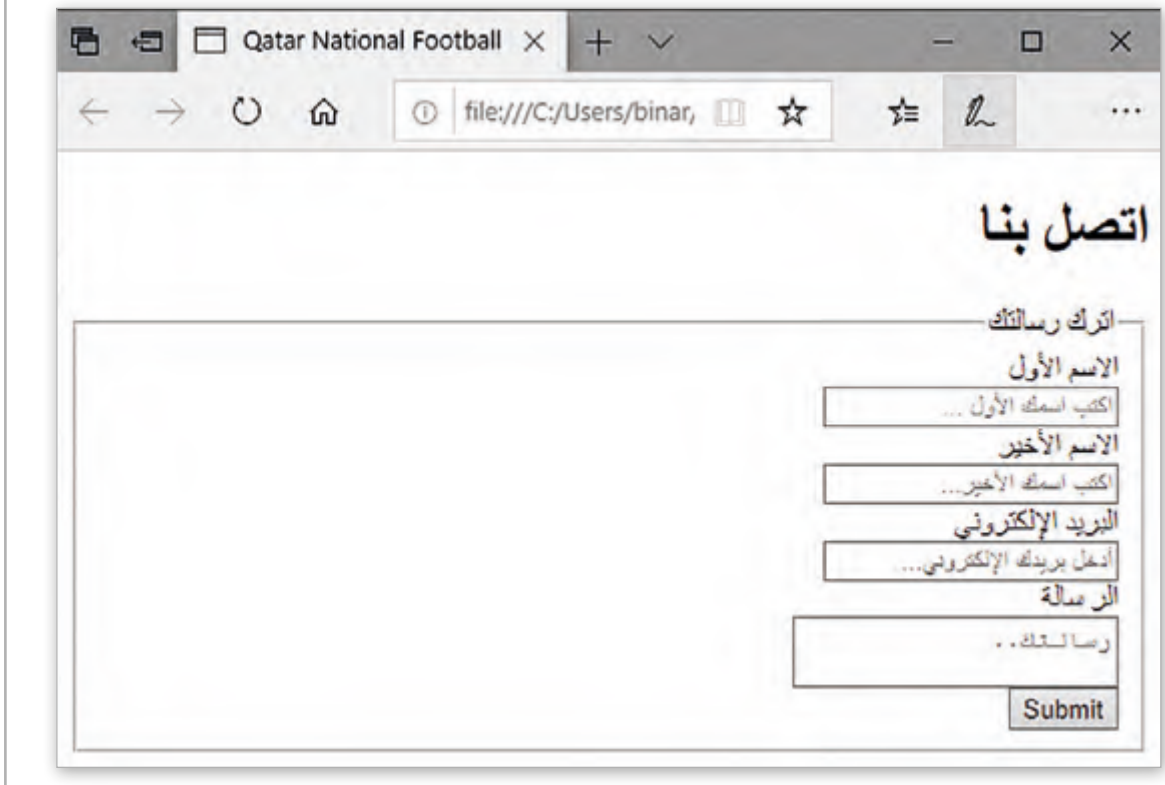
في مشروعنا، سنستخدم أيضًا العنصر <textarea> الذي يُستخدم لإنشاء حقل نصي متعدد الأسطر. على عكس عناصر الإدخال الأخرى، يحتوي هذا العنصر على وسمي فتح وإغلاق.

قم بكتابة المقطع البرمجي التالي:

```
<h1> اتصل بنا</h1>
<form action="http://www.myserver.qa/comments.php">
  <fieldset>
    <legend>اترك رسالتك</legend>
    <label for="name">الاسم الأول</label><br>
    <input type="text" placeholder="... اكتب اسمك الأول"><br>
    <label for="Last name">الاسم الأخير</label><br>
    <input type="text" placeholder="...اكتب اسمك الأخير"><br>
    <label for="email">البريد الإلكتروني</label><br>
    <input type="email" placeholder="...أدخل بريدك الإلكتروني"><br>
    <label for="message">الر سالة</label><br>
    <textarea placeholder="رسالتك.." ></textarea><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

وسم سطر جديد، وهو وسم مفرد.

placeholder هي عبارة عن خاصية قيمتها هي النص الظاهر للمستخدم في صندوق النص حتى يتم الضغط عليه.



1

اختر الإجابة الصحيحة:

1. أي من الخصائص التالية يتولى إخبار المتصفح بما سيتم فتحه عند الضغط على رابط تشعبي؟	src
	url
	link
	href
2. ما هو وسم HTML المستخدم لإنشاء قائمة تعداد (bulleted list)؟	ol
	ul
	br
	bl
3. أيّ طريقة من الطرق التالية تستخدم لإضافة التعليق	"this is a comment"
	this is a comment//
	</this is a comment>
	<!--this is a comment-->

2

قم بكتابة أوامر HTML المناسبة للقيام بما يلي:

1. تعيين رابط url الخاص بصورة.

```
<img ........="picture.jpg">
```

2. جعل العنصر بالأسفل وداخل الرابط.

```
<a........=" https://www.visitqatar.qa/ar"> </a>
```

3. تحديد نص بديل خاص بصورة ما.

```
<img src="picture.png"  ....="The image can't be displayed">
```

4. كتابة الوسم الصحيح لإضافة فقرة بنص "this is a paragraph".

```
<html>
<body>
 ............
</body>
</html>
```

5. إضافة فاصل أسطر في داخل الفقرة.

```
<p>This..... is a paragraph.</p>
```

6. إضافة النص " Souq Waqif " داخل <ul>.

```
<ul>
 .......
</ul>
```

3

قم بتعبئة الفراغات بالوسم tag، أو العنصر element، أو الخاصية attribute المناسبة لإنشاء النموذج.

1. قم بإضافة زر إدخال "button" وبقيمة "Send".

```
<form>
...........
</form>
```

2 . أضف زر إرسال "submit" بحيث يتم إرسال النموذج عند الضغط عليه إلى الرابط :

```
http://www.my server/comments.php
<form.............="/action_page.php"/>
<input type="text" name="Name">
.............................................
</form>
```

3. قم بإضافة حقل إدخال يستخدم من قبل المستخدم لإدخال كلمة سر، وحقل إدخال آخر خاص للبريد الإلكتروني.

```
<form action="http://www.my server/comments.php">
  <input type="text" name="Name">
  <input type=.............. ........name="email">
  <input type=.....................
  name="Password">
</form>
```

4

لننشئ بعض صفحات الويب.

1. أنشئ قائمة عناصر مرتبة تتكون من 3 روابط لمواقع ويب مختلفة، يفتح كل منها في نافذة جديدة.
2. أنشئ قائمة تعداد بأطعمتك المفضلة.
3. اعرض صورة بحيث يتم فتح صفحة ويب لمحرك بحث من اختيارك (مع مراعاة أن يفتح في نافذة جديدة) عندما يتم الضغط على تلك الصورة.
4. أنشئ مقطع فيديو متضمنًا صورك المفضلة باستخدام برنامج Photos ثم قم بإضافتها إلى HTML الخاص بك.

5

طور صفحة ويب بسيطة باستخدام وسوم HTML التي تعلمتها مسبقًا في الدرس. حاول أن تجعل هذه الصفحة تبدو مختلفة عن صفحة HTML التي طورناها في المثال خلال الدرس، مع تضمين العناصر التالية في هذه الصفحة:

وسوم HTML الصحيحة الخاصة بـ HTML و body.

أضف الجمل التالية بدون الأقواس كوسم للعنوان:

- < "ادخل اسمك."> HTML Exercise by

مثلًا: HTML Beginner Exercise by Saad.

- قائمة غير مرتبة unordered list (تتضمن 4 عناصر على الأقل).
- قائمة مرتبة ordered list (تتضمن 4 عناصر على الأقل).
- أضف الترويسات Headings والفقرات Paragraphs حسب الحاجة.
- رابط فعال لإرسال بريد إلكتروني.
- 3 روابط على الأقل لمواقع ويب عبر شبكة الإنترنت، مع استعمال الوصف المناسب (كلمة أو جملة) داخل الوسم بدلًا من عرض الرابط.
- صورتان بامتداد jpg. أو gif. تقوم باختيارهما من موقع ويب.

6

إضافة المحتوى إلى موقع الويب الخاص بالسفر.

- افتح المجلد "Adventure_site" الذي أنشأته في الدرس السابق.

- أنشئ قائمة غير مرتبة بالعناصر التالية: Home، About، Photo Album، Contact، هذه القائمة ستكون عبارة عن شريط التصفح بحيث يتم ربط كل عنصر بقسمه الخاص من الصفحة.

- أضف الترويسات Headings والفقرات Paragraphs حسب الحاجة.

- أضف الصور التي تريدها.

- أضف مقطع فيديو.

- قم بإنشاء قائمة غير مرتبة أسفل الصفحة بروابط مفيدة للمستخدم، إضافة إلى رابط لبريدك الإلكتروني بحيث يتمكن المستخدم من الاتصال بك.

- أخيرًا أنشئ ملف HTML جديد خاص بنموذج يتم إنشاؤه لكي يتمكن المستخدمين من ترك رسالة. سيحتوي هذا النموذج على 4 حقول: الاسم الأول (name)، والاسم الأخير (Last name)، والبريد الإلكتروني (e-mail) والحقل الأخير خاص بالتعليق (Comment)، ثم أضف زر الإرسال (Submit). حدد موقع إرسال النموذج إلى http://www.myserver.qa/comments.php.

من المهم القيام بربط عنصر "Contact" الموجود في شريط التصفح إلى هذه الصفحة.

- قم بحفظ عملك.

تطبيق قواعد CSS

تجميع النص

يعد تقسيم صفحة الويب إلى أقسام رئيسة خطوة مهمة جدا في تصميم موقع الويب.

يتم استخدام وسم التقسيم ** ** في **HTML** لإنشاء أقسام للمحتوى في صفحة الويب مثل (النص، الصور، الرأس، التذييل، شريط التنقل، إلخ).

يعتبر وسم div كحاوية يمكننا وضع أكثر من عنصر HTML داخلها، وهذا يعتبر مفيدًا جدًا في تطبيق أنماط CSS على المجموعة بأكملها.

تخطيط الصفحات

عادةً ما نستخدم خاصية **id** وخاصية class مع عنصر ** ** لتعديل نمط **CSS** بسهولة.

يتم استخدام خاصية **class** مع **HTML** لتطبيق نفس الأنماط ذات نفس اسم الفئة **(class)**، وبهذا فإن العناصر التي لها نفس خاصية **class** لها نفس الشكل والنمط.

هيا بنا ننشئ تخطيطًا للعناصر التي سنستخدمها لتطبيق خاصيتي div وclass عليها. ستحتوي صفحتنا على 5 أقسام رئيسة:

- Header
- History
- Gallery
- About
- Footer

فلنرى كيف يمكننا تجزئة المقطع كالتالي:

```
<div class="header">.....</div>
```

الاسم — القيمة

سنقوم بتنظيم صفحتنا باستخدام وسم التقسيم ** ** وخاصية الفصل **class**.

سنقوم بفصل الصفحة إلى الأقسام الرئيسة أولًا ثم أثناء الإنشاء، ربما سنفصل كل قسم إلى أجزاء.

.html

```
<div class="header">
  <div class="nav_bar">
    <ul>
      <li><a href="#top">الصفحة الرئيسة</a></li>
      <li><a href="#history">تاريخ المنتخب القطري</a></li>
      <li><a href="#gallery">ألبوم الصور</a></li>
      <li><a href="#about">من نحن</a></li>
      <li><a href="pages/contact-form.html" target="_blank">اتصل
      بنا</a></li>
    </ul>
  </div>
  <h1>العنابي</h1>
</div>
```

تاريخ المنتخب القطري History

```
<div class="history">
  <h1 id="history">منتخب قطر لكرة القدم </h1>
  <p>ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، وبحلول عام 1948
  لعب عمال النفط المغتربين أول مباراة رسمية لكرة القدم في قطر، ثم تأسس الاتحاد القطري لكرة
  القدم في عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIfA في عام 1970.</p>
  <p>المؤهلة لكأس العالم عام FIfA خاض المنتخب الوطني القطري مباراته الأولى في تصفيات
  1977، وشارك الفريق لأول مرة في كأس آسيا عام 1980 والذي استضافته دولة قطر. وقد حقق
  الفريق نجاحات ومراكز مختلفة خلال تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على
  المستوى الإقليمي، حيث فاز بكأس الخليج 1992 للمرة الأولى في تاريخه، وتوج بطل الخليج للمرة
  الثانية في تاريخه في عام 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر
  كمستضيف لكأس العالم 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى
  العالم.</p>
  <p>حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب
  كأس آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداءً راقيًا ليكون
  العنابي أول فريق يتوج بالكأس الآسيوية في نسختها الجديدة.</p>
</div>
```

ألبوم الصور Gallery

سنقوم بتقسيم gallery div إلى لبنات متشابهة للمساعدة في تصميم الموقع.

```
<h1>ألبوم الصور</h1>
<div class="ResponsiveImage">
  <div class="photos">
    <a target="_blank" href="images-videos/asian_cup1.jpg"><img
    src="images-videos/asian_cup1.jpg" width="600" height="400"></
    a>
  </div>
</div>
<div class="ResponsiveImage">
  <div class="photos">
    <a target="_blank" href="images-videos/asian_cup2.jpg"><img
    src="images-videos/asian_cup2.jpg" width="600" height="400"></
    a>
  </div>
</div>
<div class="ResponsiveImage">
  <div class="photos">
    <a target="_blank" href="images-videos/copa_america2.jpg"><img
    src="images-videos/copa_america2.jpg" ></a>
  </div>
</div>
<div class="ResponsiveVideo">
  <div class="photos">
    <video src="images-videos/NFToQ.mp4" poster="images-videos/
    qatar_national_football.png" width="600" height="400" preload
    controls loop>
      <p>لا يمكن عرض الفيديو</p>
    </video>
  </div>
</div>
```

من نحن About

```
<div class="about">
  <div class="about_header">
    <h1 id="about">من نحن</h1>
    <p>تم إنشاء هذا الموقع للتعريف بالعنابي، وإطلاع جمهوره ومشجعيه على أهم الأخبار
    والمناسبات المتعلقة به، العنابي هو المنتخب الوطني لكرة القدم بدولة قطر ويشرف عليه
    الاتحاد القطري لكرة القدم. </p>
  </div>
</div>
```

التذييل Footer

```
<div class="footer">
  <h2>الاتحاد القطري لكرة القدم</h2>
  <ul class="info_list">
    <li><a href="mailto:info@example.com">info@example.com</a></
    li>
    <li><a href="https://www.qfa.qa/first-national-team/">قم بزيارة موقع
    الاتحاد القطري لكرة القدم الإلكتروني </a></li>
  </ul>
</div>
```

يبدو موقعنا بسيطًا للغاية لأننا قمنا بكتابته بلغة **HTML** فقط، يُمكن الاقتصار على هذا النوع من التصميم ولكن من المفضل استخدام **CSS** لجعل صفحات الويب الخاصة بنا أكثر جاذبية.

Cascading Style Sheets (أوراق التنسيق المتتالية) تستخدم للتحكم في مظهر المستند المكتوب بلغة **HTML**، وهذا يمنحنا القدرة على تعديل مظهر صفحة الويب بشكل خاص والموقع بشكل عام.

وتفيدنا **CSS** في المراحل المختلفة لإنشاء موقعنا، والتي ناقشناها في بداية الوحدة، ونبين أدناه المزايا التي توفرها CSS في كل مرحلة:

1 التخطيط:

تمنحنا الكثير من المرونة، حيث جعلت CSS عملية تنسيق صفحة الويب أمرًا سهلًا مقارنةً بقدرات HTML التقليدية.

2 التصميم:

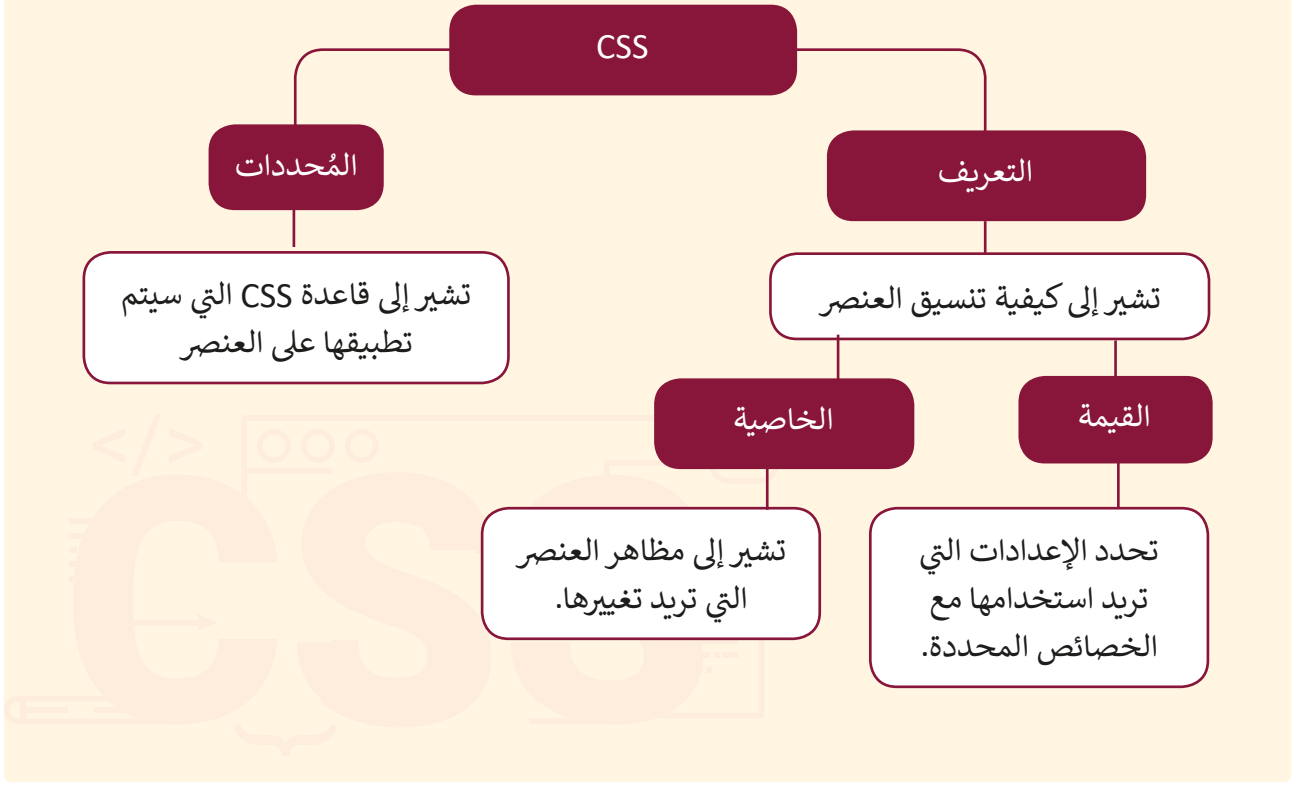
سهولة التعديل على صفحات الويب، بحيث يمكن التحكم في مظهر الموقع بأكمله بواسطة ملف CSS خارجي واحد، بل وإجراء التعديل على نمط صفحات الويب في الموقع من خلال تغيير لملف واحد فقط بدلاً من تحرير عدة نقاط في كل صفحة من صفحات الموقع.

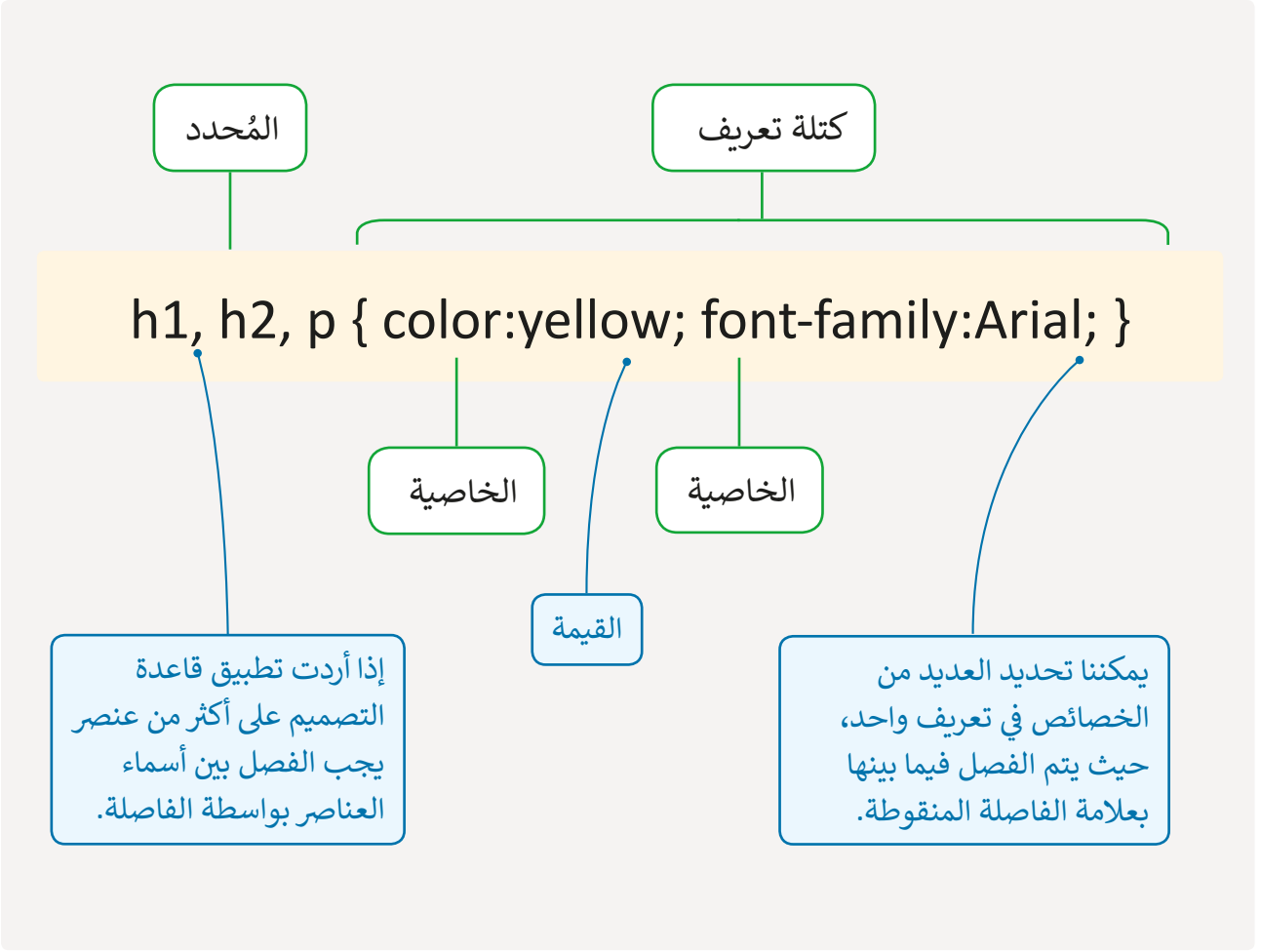
3 التنفيذ:

حجم أصغر للملف، حيث أن كل ما يتعلق بالتنسيق يكتب مرةً واحدة فقط للموقع بأكمله.

4 اختبار الموقع ونشره عبر شبكة الإنترنت:

تحميل أسرع للصفحات، فعند استخدام ملف CSS خارجي، يحفظ المتصفح صفحة من موقعنا في ذاكرة التخزين المؤقت، وبهذا لا يضطر إلى تنزيلها مرة أخرى كلما قام المستخدم بتنزيل صفحة أخرى من موقعنا.


CSS
المُحددات
التعريف
تشير إلى قاعدة CSS التي سيتم تطبيقها على العنصر
تشير إلى كيفية تنسيق العنصر
الخاصية
القيمة
تشير إلى مظاهر العنصر التي تريد تغييرها.
تحدد الإعدادات التي تريد استخدامها مع الخصائص المحددة.


المُحدد
كتلة تعريف
h1, h2, p { color:yellow; font-family:Arial; }
الخاصية
الخاصية
القيمة
إذا أردت تطبيق قاعدة التصميم على أكثر من عنصر يجب الفصل بين أسماء العناصر بواسطة الفاصلة.
يمكننا تحديد العديد من الخصائص في تعريف واحد، حيث يتم الفصل فيما بينها بعلامة الفاصلة المنقوطة.

طرق إدراج CSS داخل مستند HTML

نمط الإضافة في النسق نفسه Inline style

يستخدم لتطبيق نمط واحد لعنصر واحد فقط، ولاستخدامه فإننا نستخدم خاصية النمط style للعنصر ذو الصلة.

```
<p style="color:green;">
```

ورقة الأنماط الداخلية Internal style sheets

يتم استخدامها عندما يكون للمستند تنسيق منفصل، وهذا يطبق إذا أردنا تطبيق التنسيق على صفحة واحدة وليس على موقع الويب بكامله.

يتم تعريف الأنماط الداخلية داخل عنصر <style>، وداخل قسم <html> الموجود في صفحة HTML.

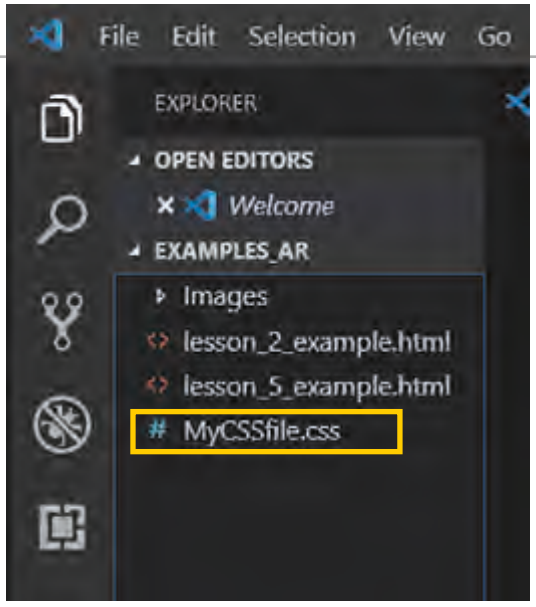
```
<html>
  <style>
    p {
        color: green;
        font-weight:bold;
        background-color:black;
    }
  </style>
</html>
```

ورقة الأنماط الخارجية External style sheets

تعتبر ورقة الأنماط الخارجية مثالية عندما يتم تطبيق النمط على العديد من الصفحات. يمكننا تغيير نمط الموقع بالكامل عن طريق تغيير ملف واحد فقط

يجب أن تكون كل صفحة مرتبطة بورقة الأنماط باستخدام الوسم <link> الموجود داخل قسم <html>.

بنفس الطريقة التي أنشأنا بها ملف **HTML**، يمكننا إنشاء ملف **CSS** بامتداد **css**. في الاسم.



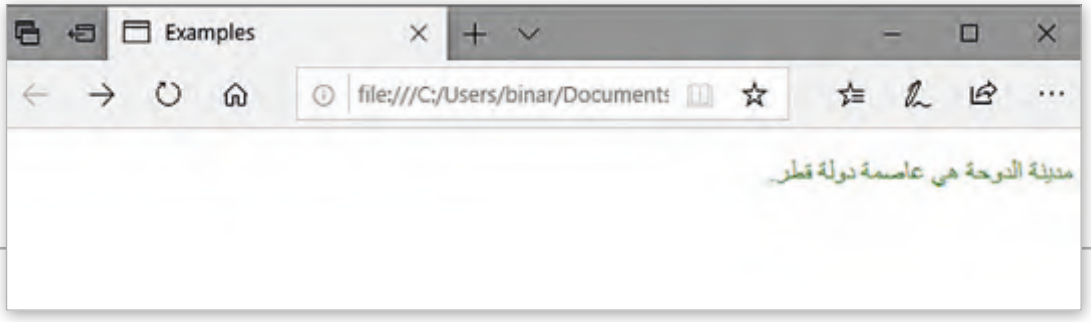
إنشاء رابط بين ملف HTML الحالي وورقة الأنماط الخارجية CSS.

.html

```
<html>
  <link rel="stylesheet" type="text/css" href="myCSSfile.css">
</html>
<body>
   <p>مدينة الدوحة هي عاصمة دولة قطر.</p>
</body>
```

.css

```
p{
   color: green;
}
```



محددات CSS

يمكننا اختيار الوسوم المراد تنسيقها من خلال محددات **CSS**، والتي هي أساسًا بعض "القواعد" التي تُمكن المتصفح من فهم ما نريد تطبيقه عند الكتابة في كل مرة. هناك العديد من محددات **CSS** التي توفر مرونة كبيرة.

مُحدد Type

عندما يحتوي المحدد على اسم لوسم HTML، سيتم تطبيق الخصائص الموجودة في قاعدة CSS على جميع عناصر HTML، على سبيل المثال، p تطبق على جميع عناصر صفحة الويب الموجودة بين وسمي ..... .

محدد id

لاختيار عنصر بمُعرف محدد، يتم استخدام رمز **(#)** متبوعا بمُعرف العنصر. مُحدد **id** يستخدم خاصية **id** الخاصة بعنصر **HTML** لاختيار عنصر محدد. على سبيل المثال إذا أردنا تطبيق نمط على عنصر **HTML** بمُعرف **"header" = id** سنقوم بكتابة **#header {}**.

مُحدد Class

كما نعلم فإن الفئة class تستخدم بشكل عام لتجميع بعض عناصر HTML التي لم يتم تطبيق أي مُحدد عليها، ولذلك فإننا نُعين class خاص بهذه العناصر حتى يتم الرجوع إليها في CSS ومن ثمَ تنسيقها.

هكذا يبدو موقع الويب الخاص بنا مكتوبًا بلغة HTML قبل تصميمه بواسطة CSS.

خصائص CSS الأساسية المرتبطة بتنسيق النص

- color < (اللون)
- font-size < (حجم الخط)
- font-family < (نوع الخط)
- font-style < (نمط الخط)
- font-weight < (عرض الخط)
- text-decoration < (زخرفة النص)

Color (اللون)

تتيح لنا خاصية **Color** تحديد لون النص داخل عنصر ما. يتم تحديد اللون في **CSS** بثلاث طرق:

- أسماء الألوان : يوجد 147 لونًا مُعرفًا مسبقًا، يتم التعرف عليها بواسطة المتصفحات.
- قيم RGB: درجة كل لون من الألوان الأحمر والأخضر والأزرق المستخدمة في تكوين اللون بحيث تتراوح كثافة كل لون ما بين 0 و 255. على سبيل المثال RGB(120,100,80).
- رموز Hex: رموز مكونة من ستة أرقام تمثل مقدار كل لون من الألوان الأحمر والأخضر والأزرق.

```
Hex: #ef7d00
RGB: (239,125,0)
```

.html

```
<html>
   <link rel="stylesheet" type="text/css" href="myCSSfile.css">
</html>
<body>
   <h1>قطر</h1>
   <p> أرضٌ تدركُ أن الزائرَ لا يشعرُ بالغربة فيها، بل صديقٌ لم تلتقِ به بعد. </p>
   <h2>الدوحة</h2>
   <p>مدينة الدوحة هي عاصمة دولة قطر.</p>
</body>
```

.css

```
h1 {
   color: green;}
h2 {
   color: #ee82ee; }
p {
   color: rgb(0, 0, 0); }
```

Font-size حجم الخط

خاصية حجم الخط تسمح لنا بتحديد حجم الخط الخاص بالنص. يمكننا تحديد حجم الخط إما بوحدة البكسل أو بالنسبة المئوية.

.css

```css
h1 {
   color: green;
   font-size: 30px; }
h2 {
   color: #ee82ee;
   font-size: 100%; }
p {
   color: rgb(0, 0, 0); }
```

font family (عائلة الخط)

تتيح لنا خاصية عائلة الخطوط تحديد عائلة الخط التي يجب استخدامها لأي نص داخل العناصر. يجب أن يتم تثبيت عائلة الخط التي تم تحديدها لاستخدامها في موقعنا، وكذلك يجب أن يتم تثبيتها على جهاز زوار الموقع. لهذا السبب، يتم تحديد قائمة الخطوط مفصولة بفواصل بحيث إذا لم يكن لدى المستخدم أول نوع خط مثبت فإن المتصفح سيستخدم النوع التالي.

إذا لم يكن أي من الخطوط المحددة موجود على جهاز الحاسوب الخاص بالمستخدم فسيختار المتصفح خطًا ينتمي إلى عائلة الخطوط العامة.

.css

```
h1 {
   color: green;
   font-size: 30px;
   font-family:Tahoma,sans-serif;}
h2 {
   color: #ee82ee;
   font-size: 100%;
   font-family:Tahoma,sans-serif; }
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif; }
```

إذا لم يوجد أي من الخطوط التي نحددها على حاسوب المستخدم، فعندئذٍ يحدد المتصفح واحدًا ينتمي إلى عائلة الخطوط العامة.

font-style (نمط الخط)

تسمح لنا خاصية نمط الخط بإنشاء نص مائل. تتعدد قيم هذه الخاصية ما بين أن يكون الخط **normal** (عادي)، **italic** أو **oblique** (مائل).

font-weight (عرض الخط)

تسمح لنا خاصية عرض الخط بإنشاء نص غامق، يوجد قيمتان لهذه الخاصية وهي إما **bold**: (غامق) أو **normal** (عادي).

.css

```
h1 {
   color: green;
   font-size: 30px;
   font-family: Tahoma,sans-serif;}
h2 {
   color: #ee82ee;
   font-size: 100%;
   font-style: italic;}
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif; }
```

text-decoration (زخرفة/تنميق النص)

تسمح لنا خاصية زخرفة النص بتطبيق العديد من التأثيرات على النص.

تأخذ هذه الخاصية القيم التالية:

- none > : يزيل أي زخرفة تم تطبيقها.
- underline > : يضيف خطًا أسفل النص.
- overline > : يضيف خطًا أعلى النص.
- line-through > : يضيف سطرًا أفقيًا عبر الكلمات.
- blink > : يقوم بإضافة وميض متحرك للنص.

.css

```
h1 {
   color: green;
   font-size: 30px;
   font-family:Tahoma,sans-serif;
   text-decoration: underline;}
h2 {
   color: #ee82ee;
   font-size: 100%;
   font-style: italic;}
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif; }
```

1

في مقطع HTML التالي، قم بإضافة قاعدة CSS ومقطع HTML المناسبين لتغيير لون كل خط من القائمة غير المرتبة إلى لون مختلف.

```
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
</head>
<body>
   <ul>
      <li>one</li>
      <li>two</li>
      <li>three</li>
      <li>four</li>
      <li>five</li>
      <li>six</li>
   </ul>
</body>
</html>
```

2

في مقطع HTML التالي، استخدم ورقة التصميم الداخلية (internal style sheet) لتحرير المقطع بحيث يتم تغيير لون جميع عناصر <p> إلى اللون الأحمر "red".

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>

</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p>This is a second paragraph.</p>

</body>
</html>
```

3

في مقطع HTML التالي، استخدم ورقة التصميم الداخلية (internal style sheet) لتحرير المقطع لتغيير لون جميع العناصر الموجودة بمعرف id = "para1" إلى اللون الأخضر "green"

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>

   </style>
</head>
<body>

   <h1>This is a Heading</h1>
   <p id="para1">This is a paragraph.</p>
   <p>This is a second paragraph.</p>

</body>
</html>
```

4

في مقطع HTML التالي، استخدم ورقة التصميم الداخلية (internal style sheet) لتحرير المقطع، ليتم تغيير لون جميع عناصر <p> و<h1> إلى "red" و"italic" و"20px"، ولون جميع العناصر تحت فئة "colortext" class إلى اللون البنفسجي "purple" ونسقها بتنسيق غامق "bold"، وأخيرًا قم بتجميع المقاطع البرمجية معًا لتصغير حجم المقطع الكلي.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
   </style>
</head>
<body>

   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p class="colortext">This is a second paragraph.</p>
   <p class="colortext">This is a third paragraph.</p>

</body>
</html>
```

5

استخدم ورقة التصميم الداخلية (internal style sheet) لتحرير المقطع ليتم تغيير لون جميع عناصر تحت فئة "class " إلى اللون البنفسجي "purple".

```
<!DOCTYPEhtml>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
   </style>
</head>
<body>

   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p class="colortext">This is a second paragraph.</p>
   <p class="colortext">This is a third paragraph.</p>

</body>
</html>
```

6

في مقطع HTML التالي، أضف ورقة تصميم خارجية برابط "mystyle.css"، ثم قم بتحرير المقطع البرمجي لتغيير لون جميع العناصر تحت فئة "colortext": "class إلى اللون البنفسجي "purple".

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   
</head>
<body>

   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p class="colortext">This is a second paragraph.</p>
   <p class="colortext">This is a third paragraph.</p>

</body>
</html>
```

خصائص CSS الأساسية المتعلقة بالخلفية

background-color > (لون الخلفية)	background-image > (صورة الخلفية)
background-repeat > (تكرار الخلفية)	background-position > (موضع الخلفية)

background-color (لون الخلفية)

تُستخدم هذه الخاصية لتعيين لون خلفية العناصر التي تم اختيارها وتحديدها بواسطة المُحدد. يمكن كتابة اللون بأي من التنسيقات الموجودة ضمن خاصية اللون.

.css

```
body {
    background-image: url("Images/pattern2.png");
    background-repeat: repeat;
}
h1 {
    color: green;
    font-size: 30px;
    font-family:Tahoma,sans-serif;
    text-decoration:underline;}
h2 {
    color: #ee82ee;
    font-size: 100%;
    font-family:Tahoma,sans-serif;
    font-style: italic;
}
p {
    color: rgb(0, 0, 0);
    font-family: Arial, Times, sans-serif;
    background-color:rgb(230, 226, 247);
}
```

background-image (صورة الخلفية)

تسمح لنا هذه الخاصية بتعيين صورة كخلفية من خلال استخدام قيمة (image_address url). على سبيل المثال "background-image: url("images/doha.jpg". بشكل عام، عندما ندخل URL كقيمة لخاصية CSS، فإننا نكتبها بين القوسين الخاصين برابط url.

background-repeat (تكرار الخلفية)

قد تحتوي هذه الخاصية على قيم مختلفة، منها:

repeat: يتم تكرار الخلفية أفقياً وعموديًا بشكل افتراضي، بحيث يتم ملء المربع الموجود بكامله.

repeat-x: يتم تكرار صورة الخلفية أفقياً فقط.

repeat-y: يتم تكرار صورة الخلفية عموديًا فقط.

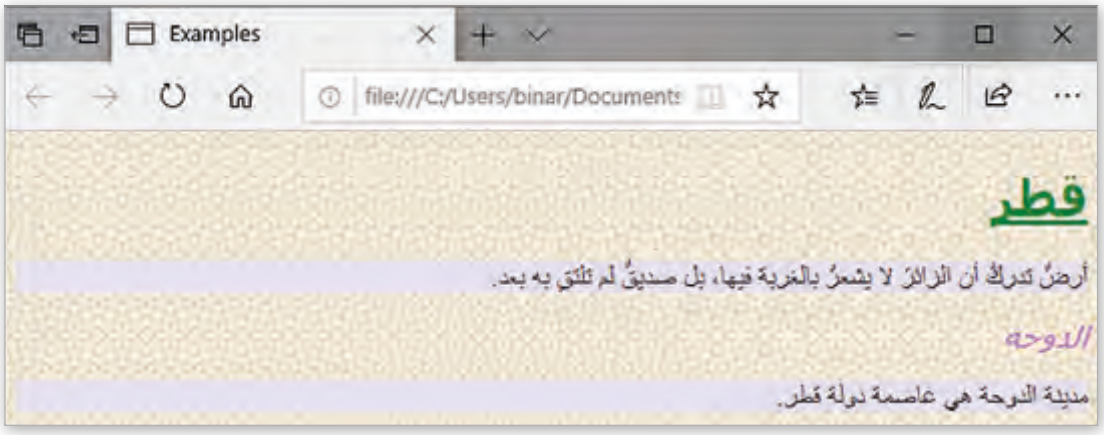
no-repeat: يتم عرض صورة الخلفية فقط.

fixed: تبقى الصورة في نفس الموضع في الصفحة.

scroll: تتحرك الصورة لأعلى ولأسفل أثناء قيام المستخدم بالتمرير.

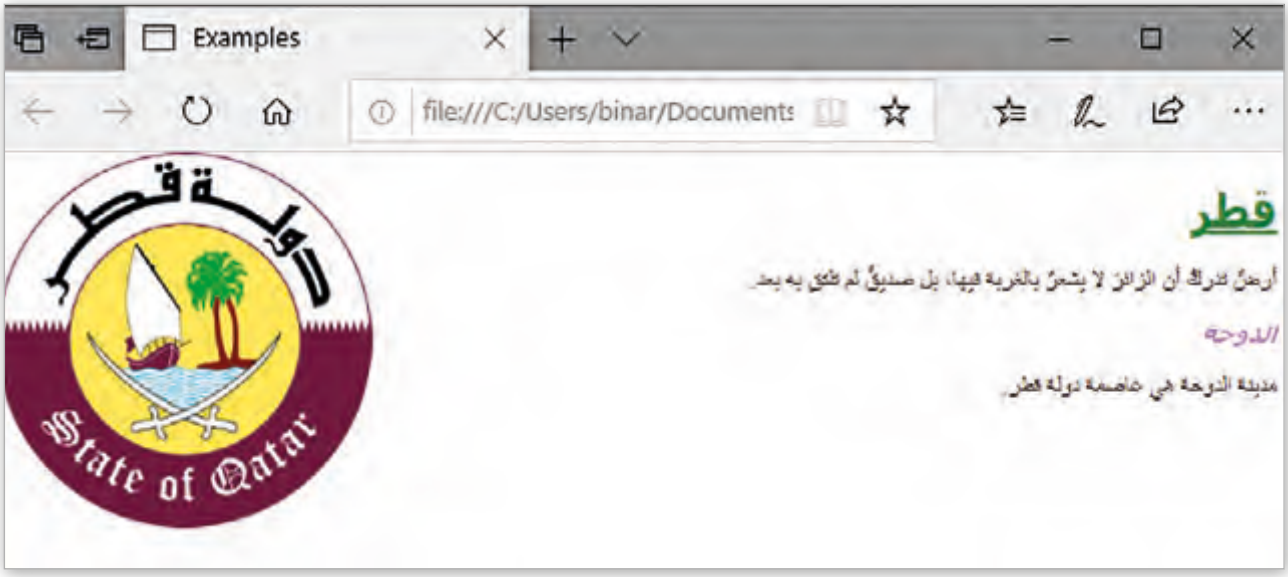
.css

```css
body {
    background-image: url("Images/pattern2.png");
    background-repeat: repeat;
}
```



.css

```
body {
   background-image:url("/images/Emblem_of_Qatar.png");
   background-repeat: no-repeat;
}
h1 {
   color: green;
   font-size: 30px;
   font-family:Tahoma,sans-serif;
   text-decoration:underline;
}
h2 {
   color: #ee82ee;
   font-size: 100%;
   font-family:Tahoma,sans-serif;
   font-style: italic;
}
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif;
}
```



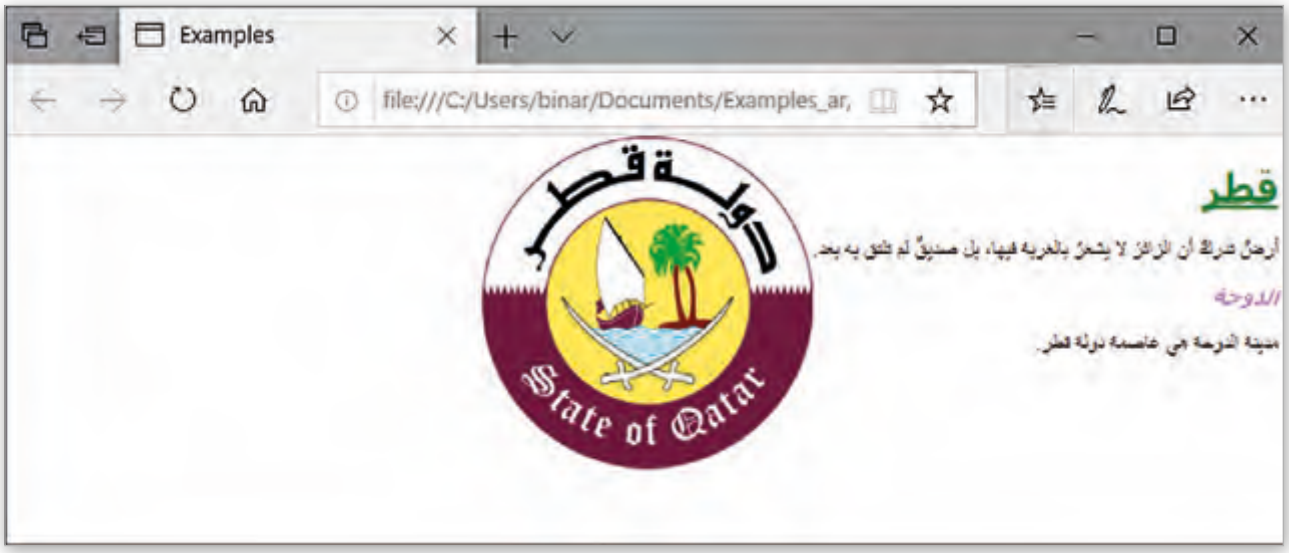
background-position (موضع الخلفية)

تتيح لنا هذه الخاصية تحديد موضع الصورة في نافذة المتصفح، وتوجد عدة قيم لهذه الخاصية:

- left top > (أعلى اليسار)
- left center > (مركز اليسار)
- left bottom > (أسفل اليسار)
- center top > (أعلى الوسط)
- center center > (مركز الوسط)
- center bottom > (أسفل المركز)
- right top > (أعلى اليمين)
- right center > (مركز اليمين)
- right bottom > (أسفل اليمين)

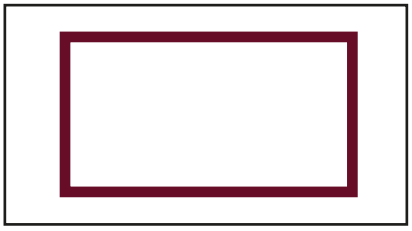
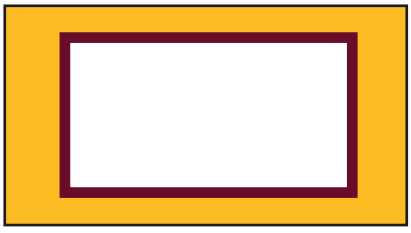
.css

```css
body {
   background-image:url("images/Emblem_of_Qatar.png");
   background-repeat: no-repeat;
   background-position:center top;
}
h1 {
   color: green;
   font-size: 30px;
   font-family:Tahoma,sans-serif;
   text-decoration:underline;
}
h2 {
   color: #ee82ee;
   font-size: 100%;
   font-family:Tahoma,sans-serif;
   font-style: italic;
}
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif;
}
```



نموذج الصندوق Box-Model

تتعامل **CSS** مع كل عنصر في **HTML** كما لو كان له ما يشبه صندوقه الخاص، مما يسمح لنا بالتحكم في الأبعاد والحدود والفراغ حول كل صندوق على حدة.

الخصائص		
الخاصية	الوصف	مثال
الإطار (Border)	لكل صندوق حدود سواء كان مرئيًا أم لا، بحيث يفصل هذا الإطار بين حافة كل صندوق عن الآخر.	
الهامش (margin)	تأتي الهوامش خارج الحدود.	
الفراغ (Padding)	الفراغ هو المسافة ما بين الإطار والمحتوى.	

يمكننا تحديد القيم الخاصة بكل جانب من الفراغ المحيط بالصندوق:

الفراغ (Padding)

- padding-top > (الفراغ العلوي)
- padding-right > (الفراغ على الجهة اليمنى)
- padding-bottom >(الفراغ في الأسفل)
- padding-left > (الفراغ على الجهة اليسرى)

يمكننا أيضًا الاختصار بكتابة القيم مع عقارب الساعة، (الأيسر، الأسفل، اليمين، الأعلى).

Padding: 15px 10 px 15 px 10 px

يمكن تحديد نفس القيم لخاصية الهامش **(margin)** أيضًا، وبنفس الطريقة المختصرة.

.css

```
p {
   color: rgb(0, 0, 0);
   font-family: Arial, Times, sans-serif;
   background-color:rgb(230, 226, 247);
   border:2px solid green;
   padding:5px 6px 5px 6px;
   margin:10px 15px 10px 15p;
}
```

الاختصار الخاص بالحدود Borders كالتالي (اللون، النمط، العرض).

أنماط التصميم

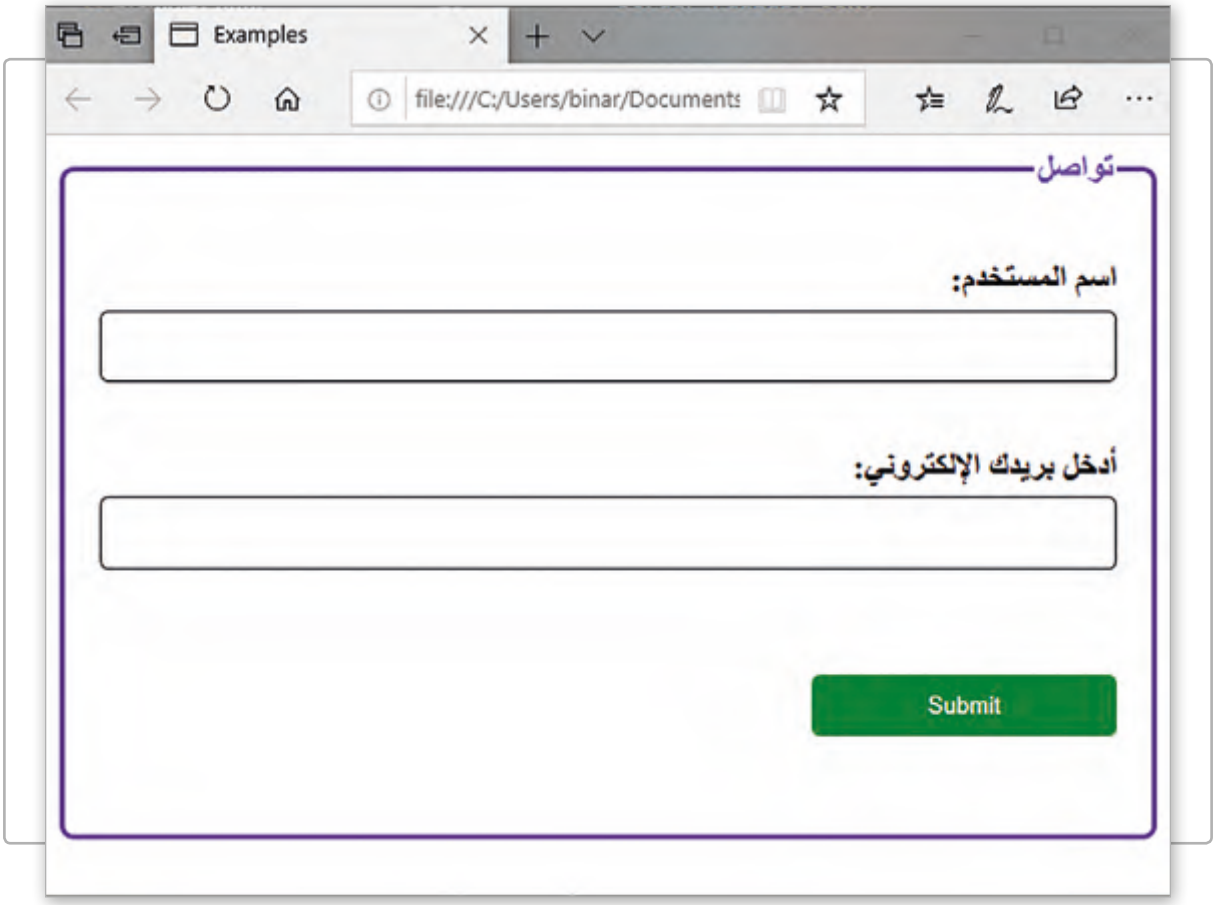
تعتبر خصائص الحدود، والهوامش، والفراغات المحيطة مفيدة جدًا في تصميم النموذج، ومن الشائع استخدامها في النماذج في تصميم المدخلات النصية **Text Inputs** ومساحات النص **Text Areas** وعناصر التسمية **Labels** وأزرار الإرسال **Submit Buttons**.

فلنستعرض هنا بعض الأمثلة:

.css

```
p{
   font-weight:bold;
   font-size:20px;}
input {
   width: 100%;
   padding: 12px;
   border: 1px solid black;
   border-radius: 5px;
   box-sizing: border-box;
   margin-top: 6px;
   margin-bottom: 16px;
   resize: vertical;}
input [type=submit] {
   width:30%;
   background-color: green;
   color: white;
   padding: 10px 20px;
   border:none;
   border-radius: 5px;
   cursor: pointer;
   text-align: center; }
fieldset{
   border:3px solid rgb(99, 52, 143);
   border-radius:8px;
   padding:20px;}
legend{
   font-weight:bold;
   font-size:20px;
   color:blueviolet; }
```

نمط مؤشر الفأرة الذي سيتم عرضه للمستخدمين.



لتغيير مظهر الزر عند مرور مؤشر الفأرة فوقه.

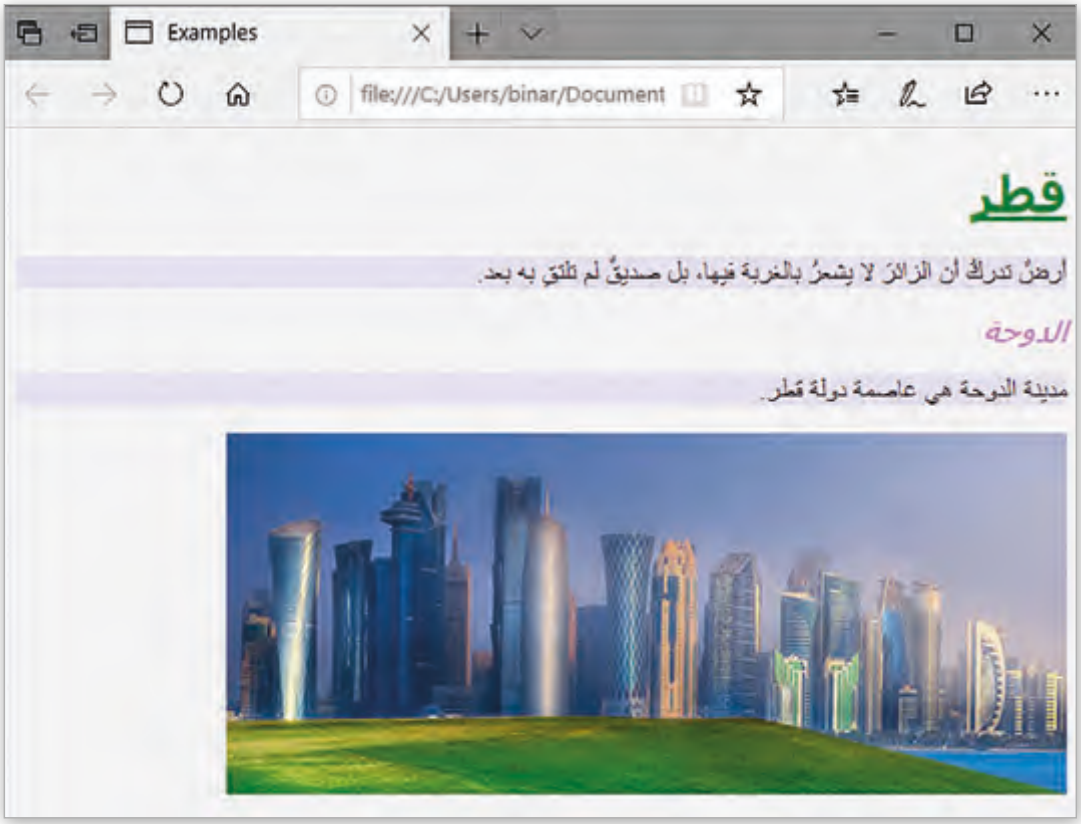
.css

```
input [type=submit]:hover {
   background-color:blueviolet;
   color:black;
}
```

التحكم في حجم ومحاذاة الصور

يمكننا باستخدام قواعد **CSS** تحديد حجم ومحاذاة الصور، كما يمكننا جعلها أكثر جاذبية باستخدام بعض الخصائص الأخرى.

فلنستعرض صفحة الويب التالية:

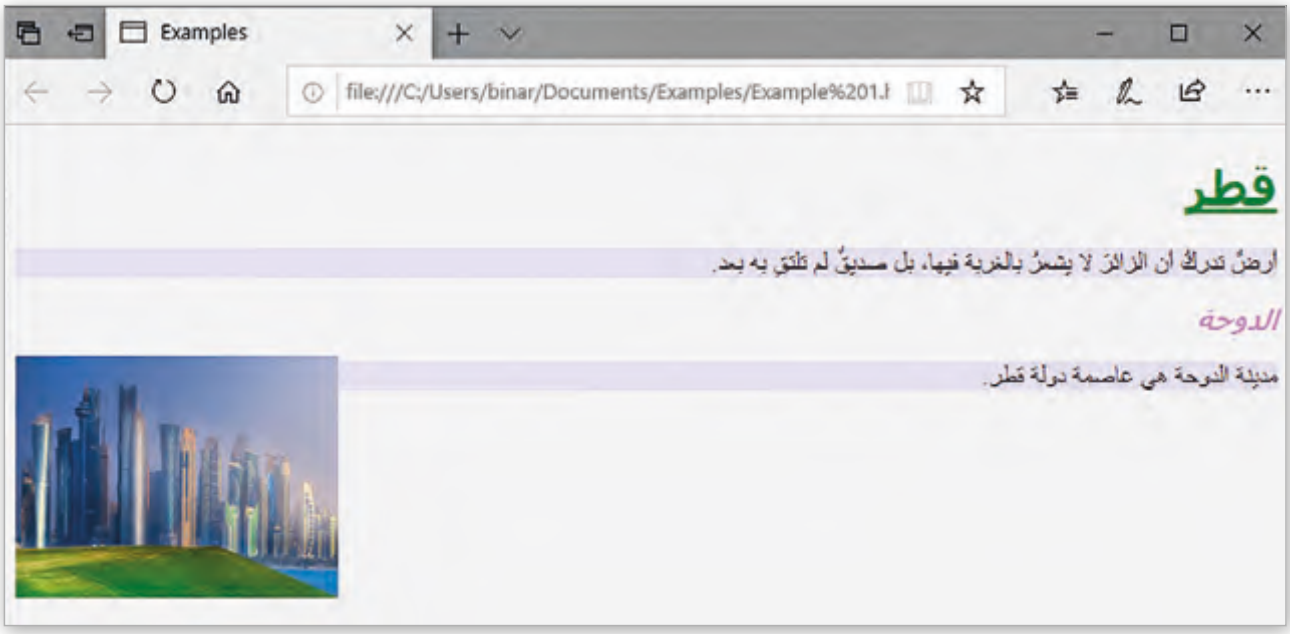


سنطبق الآن بعض قواعد **CSS** على الصور داخل الصفحة.

.css

```
img {
   width:200px;
   height:150px;
   float:left;
}
```

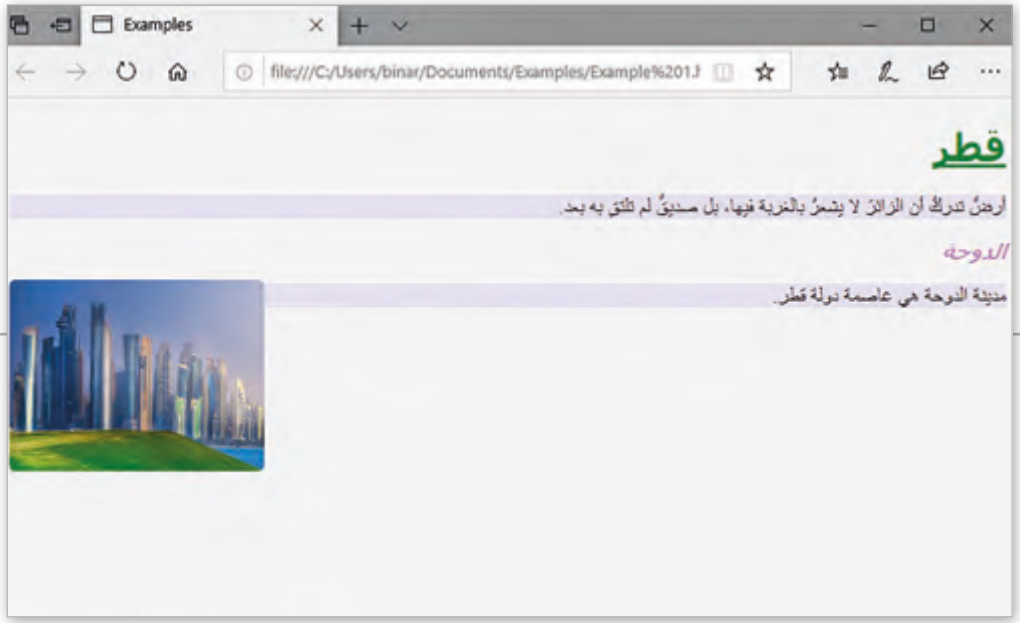
خاصية float تسمح لنا بأخذ العنصر من سياقه الطبيعي ووضعه إلى يسار أو يمين العنصر المحتوى إن أمكن ذلك.



يمكننا استخدام خاصية **border-radius** أيضًا لتغيير مظهر حواف الصورة.

.css

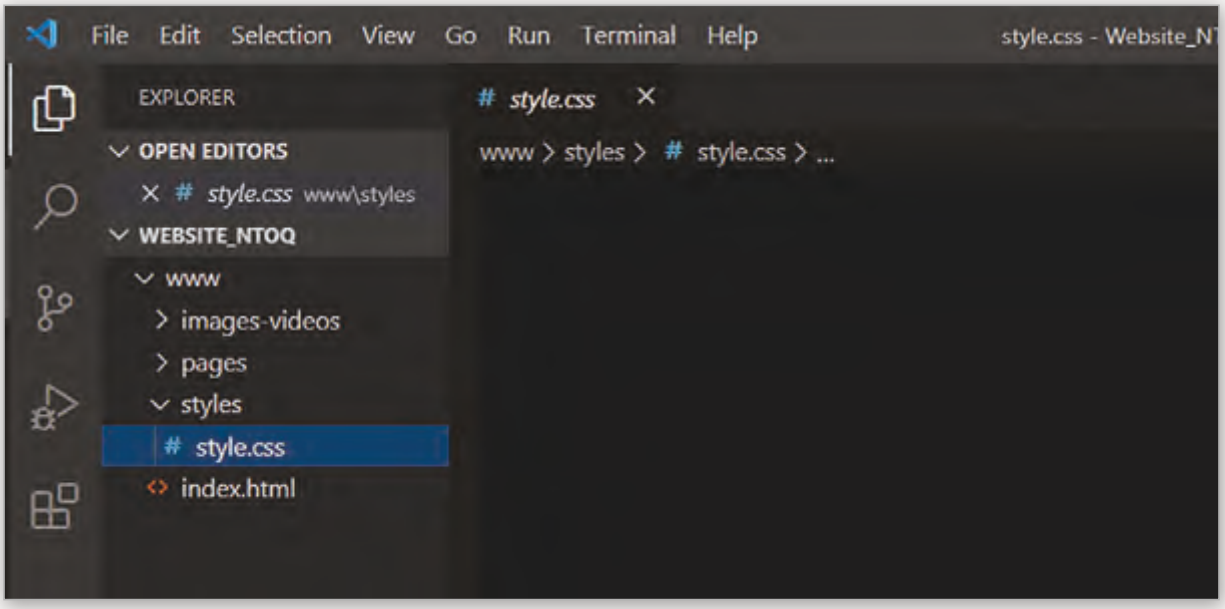
```
img {
  width:200px;
  height:150px;
  float:left;
  border-radius:5px;
}
```



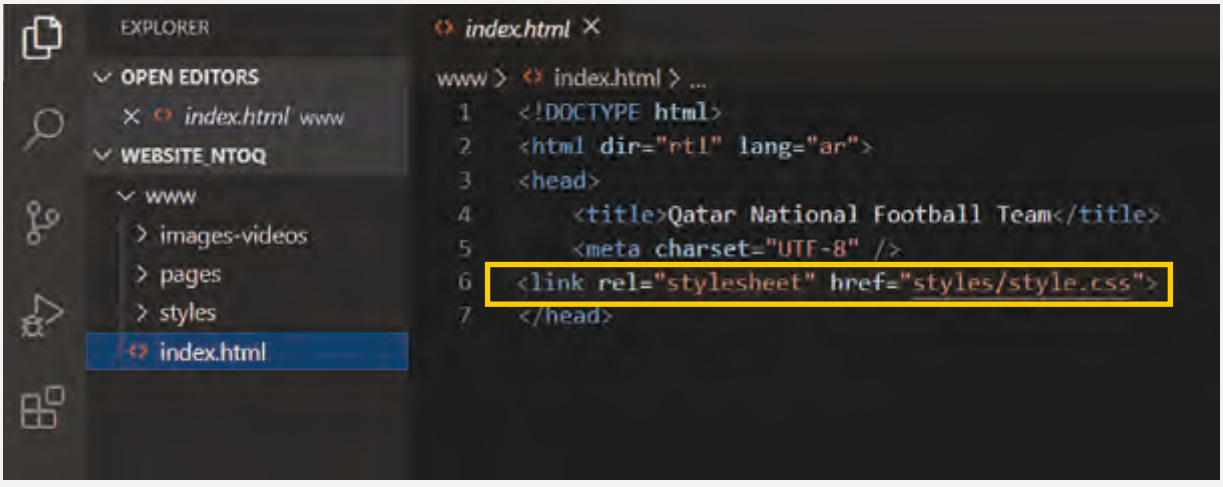
الآن وبعد أن تعلمنا الصيغة العامة لـ CSS وقواعدها، سنستخدمها في إثراء موقعنا بتصميم مناسب،

أيضًا سنتبع بعض مبادئ التصميم (على شكل نصائح) خاصة بتطوير الويب والتي سترشدنا لإنشاء تجربة مستخدم تتضمن النواحي الجمالية والوظيفية للموقع.

في البداية سنقوم بإنشاء تصميم CSS في مجلد الصفحات في محرر Visual Studio Code باسم: style.css،



وبعد ذلك سنقوم بإنشاء مرجع إلى ورقة التصميم stylesheet الموجودة في كود HTML باستخدام رابط لوسم يوضع داخل لبنة html البرمجية الرئيسة.



الخطوات الأساسية الأكثر أهمية هي تعيين نوع خط عام إلى المحتوي النصي الذي لم يتم تطبيق أي تصميم عليه.

يجب عليك استخدام نوع خط سهل للقراءة ومريـح للعين.

.css

```
body {
   margin:0;
   font-family: 'Open Sans', sans-serif;
   background:white;
}
h1,h2 {
   font-family: Arial, Helvetica, sans-serif;
}
```

الخطوات التالية هي إضافة تصميم إلى **header** ووسوم **h1** الرئيسة أثناء ضبط موضع كل عنصر مرتبط بالآخر، وذلك باستخدام خواص نموذج الصندوق **BOX – Model**.

يجب الابتعاد عن استخدام أرقام كبيرة عند تغيير خصائص نموذج الصندوق BOX-Model، حيث أنها من الممكن أن تتسبب في تشويه مظهر صفحات الويب.

.css

```
/*Styling the header*/
.header {
  background-image: url(../Images_Videos/Team_Qatar.jpg);
  background-size: cover;
}
h1.header_title {
  text-align: center;
  font-size:96px;
  color:rgb(73, 3, 3);
}
```

عملية تصميم شريط التصفح مهمة جدًا نظرًا لدورها في تسهيل استخدام موقع الويب الخاص بنا.

.css

```css
/*Styling the navigation bar*/
.nav_bar {
  width: 100%;
  text-align: center;
  margin-top: 0;
  background-color:whitesmoke;
  position: fixed;
  top: 0;
}
.nav_bar ul {
  list-style-type: none;
  margin: 0;
  padding: 20px;
  overflow: hidden;
  background-color: none;
}
.nav_bar li {
  display:inline;
  float:none;
}
```

تقوم هذه الخاصية بإخفاء المحتوى الإضافي إذا كان يحتاج إلى مساحة أكبر من مساحة الإطار المحدد له.

يحدد طريقة عرض عنصر ما.

تنسيق شريط التصفح

سنقوم الآن بتنسيق العناصر الموجودة في شريط التصفح

.css

```
/* Styling the links inside the navigation bar */
.nav_bar a {
  float:center;
  text-align:center;
  font-size:20px;
  padding:12px 14px;
  color:rgb(73, 3, 3);
  font-weight:bold;
  text-decoration: none;
}
```

ضبط أجزاء الصفحة

سنقوم الآن بضبط مواضع باقي أجزاء الصفحة باتباع نفس القواعد التي أشرنا إليها بالأعلى.

.css

```
/*Styling the History section*/
.history {
  margin-left:100px;
  margin-right: 100px;
}
.history h1 {
  text-align: center;
  color:rgb(26, 102, 26);
  font-size:50px
}
```

تنسيق ألبوم الصور

الآن سنقوم بتنسيق قسم ألبوم الصور.

.css

```
/*Styling the Gallery section*/
.gallery{
   background:#e0e0e0;
}
.gallery h1{
   text-align: center;
   color:rgb(26, 102, 26);
   font-size:50px
}
.photos img {
   width: 100%;
   height: auto;
}
.gallery_centertext p{
   text-align: center;
}
.ResponsiveImage{
   padding:6px;
   float: left;
   width: 24.99999%;
}
```

تنسيق قسم "من نحن"

ثم سنقوم بتنسيق قسم **About**.

.css

```css
/*Styling the About section*/
.about{
  background:rgb(224, 224, 224);
}
.about_header{
  margin-left:100px;
  margin-right: 100px;
  padding-bottom: 10px;
}
.about_header h1{
  text-align: center;
  color:rgb(26, 102, 26);
  font-size:50px;
}

.left_logo{
  float: left;
  margin: 15px 35px 0px 0px;
}
```

فلنرتب موضع التذييل **footer**، ولنصمم مظهر الروابط الخارجية الخاصة بموقع الويب.

.css

```
/*Styling the footer*/
.footer {
  background:black;
  padding:20px;
}
.footer h2 {
  color:white;
}
.footer ul {
  list-style-type:none;
}
.info_list li a {
  color: #fff;
  text-decoration: none;
}
```

يجب علينا أيضًا ضبط المرجع الخاص بأيقونات التصميم الموجودة في كود **HTML** الخاص بنا أيضًا.

```
<div class="footer">
  <h2>الاتحاد القطري لكرة القدم</h2>
  <ul class="info_list">
    <li><a href="mailto:info@example.com">info@example.com</a></
    li>
    <li><a href="https://www.qfa.qa/first-national-team/">قم بزيارة موقع
    الاتحاد القطري لكرة القدم الإلكتروني </a></li>
  </ul>
  <ul class="social_icons">
    <li><a href="https://www.facebook.com/QFA/" class="fa fa-
    facebook"></a></li>
    <li><a href="https://twitter.com/qfa?lang=el" class="fa fa-
    twitter"></a></li>
    <li><a href="https://www.instagram.com/qfa/" class="fa fa-
    instagram"></a></li>
  </ul>
</div>
```

سيتم أخذ أيقونات وسائل التواصل الاجتماعي من مكتبة خارجية.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<html>
  <title>Qatar National Football Team</title>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Add icon library -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
  awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="styles/style.css">
```

لنمنح أيقونات وسائل التواصل الاجتماعي تصميمًا ما.

.css

```
/*Styling the social icons*/
.social_icons li {
  display: inline-block;
}
.social_icons li a {
  padding:20px;
  display: block;
  color: #fff;
  text-decoration:none;
  text-align:center;
  font-size:40px;
}
.fa-facebook:hover {
  background:#3B5998;
}
.fa-twitter:hover {
  background: #55ACEE;
}
.fa-instagram:hover {
  background:#833ab4;
}
```

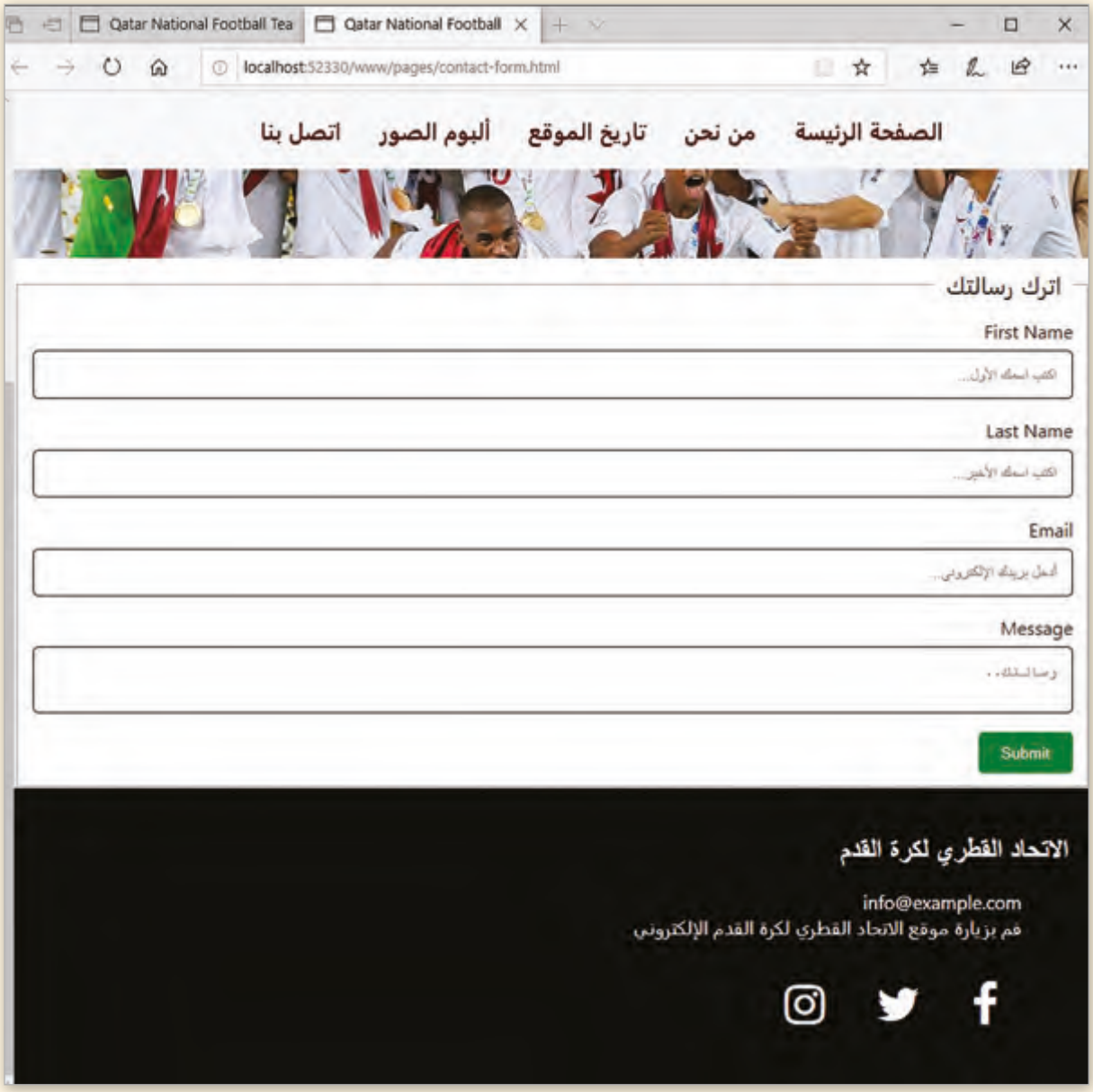
إن نمط تصميم أيقونات وسائل التواصل الاجتماعي مهم جدًا في تقييم شعبية موقع الويب الخاص بنا، لأن زائري الموقع قد يستخدمون تلك الأيقونات كمراجع تخص الموقع في المستقبل.

قم بإنشاء روابط إلى مواقع الويب الموثوقة. يجب علينا مشاركة محتوى الآخرين من خلال موقعنا، لكي يتم الربط إلى موقعنا من قبل الآخرين.

سنستخدم الآن نفس المبادئ وقواعد التصميم في نموذج اتصال موقع الويب الذي تم ربطه بالصفحة الرئيسة.

.css

```css
/*Styling the form*/
legend{
  padding:10px;
  font-size: 20px;
}
input [type=text], textarea {
  width: 100%;
  padding: 12px;
  border: 1px solid black;
  border-radius: 5px;
  margin-top: 6px;
  margin-bottom: 16px;
  resize: vertical;
}
input [type=email], textarea {
  width: 100%;
  padding: 12px;
  border: 1px solid black;
  border-radius: 5px;
  margin-top: 6px;
  margin-bottom: 16px;
  resize: vertical;
}
input [type=submit] {
  background-color: green;
  color: white;
  padding: 10px 20px;
  border:none;
  border-radius: 5px;
  cursor: pointer;
  text-align: center;
}
input [type=send]:hover {
  background-color:rgb(47, 161, 255);
  color:black
}
```


Qatar National Football Tea
Qatar National Football
localhost:52330/www/pages/contact-form.html
الصفحة الرئيسة
من نحن
تاريخ الموقع
ألبوم الصور
اتصل بنا
اترك رسالتك
First Name
Last Name
Email
Message
Submit
الاتحاد القطري لكرة القدم
info@example.com
قم بزيارة موقع الاتحاد القطري لكرة القدم الإلكتروني

1

قم بعمل التغييرات الضرورية لضبط لون خلفية الصفحة "background-color" باللون الزهري "pink". استخدم (Inline style).

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
</head>


<body>
   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p>This is a second paragraph.</p>
</body>

</html>
```

2

استخدم خاصية الخلفية "shorthand" في اختيار خلفية من اختيارك، قم بعرضها مرةً واحدة في الزاوية العلوية اليمنى من الصفحة.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>

</head>
<body>
   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p>This is a second paragraph.</p>
</body>

</html>
```

3

قم بتحرير المقطع التالي لجعل جميع الهوامش margins الخاصة بـ <h1> تصبح "20px".

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
      h1 {
         background-color: light-grey;

      }
   </style>
</head>
<body>

   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>

</body>
</html>
```

4

قم بتحرير المقطع التالي واستخدم خاصية ضبط الحدود بحيث تكون قيمة p إلى "5px"، "solid"" وبلون أزرق "blue".

```
<!DOCTYPEhtml>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
      h1 {
         background-color: light-grey;

      }
   </style>
</head>
<body>

   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>

</body>
</html>
```

5

قم بتحرير المقطع التالي بحيث يتم ضبط لون الروابط إلى اللون الأخضر "green"، مع تحويله إلى اللون البنفسجي "purple" عند تمرير مؤشر الفأرة فوقه.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
      
   </style>
</head>
<body>
   <h1>This is a Heading</h1>
   <p>This is a paragraph.</p>
   <p>This is a second paragraph.</p>
   <p><a href="https://www.visitqatar.qa/ar">This is a
   link</a></p>
</body>

</html>
```

6

قم بإنشاء موقع ويب خاص بوصفات الطعام حسب التعليمات التالية:

سيكون لكل وصفة اسم، وقائمة من المكونات، وقسم خاص بالتعليمات .

- قم بإنشاء صفحة خاصة بوصفة لعمل شطيرة صحية من الخضار والدجاج.

في هذه الصفحة يجب عرض المكونات كقائمة غير مرتبة دون نقاط تعداد.

قم بعرض مكونات اللحوم بخلفية حمراء فاتحة، ومكونات الخضار بخلفية بلون أخضر فاتح، ومنتجات الألبان بخلفية بلون أصفر فاتح.

- قم باختيار الخط من نوع sans serif للعنوان، أما تعليمات الوصفة فيجب أن تكون تحت عنوان الترويسة "Instructions" بخط مائل "italic".

7

افتح المجلد الخاص بموقع السفر.

سنبدأ عملية تصميم الموقع، حيث سنقوم بتقسيم صفحة الويب إلى عدة أجزاء.

ستقوم بتطبيق خصائص CSS الأساسية لتنسيق النص كما تعلمت سابقاً لجعل موقع الويب أكثر جاذبية.

قم بتغيير خلفية الموقع من خلال استخدام صورة كخلفية، أو تلوين كل قسم بلون مختلف.

في النهاية قم بحفظ عملك.

مشروع الوحدة

العنوان: إنشاء موقع ويب تعريفي لشخصية من اختيارك

الوصف: يهدف هذا المشروع إلى إنشاء موقع ويب خاص بإحدى الشخصيات البارزة كأحد العلماء أو الرياضيين أو المؤثرين، سيتضمن هذا المشروع عرض بعض المحتوى عن هذه الشخصية، كالأعمال و الصور والنصوص التي تقدم الشخصية لزوار موقعك.

الأدوات: محرر Visual Studio Code

خطوات التنفيذ:

أنشئ مجلدًا في مستنداتك باسم "Tribute_website".

افتح محرر Visual Studio Code ثم افتح المجلد الذي أنشأته.

> أنشئ ملفي HTML، الأول سيعتبر بمثابة الصفحة الرئيسة والآخر سيحتوي على نموذج form.

> أضف وسوم HTML المناسبة والفقرات إلى موقعك.

> أنشئ قائمة غير مرتبة بمثابة شريط تصفح الموقع، ثم قم بإضافة العناصر المناسبة داخلها.

> قم بربط علامات تبويب شريط التصفح إلى الصفحة أو الأقسام التي تريدها.

> أضف بعض الصور للشخصية التي ستقدمها في موقعك.

> أنشئ نموذج اتصال يتيح لزائري الموقع الاتصال بك.

> قم بتقسيم الصفحة لتتمكن من تصميمها بشكل ميسر.

> طبق قواعد CSS التي تعلمتها لتصميم موقع الويب الخاص بك لجعله أكثر جاذبية.

> احفظ عملك.

ماذا تعلمت

تعلمت في هذه الوحدة:

> استخدام محرر Visual Studio Code لإنشاء موقع ويب.
> استخدام وسوم HTML وCSS لتنسيق صفحة الويب.
> إضافة نموذج HTML إلى صفحة الويب.

المصطلحات:

الدرس 1	الملف File	لغة ترميز النصوص التشعبية HTML	أوراق التنسيق المتتالية CSS
	خاصية Attribute	جافا سكربت JavaScript	موقع الويب Website
الدرس 2	فقرة Paragraph	الترويسة Heading	وسم Tag
	قائمة غير مرتبة Unordered List	قائمة مرتبة Ordered List	رابط Link
	مسار Path	تمييز Identify	نموذج Form
الدرس 3	التصميم الداخلي Internal Style	ورقة التصميم الخارجية External Stylesheet	فئة الخط Font-family
	حجم الخط Font-size	سُمك الخط Font-weight	نمط الخط Font-style
	زخرفة الخط Text-decoration	الخلفية Background	موضع الخلفية Background-position
الدرس 4	نموذج الصندوق Box model	الحاشية Padding	الهامش Margin

2. تطوير مواقع الويب II

سنتعرف في هذه الوحدة على المقصود بوسوم CSS الخاصة بتنسيق صفحات الويب بحيث تكون متجاوبة، ويتم عرضها بشكل مناسب على أجهزة الحواسيب المكتبية والأجهزة المحمولة، كما سنتعلم أيضًا كيفية حفظ بيانات موجودة في نموذج بلغة HTML إلى قاعدة بيانات، وأخيرًا سنتعرف على المقصود بمصطلح تحسين محركات البحث SEO وكيفية استخدامه من أجل رفع ترتيب الموقع وزيادة عدد زائريه.

ماذا سنتعلمْ؟

في هذه الوحدة سنتعلم:

> تطبيق التغييرات اللازمة لتحويل موقع الويب إلى موقع تفاعلي.

> إضافة مقاطع JavaScript البرمجية لإنشاء موقع تفاعلي.

> المقصود بتحسين محرك البحث SEO، وكيف يمكن القيام بذلك.

> ما المقصود بأجهزة الخوادم Servers وأنواعها.

> ما المقصود بقاعدة البيانات ونظم إدارة قاعدة البيانات DBMS.

> أوجه الاختلاف بين مواقع الويب الساكنة والتفاعلية.

> ما المقصود بالبرمجة النصية المعتمدة على الخوادم (Server-Side).

> استخدام البرمجة النصية المعتمدة على الخوادم لحفظ البيانات في قاعدة بيانات.

مواضيع الوحدة

> التصميم المتجاوب

> موقع الويب التفاعلي

> المعالجة المعتمدة على الخادم

الأدوات

> محرر Visual Studio Code.

هل تذكر؟

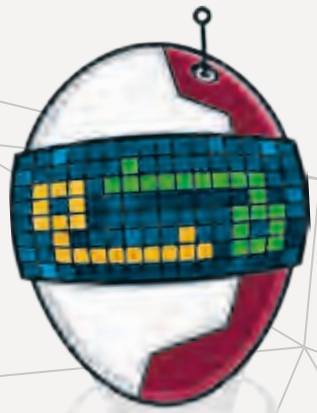
شبكة الخادم/العميل Client-Server model

تحتوي الشبكة على جهازي حاسوب أو أكثر يتبادلان الاتصالات معًا، وكلما زاد حجم الشبكة كلما زادت المهام والأعباء المنوطة بها. لكي يتم تجزئة مهام وأعباء العمل فعلى كل حاسوب داخل الشبكة أن يقوم بدور محدد.

شبكة الخادم/العميل يعمل كتطبيق موزع Distributed Application، حيث يتم التعامل مع بعض الأجهزة كخوادم والأخرى كعملاء.

على سبيل المثال، افترض أننا لدينا متجر لديه موظفون وعملاء، حيث يقوم الموظفون بتلبية طلبات العملاء. يشبه هذا تمامًا ما يحدث في نموذج الخادم/ العميل، حيث تقدم أجهزة الخوادم الخدمات التي تلبي طلبات أجهزة العملاء.

الخدمات مثل البريد الإلكتروني والوصول للويب هي خدمات موجودة بالفعل في شبكة الخادم/العميل، عندما نستخدم الإنترنت فإن حواسيبنا تستخدم المتصفحات لإرسال الطلبات إلى خادم الويب، والذي بدوره يستجيب ويرسل البيانات الصحيحة.



شبكة الخادم/العميل هو من أكثر الشبكات شيوعًا، يمكننا رؤية هذه الشبكة في كل مكان تقريباً، في المدارس، وفي البنوك وما إلى ذلك.

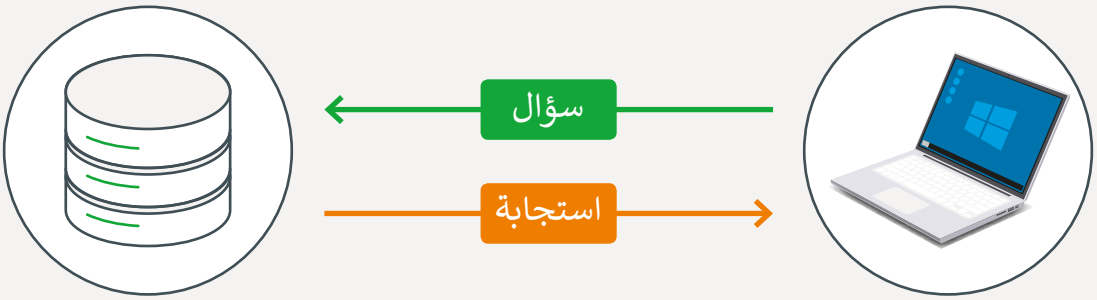
شبكات P2P

في شبكة P2P (نظير إلى نظير)، يعتبر "النظراء" أنظمة حاسوب متصلة ببعضها البعض عبر الإنترنت. يمكن مشاركة الملفات مباشرة بين الأنظمة على الشبكة دون الحاجة إلى خادم مركزي. بمعنى آخر، يصبح كل حاسوب على شبكة P2P خادم ملفات وعميل بنفس الوقت.

يتيح لنا الإنترنت تبادل المعلومات حول العالم في ثوانٍ، عندما نقوم بتوصيل جهاز الحاسوب الخاص بنا بالإنترنت، فإننا نقوم بالاتصال بنوع خاص من الخوادم يسمى مزود خدمة الإنترنت (ISP)، والذي يوفر خدمة الإنترنت رابطًا بين الحاسوب والعالم الخارجي (الإنترنت).

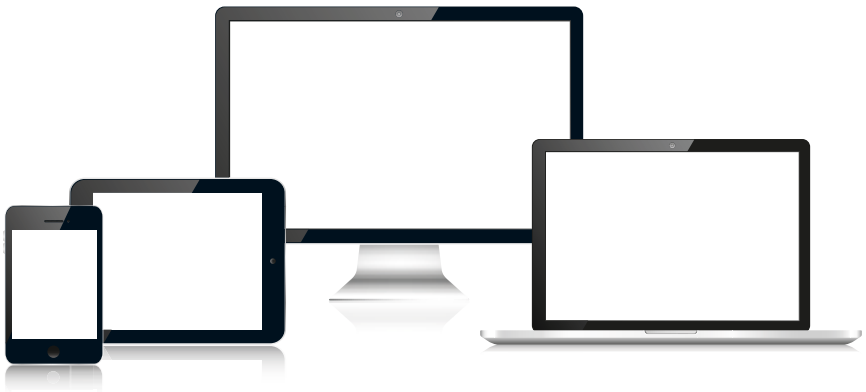
عندما نريد عرض صفحات الويب أو التحقق من البريد الإلكتروني، يرسل جهاز الحاسوب الخاص بنا الطلبات إلى خادم ISP (مزود خدمة الإنترنت)، الذي يتصل بدوره بخوادم الإنترنت الأخرى ويعيد توجيه الطلبات. يسمى هذا النوع من خوادم الإنترنت بخادم الويب Web Server.

إذا أردت عرض صفحة ويب في المتصفح الخاص بك:	
1.	اكتب عنوان موقع الويب في شريط عنوان المتصفح.
2.	سيرسل المتصفح طلبًا إلى خادم ISP يطلب منه الصفحة.
3.	سيبحث خادم ISP في قاعدة بيانات ضخمة من عناوين مواقع الإنترنت ويعثر على خادم الويب الصحيح الذي يستضيف موقع الويب، ثم سيرسل طلب هذه الصفحة من خادم الويب الخاص بها.
4.	سيرسل خادم الويب الصفحة المطلوبة إلى خادم ISP الخاص بك.
5.	سيرسل خادم ISP الصفحة إلى متصفحك وسيتم عرضها على شاشتك.



الدرس الأول

التصميم المتجاوب



موقع الويب المتجاوب Responsive Website

التصميم المتجاوب هو أحد طرق التصميم والتطوير التي يراعى فيها أن يكون الموقع متجاوبًا مع سلوك المستخدم وبيئته، وذلك وفق متغيرات عديدة مثل حجم الشاشة ونظام التشغيل، واتجاه الشاشة، وذلك باستخدام مخططات وصور مرنة في الموقع.

ميزات موقع الويب المتجاوب

- **تحسين قابلية الاستخدام:** عند زيارة مواقع الويب التقليدية من خلال الهواتف المحمولة، يحتاج المستخدم إلى تكبير الصفحة ليتمكن من تصفح الموقع، بينما يتكيف موقع الويب المتجاوب مع المتصفحات المختلفة على الأجهزة المحمولة أيًا كان حجم شاشتها.
- **سهولة التحديث:** بدلًا عن الحاجة إلى وجود موقع ويب خاص بأجهزة الحاسوب، وآخر خاص بالهواتف المحمولة، مما يستدعي التحديث في عدة أماكن، يمكن استخدام موقع الويب المتجاوب من قِبل الجميع وهو ما يسهل على مدير الموقع تحديث موقع واحد فقط ، ويوفر الكثير من الوقت والجهد.
- **زيادة وقت التصفح:** يميل مستخدمو الأجهزة المحمولة إلى زيارة مواقع الويب لفترة قصيرة جدًا إذا وجدوا صعوبة في التصفح، و لكن موقع الويب المتجاوب يقدم تجربة تصفح سهلة وممتعة لجميع الخدمات والمنتجات في الموقع، مما يعني قيام المستخدم بقضاء وقت أكثر خلال زيارة الموقع.
- **توفير المال:** إن تكلفة إنشاء موقع ويب لمستخدمي الحواسيب إضافة إلى تطبيق أو موقع خاص لحاملي الهواتف المحمولة يعتبر مكلفًا. يمكن توفير المال من خلال إنشاء موقع متجاوب يناسب جميع المستخدمين.

عند تصميم صفحات الويب المتجاوبة يتم استخدام إحداثيات X,Y على شبكة تخطيطية، ويتم استخدام النسب المئوية للصور بدلًا من استخدام متغيرات عرض ثابتة، مما يعطينا تخطيطًا مرنًا للصفحة، يمكنه أن يعيد تغيير حجمه ليتناسب مع حجم جهاز العرض.

إن عملية إنشاء مخطط ديناميكي/متجاوب هي ناتج عمليتي دمج إطار العرض **Meta viewport** مع استعلامات الوسائط **Media queries** في وسم **<Meta>** داخل مقطع **HTML**

إطار العرض Viewport هو منطقة المستخدم المرئية في صفحة الويب، ويتم التحكم بها من خلال الوسم <Meta>.

لجعل استعلامات الوسائط **@media** تعمل على جميع الأجهزة، وخاصة عند عرض موقع الويب الديناميكي/المتجاوب على جهاز محمول، يجب أن نقوم بتضمين بيانات إطار العرض **<"meta name=viewport">** في جميع رؤوس صفحات الويب في العنصر **<html>**.

يضبط عرض الصفحة ليتناسب مع عرض الجهاز (والذي يتغير حسب الجهاز).

```
<"meta name="viewport" content="width=device-width, initial-scale=1.0>
```

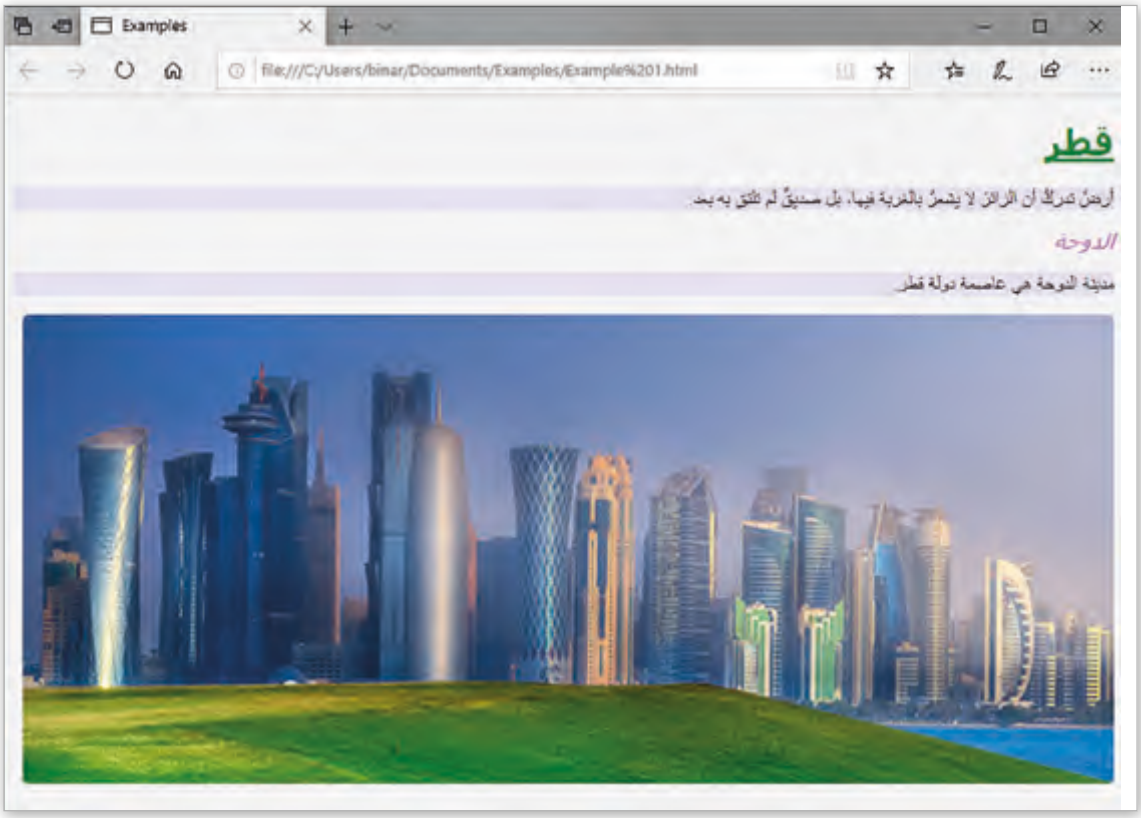
يُعطي المتصفح التعليمات اللازمة للتحكم بأبعاد وحجم الصفحة.

استعلامات الوسائط Media Queries هي عبارة عن خاصية في CSS تسمح للمطور باستخدام مجموعة خصائص CSS عند تحقق شرط معين فقط.

يخصص لاستعلامات الوسائط قسم خاص داخل ورقة التصميم **CSS**.

يحتوي استعلام الوسائط **(media query)** على نوع الوسيط ويمكن أن يحتوي على أكثر من تعبير **(expression)** سيترجم إما إلى صواب أو إلى خطأ.

في المثال التالي سيتم تغيير حجم الصورة إذا كان إطار العرض أقل من 800 بكسل.



سنضيف الآن استعلام الوسائط في ملف CSS.

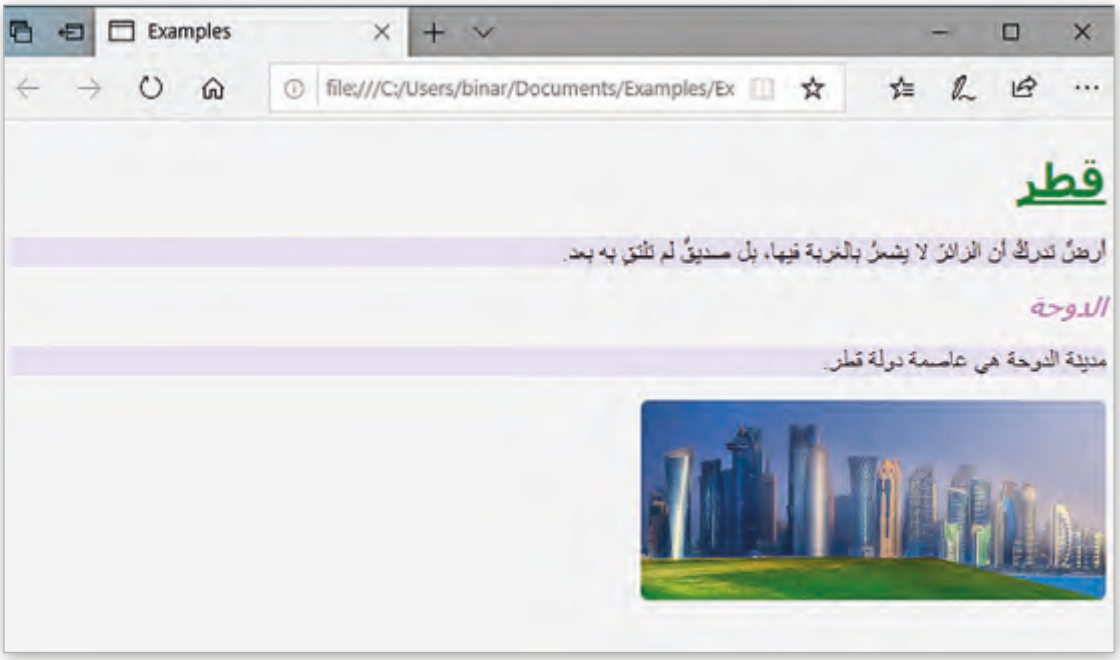
تستخدم لشاشات الحواسيب، والأجهزة اللوحية، والهواتف المحمولة.

.css

```
@media screen and (max-width: 800px) {
   img {
      width:300px;
      height:auto;
   }
}
```

أقصى عرض للمساحة المعروضة مثل نافذة المتصفح.

سنغير من نافذة المتصفح، فعندما يكون عرض المستند 800 بكسل أو أقل فإن حجم الصورة سيتغير.



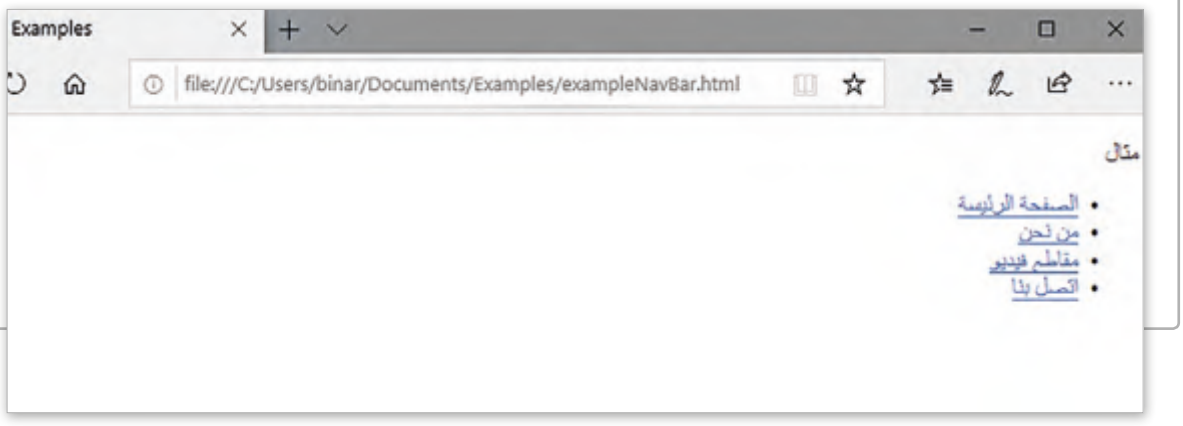
لنفتح الآن المجلد الخاص بموقع المتحف الوطني والذي قمنا بإنشائه سابقًا وذلك لاستكمال عملية التصميم.

يمكننا استخدام استعلامات الوسائط ووسم <meta> لإنشاء شريط تصفح خاص بصفحة الويب المتجاوبة.

لنستعرض شريط التصفح التالي.

.html

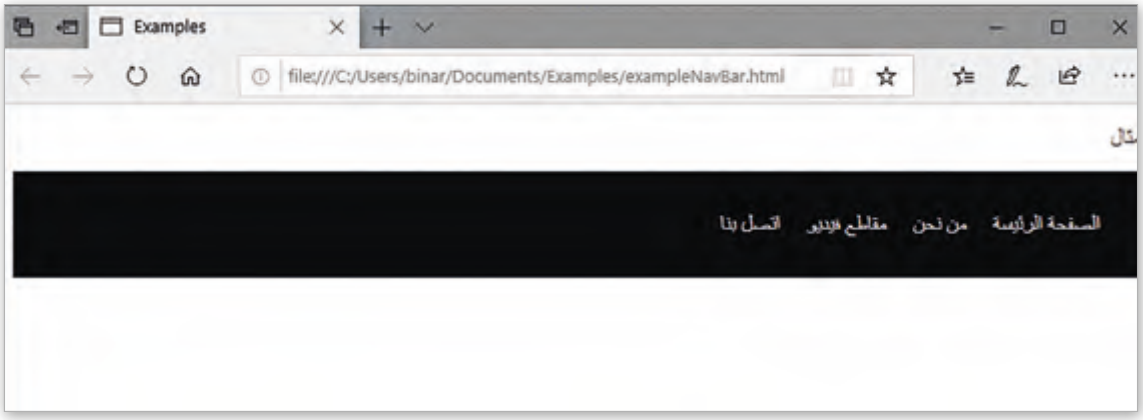
```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <title>Examples</title>
   <meta charset="UTF-8"/>
   <link rel="stylesheet" type="text/css" href="Stylesheet.css">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
   <p>مثال</p>
   <div class = "topNav">
      <ul>
         <li><a href = "#">الصفحة الرئيسة</a></li>
         <li> <a href = "#">من نحن</a></li>
         <li><a href = "#">مقاطع فيديو</a></li>
         <li><a href = "#">اتصل بنا</a></li>
      </ul>
   </div>
 </body>
```



لننسق شريط التصفح باستخدام ملف CSS خارجي:

.css

```css
.topNav{
  overflow: hidden;
  background-color: rgb(11, 11, 12);
}
.topNav ul{
  list-style-type: none;
  margin: 0;
  padding: 20px;
  overflow: hidden;
  background-color: none;
}
.topNav li{
  display:inline;
  float:none;
}
.topNav a{
  float: right;
  display: block;
  color: white;
  text-align: center;
  padding: 10px;
  text-decoration: none;
}
```



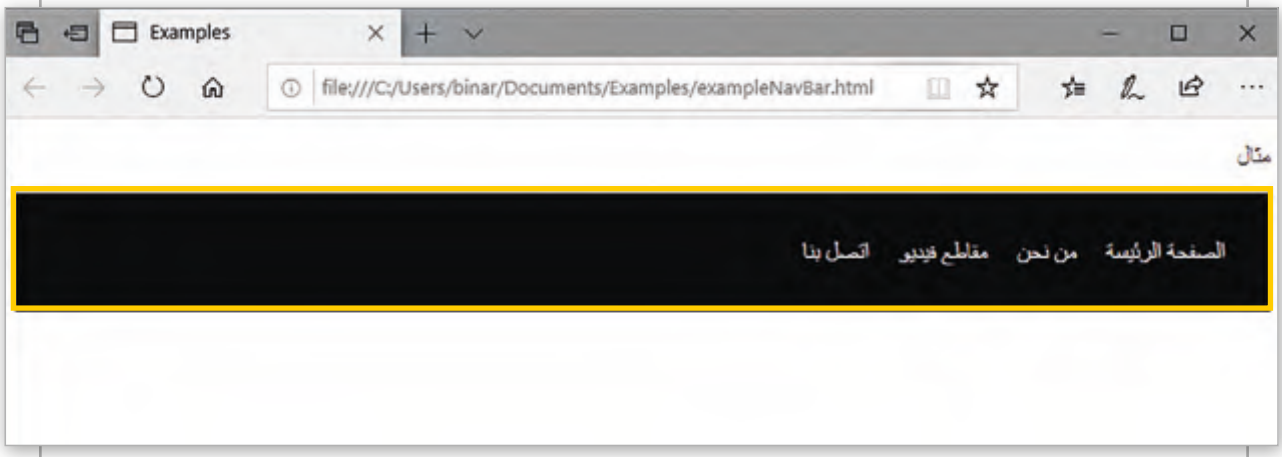
الآن سنضيف استعلام الوسائط في ملف CSS:

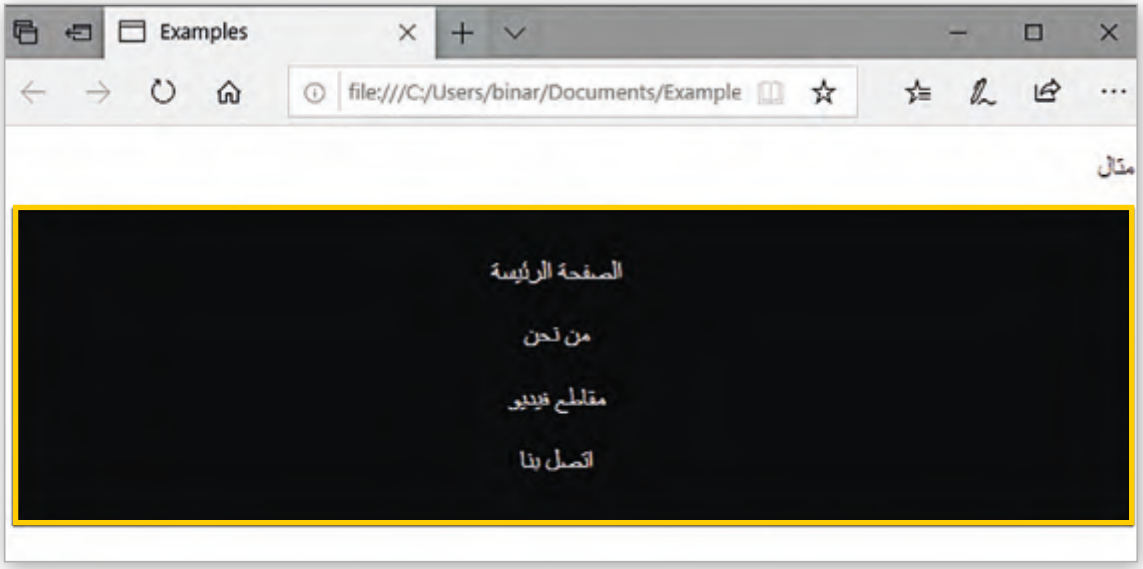
.css

```
@media screen and (max-width: 700px) {
  .topNav a {
    float: none;
    display: block;
    width: 100%;
  }
}
```

إذا كان عرض الشاشة أقل من 70 بكسل فإن شريط التصفح سيظهر بشكل عمودي.

إذ أعدنا تغيير حجم نافذة المتصفح فإن شريط التصفح سيتغير تلقائيًا.





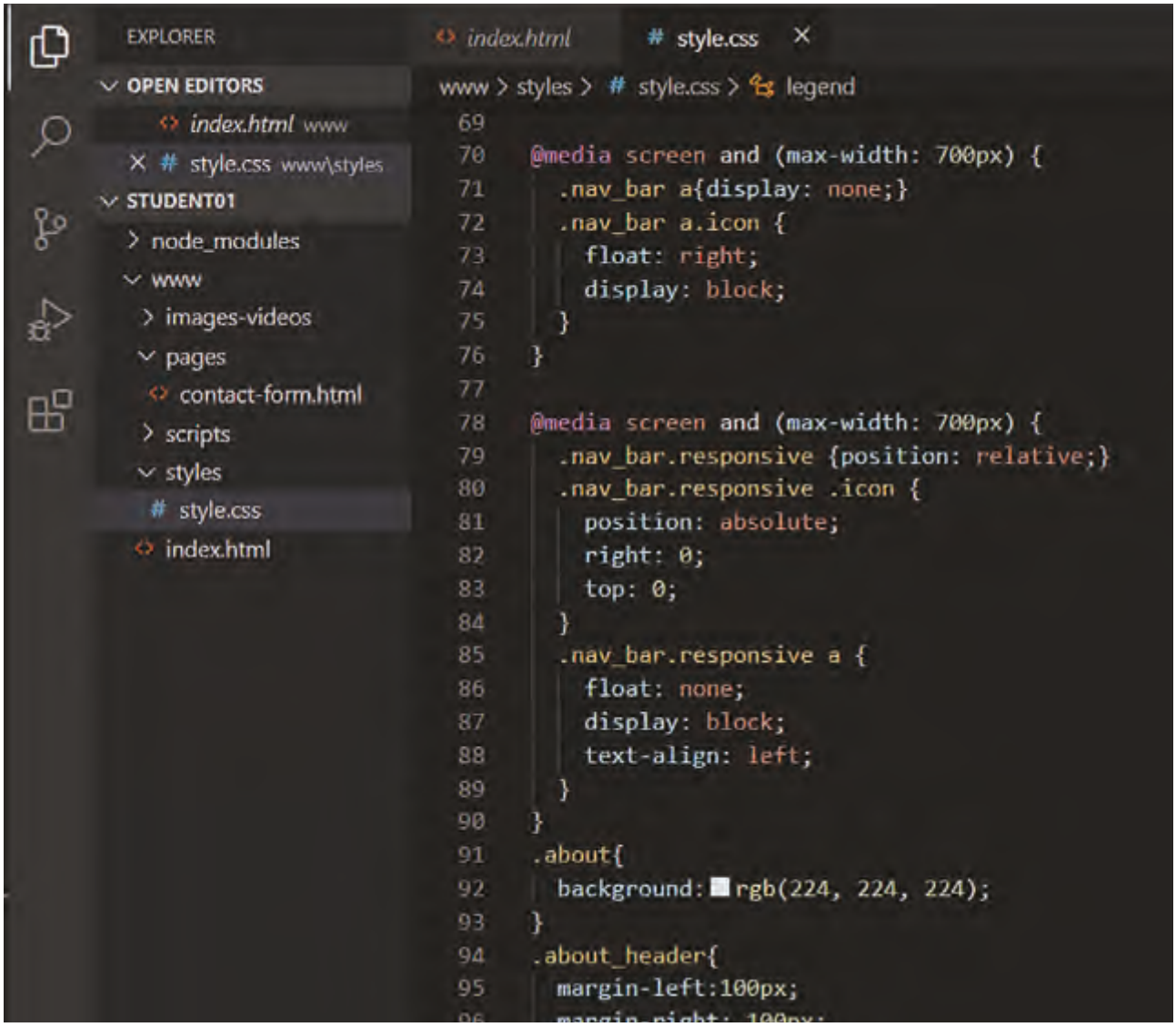
لنعد إلى مشروعنا، حيث سنجعل ألبوم الصور الآن يستجيب لحجم الشاشة التي يتم تحميل صفحتنا عليها.

هذا يعني أن عناصر الصفحة ستظهر بشكل مختلف في الحاسوب المكتبي أو في الجهاز اللوحي أو في الهاتف الذكي.

.css

```
@media only screen and (max-width: 900px) {
   .ResponsiveImage {
     width: 49.99999%;
     margin: 6px 0;
   }
}
@media only screen and (max-width: 700px) {
   .ResponsiveImage {
     width: 100%;
   }
}
video {
   max-width:100%;
   height:auto;
   margin-left: auto;
   margin-right: auto;
   display: block;
}
```

سيبدو المقطع بالشكل التالي داخل برنامج **Visual Studio Code**:



الآن سنبين كيفية استخدام ميزة محاكاة الجهاز **Device Emulation** في أدوات المطور في **Microsoft Edge**.

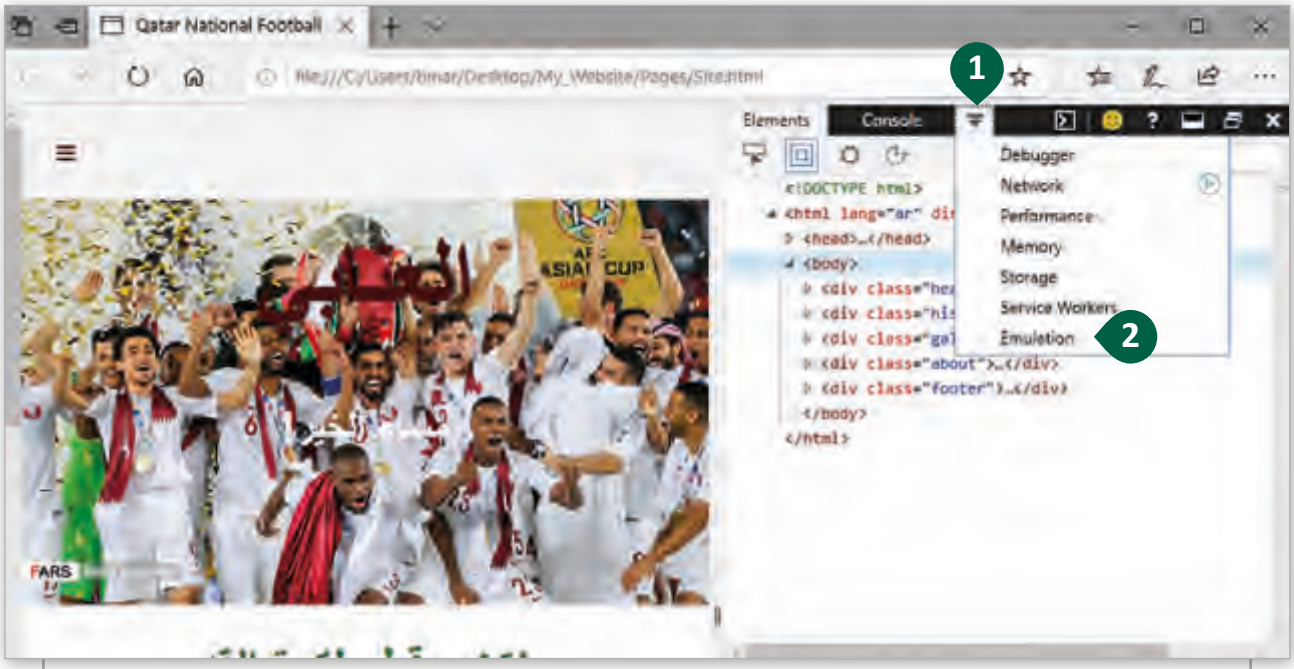
تفيدنا هذه الميزة في اختبار مظهر الموقع على أجهزة مختلفة مثل الهواتف الذكية والأجهزة اللوحية.

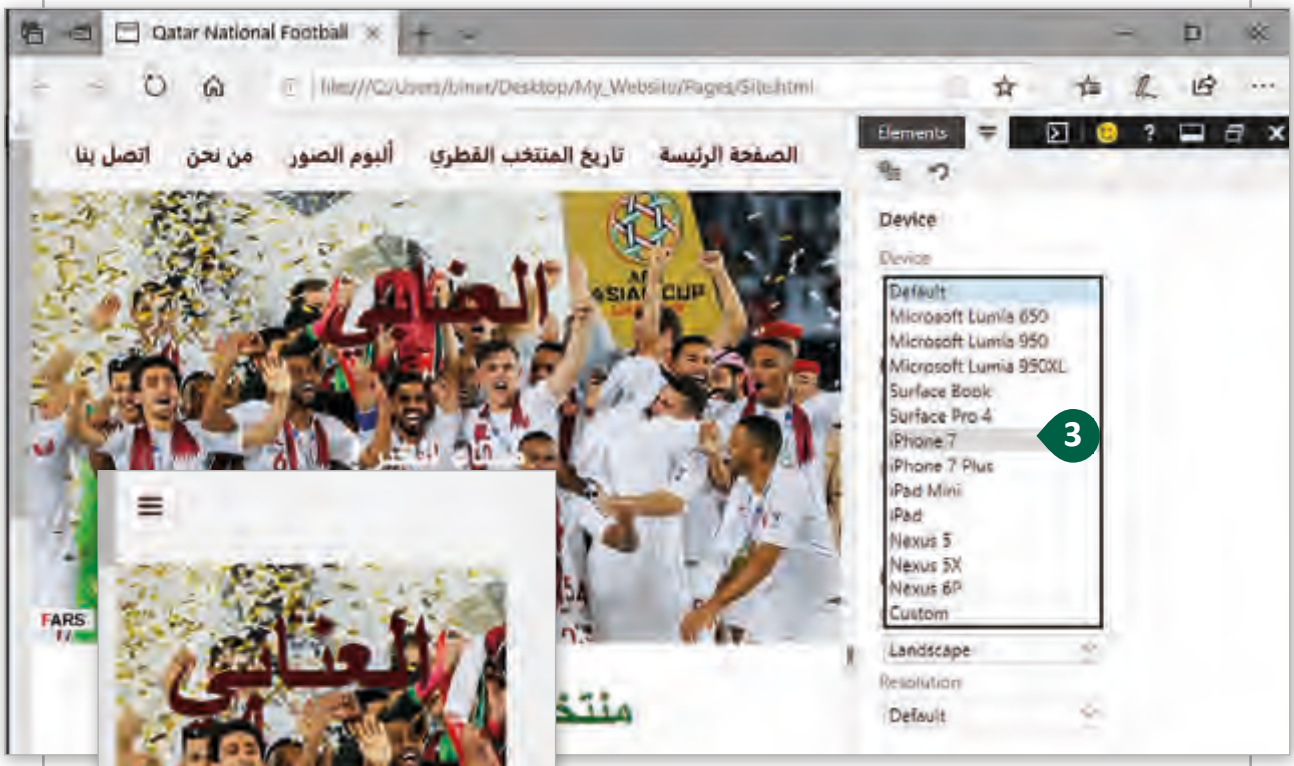
كيف نستخدم ميزة محاكاة الجهاز:

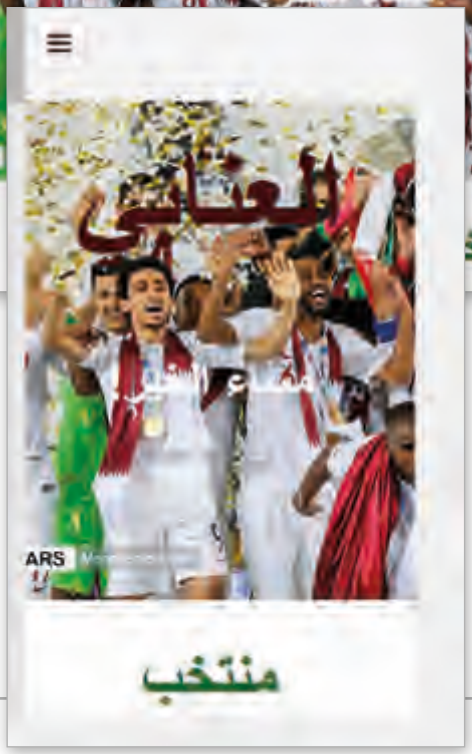
اضغط Ctrl + Shift ⇧ + i.

> اضغط المؤشر ❶ ومن القائمة المنسدلة اضغط Emulation. ❷

> ثم قم باختيار الجهاز الذي تريده.❸


Qatar National Football
Elements
Console
Debugger
Network
Performance
Memory
Storage
Service Workers
Emulation
1
2


Qatar National Football
الصفحة الرئيسة
تاريخ المنتخب القطري
ألبوم الصور
من نحن
اتصل بنا
Elements
Device
Default
Microsoft Lumia 650
Microsoft Lumia 950
Microsoft Lumia 950XL
Surface Book
Surface Pro 4
iPhone 7
iPhone 7 Plus
iPad Mini
iPad
Nexus 5
Nexus 5X
Nexus 6P
Custom
Landscape
Resolution
Default
3


منتخب

1

قم بتحرير المقطع التالي واستخدم خاصية padding لضبط كل من التعبئة العلوية والتعبئة السفلية الخاصة بـ <p> إلى "20px"، واليسرى واليمنى إلى "50px".

ثم قم بإضافة استعلام وسائط لتغيير لون الخلفية اذا كانت الشاشة أصغر من 400 بكسل.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
      h1 {
         background-color: light-grey;
      }

   </style>
</head>
<body>

   <h1>This is a heading</h1>
   <p>This is a paragraph.</p>

</body>
</html>
```

2

قم بتحرير المقطع التالي واضبط الهامش الخاص بـ div إلى "20px"، ولون النص أخضر "green"، والحدود border إلى "3px، black" و solid.

ثم قم بإضافة استعلام وسائط لتغيير لون الخلفية اذا كانت الشاشة أصغر من 600 بكسل.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>
      h1 {
         background-color: light-grey;
      }
      
   </style>
</head>
<body>

   <h1>This is a heading</h1>
   <p>This is a paragraph.</p>
   <div class="paragraph_2">
      <p>This is a second paragraph</p>
      <p>This is a third paragraph</p>
   <div>
</body>
</html>
```

3

على فرض أنك كنت قد أضفت صورة إلى موقعك، قم بتحرير الكود التالي واضبط خاصية العرض width لهذه الصورة إلى "150px" والارتفاع height إلى "100px"، وخاصية border-radius إلى "5px".

قم باستخدام استعلام الوسائط لتغيير الصورة اذا كانت الشاشة أصغر من 500 بكسل.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <style>

   </style>
</head>
<body>

   <h1>This is a heading</h1>
   <p>This is a paragraph.</p>
   <img src="picture.png">
</body>
</html>
```

4

ما هي أهمية إنشاء موقع ويب متجاوب، و كيف يمكنك إنجاز ذلك؟

5

افتح المجلد الخاص بموقع Adventure_Site وقم باكمال التصميم كالتالي:

> أولاً قم بتطبيق قواعد CSS الخاصة padding، الهوامش margin والحدود border الخاصة بالنموذج كما تعلمت في الدرس السابق.

> طبق التصميم الذي تريده على الروابط التي ستشكل شريط التصفح للموقع. قم بإنشاء شريط تصفح أفقي.

> تأكد من أن موقعك تفاعلي من خلال إضافة استعلامات الوسائط (media queries) المناسبة لجعل الصور وشريط التصفح تفاعليين.

> قم بحفظ عملك.

6

في المقطع البرمجي التالي، ما هي قاعدة CSS المفقودة التي إذا قمنا بتغيير حجم نافذة المتصفح بحيث أصبح عرض المستند 600 بكسل أو أقل، فإن لون الخلفية سيتغير إلى lightpink. اختر الإجابة الصحيحة.

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<head>
   <meta charset="UTF-8"/>
   <meta name="viewport" content="width=device-
   width, initial-scale=1.0">
   <style>
      body {
         background-color: lightgrey;
      }
   </style>
</head>
<body>
   <h1>This is a heading</h1>
   <p>This is a paragraph.</p>
</body>
</html>
```

```
@media screen and (max: 600px) {
  body {
    background-color: lightpink;
  }
}
```

```
@media (max-width: 600px) {
  body {
    background-color: lightpink;
  }
}
```

```
@media screen and (max-width: 600px) {
  body {
    background-color: lightpink;
  }
}
```

```
@media screen and (max-width: 600px) {
  body {
    background-color: lightpink;
  }
```

لغة Javascript

تعتبر Javascript لغة برمجة خاصة تسمح لنا بالتفاعلية اللازمة، والتي لا يمكن أن تتم من خلال مقاطع HTML العادية وأوراق التنسيق المتتالية CSS.

ما هو دور لغة JavaScript؟

- تُغير محتوى الصفحة الذي تم إنشاؤه بمقاطع HTML.
- تُغير من قيم خصائص HTML.
- تغير CSS.

تتم كتابة مقاطع JavaScript بين وسمي <script> و</script>.

أين يمكننا إضافة Javascript؟

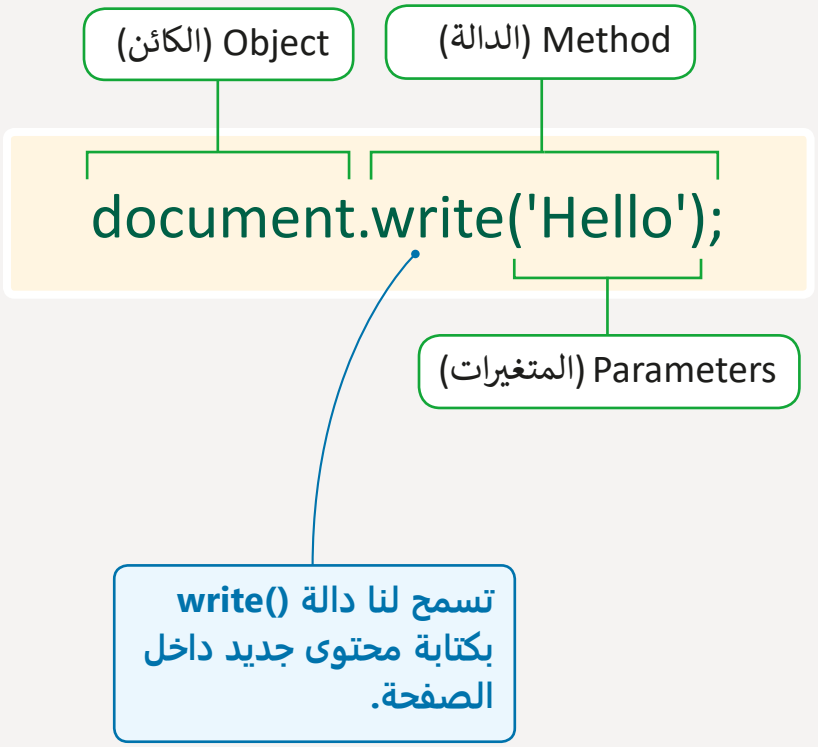
- في <head> و<body> في صفحة HTML.
- في ملف خارجي.

تحتوي الملفات على كود JavaScript بامتداد ".js"، ولأجل استخدامه، يتم إضافة اسم الملف المحتوي على الكود script في خاصية src من وسم <script>.

```
<script src="first_script.js"></script>
```

توجد 3 طرق تُمكننا من عرض مقطع **JavaScript**:

- الكتابة داخل وسم HTML باستخدام innerHTML.
- الكتابة في مقطع HTML باستخدام document.write().
- الكتابة داخل نافذة/صندوق تحذيري باستخدام windows.alert().
- الكتابة في لوحة التحكم لمستعرض الويب باستخدام console.log للوصول إلى العنصر، نستخدم الدالة document.getElementByid('id').

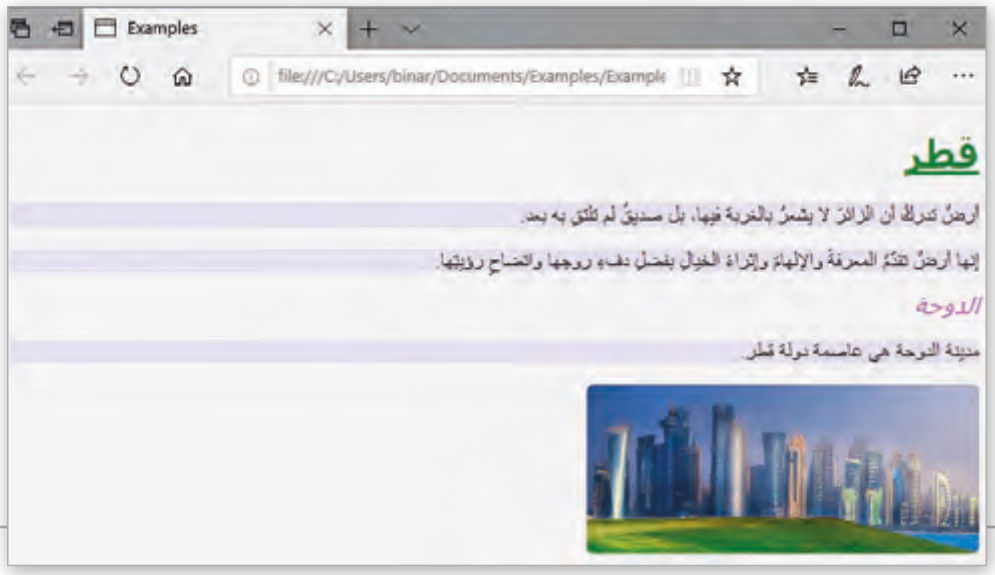


استخدام دالة document.write()

في المثال التالي سنستخدم **document.write()** لأجل إضافة فقرة إلى المستند.

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<html>
   <title>Examples</title>
   <meta charset="UTF-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</html>
<body>
   <h1>قطر</h1>
   <p> أرضٌ تدركُ أن الزائرَ لا يشعرُ بالغربة فيها، بل صديقٌ لم تلتقِ به بعد. </p>
<!--We add the JavaScript in the html file-->
   <script>
      document.write("<p> إنها أرضٌ تقدّمُ المعرفةَ والإلهامَ وإثراءَ الخيالِ بفضلِ دفءِ
      روحِها واتضاحِ رؤيتِها.</p>")
   </script>
   <h2>الدوحة</h2>
   <p>مدينة الدوحة هي عاصمة دولة قطر.</p>
   <img src="Images/doha (1).jpg">
</body>
</html>
```

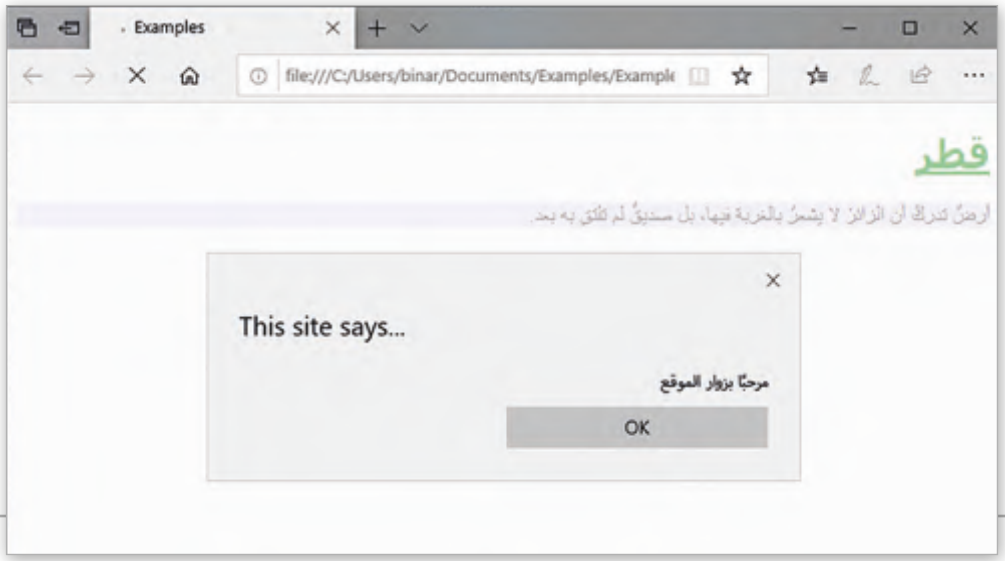


استخدام دالة window.alert()

يمكننا استخدام صندوق تحذيري لعرض البيانات بواسطة دالة **window.alert()**، والتي تنشئ صندوقا حواريا يحتوي على رسالة، وعلى المستخدم الضغط على زر OK لإغلاق ذلك الصندوق.

.html

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<html>
   <title>Examples</title>
   <meta charset="UTF-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</html>
<body>
   <h1>قطر</h1>
   <p> أرضٌ تدركُ أن الزائرَ لا يشعرُ بالغربة فيها، بل صديقٌ لم تلتقِ به بعد. </p>
<!--We add the JavaScript in the html file-->
   <script>
      window.alert(مرحبًا بزوار الموقع)
     </script>
   <h2>الدوحة</h2>
   <p>مدينة الدوحة هي عاصمة دولة قطر.</p>
   <img src="Images/doha (1).jpg">
</body>
</html>
```



استخدام دالة console.log()

تستخدم دالة **console.log()** في طباعة أي نوع من أنواع المتغيرات المعرفة قبله، أو في طباعة أي رسالة من اختيارنا نرغب بعرضها للمستخدم.

تكتب دالة **console.log()** رسالة إلى لوحة التحكم **console**.

المتغير: Object (كائن) أو message (رسالة).

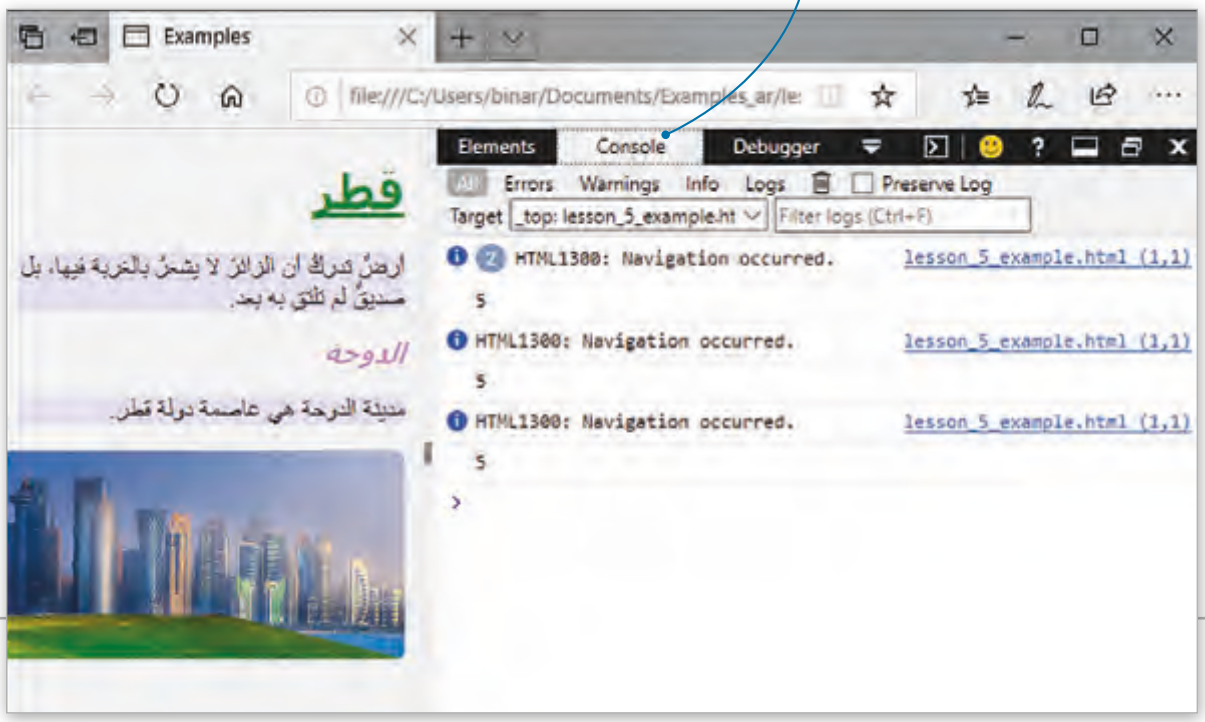
```
console.log(message)
```

عليك أن تقوم بمراجعة النتيجة في لوحة تحكم المتصفح. لفتح اللوحة عليك أن تضغط F12.

فلنستعرض هنا بعض الأمثلة:

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
<html>
   <title>Examples</title>
   <meta charset="UTF-8"/>
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" type="text/css" href="MyCSSfile2.css">
</html>
<body>
   <h1>قطر</h1>
   <p> أرضٌ تدركُ أن الزائرَ لا يشعرُ بالغربة فيها، بل صديقٌ لم تلتقِ به بعد. </p>
<!--We add the JavaScript in the html file-->
   <script>
      var x=5;
      console.log(x);
   </script>
   <h2>الدوحة</h2>
   <p>مدينة الدوحة هي عاصمة دولة قطر.</p>
   <img src="Images/doha (1).jpg">
</body>
</html>
```

لرؤية نتيجة المقطع في المتصفح، يجب أن نضغط على مفتاح F12 ثم نضغط علامة تبويب Console (لوحة التحكم).



تزودنا متصفحات الويب بلوحة تحكم JavaScript كجزء من أدوات المطور Developer Tools، والتي تعتبر مفيدة في التعرف على:

- الأخطاء والتحذيرات التي تحدث في صفحة الويب يتم حفظها داخل لوحة التحكم.
- أوامر JavaScript الخاصة بالتفاعل مع صفحة الويب يمكن تنفيذها في لوحة التحكم.

المقارنة بين لغة Python و Javascript

تتشابه لغتا البرمجة Python و Javascript في دعمهما لنمط البرمجة المعتمد بشكل أساسي على المتغيرات و الدوال.

تعرفنا سابقًا على مبادئ المتغيرات و الدوال و الجمل الشرطية و التراكيب البرمجية و التكرارات في Python. يمكننا استخدام نفس تلك المفاهيم بتعديلات بسيطة في Javascript.

سنقارن أدناه بين اللغتين من خلال بعض الأمثلة.

Python

```python
#---  Declaring and printing variables ---#

# Integer Variable
integerNumber = 8

# Float Variable
floatNumber = 5.678

# String Variable
welcomeMessage = "Hello from JavaScript!"

# Printing the variables
print(integerNumber)

print(floatNumber)

print(welcomeMessage)
```

لإنشاء متغير، سنقوم بتعريفه و تعيين قيمة لذلك المتغير.

النتيجة

```
8

5.678

Hello from JavaScript
```

.javascript

```javascript
//--- Declaring and printing variables ---//

// Integer Variable
var integerNumber = 8;

// Float Variable

var floatNumber = 5.678;

// String Variable

var welcomeMessage = "Hello from JavaScript!";

// Printing the variables
console.log(integerNumber);

console.log(floatNumber);

console.log(welcomeMessage);
```

تستخدم الكلمة المفتاحية Var لتعريف متغير.

النتيجة

```
8

5.678

Hello from JavaScript
```

القرارات (الجمل الشرطية)

جملة **if..else** لا تختلف عن **Python**.

Python

```
# Flow Control

# Basic if-branch
height1 = 1.50

if height1 > 1.65:
   print("You are permitted to pass")

# Basic if-else-branch
height2 = 1.30

if height2 > 1.65:
   print("You are permitted to pass")
else:
   print("You are NOT allowed to enter")

# Basic if-else-if branch
height3 = 1.47

if height3 > 1.65:
   print("You are permitted to pass")
elif height3 > 1.40:
   print("You are allowed into in the narby machine")
else:
   print("You are NOT allowed to enter")
```

.javascript

```javascript
// Flow Control

// Basic if branch
var height1 = 1.50;
if (height1 > 1.65) {
   console.log("You are permitted to pass");
}

// Basic if-else branch
height2 = 1.30;
if (height2 > 1.65) {
   console.log("You are permitted to pass");
}
else {
   console.log("You are NOT allowed to enter");
}

// Basic if-else-if branch
var height3 = 1.47;
if (height3 > 1.65) {
   console.log("You are permitted to pass");
}
else if (height3 > 1.40) {
   console.log("You are allowed into in the narby machine");
}
else {
   console.log("You are NOT allowed to enter");
}
```

الدوال

الدالة هي مقطع برمجي يعمل فقط عند استدعاء تلك الدالة داخل البرنامج. يتم تعريف الدالة في لغة بايثون **Python** باستخدام الكلمة المفتاحية **def**، بينما تستخدم الكلمة المفتاحية **function** لتعريف دالة في **Javascript**.

Python

```
# Using functions

# Functions without arguments

def squareRootOf5():
   return 5 * 5
finalNumber = squareRootOf5()
   print(finalNumber)

# Functions with arguments

def profileMessage(name, age):
   print("Your name is", name, "and you are", age, "years old.")
   profileMessage("Saad", 17)
```

.javascript

```
// Using functions
// Functions without arguments
function squareRootOf5() {
   return 5 * 5;
}
var finalNumber = squareRootOf5();
console.log(finalNumber);

// Functions with arguments
function profileMessage(name, age) {
   console.log(`Your name is ${name} and you are ${age} years
   old`);
}
profileMessage("Saad", 17)
```

في موقع الويب الخاص بالمنتخب الوطني القطري الذي سنقوم بإنشائه، سنضيف كود Javascript لجعل الموقع أكثر جاذبية للمستخدمين.

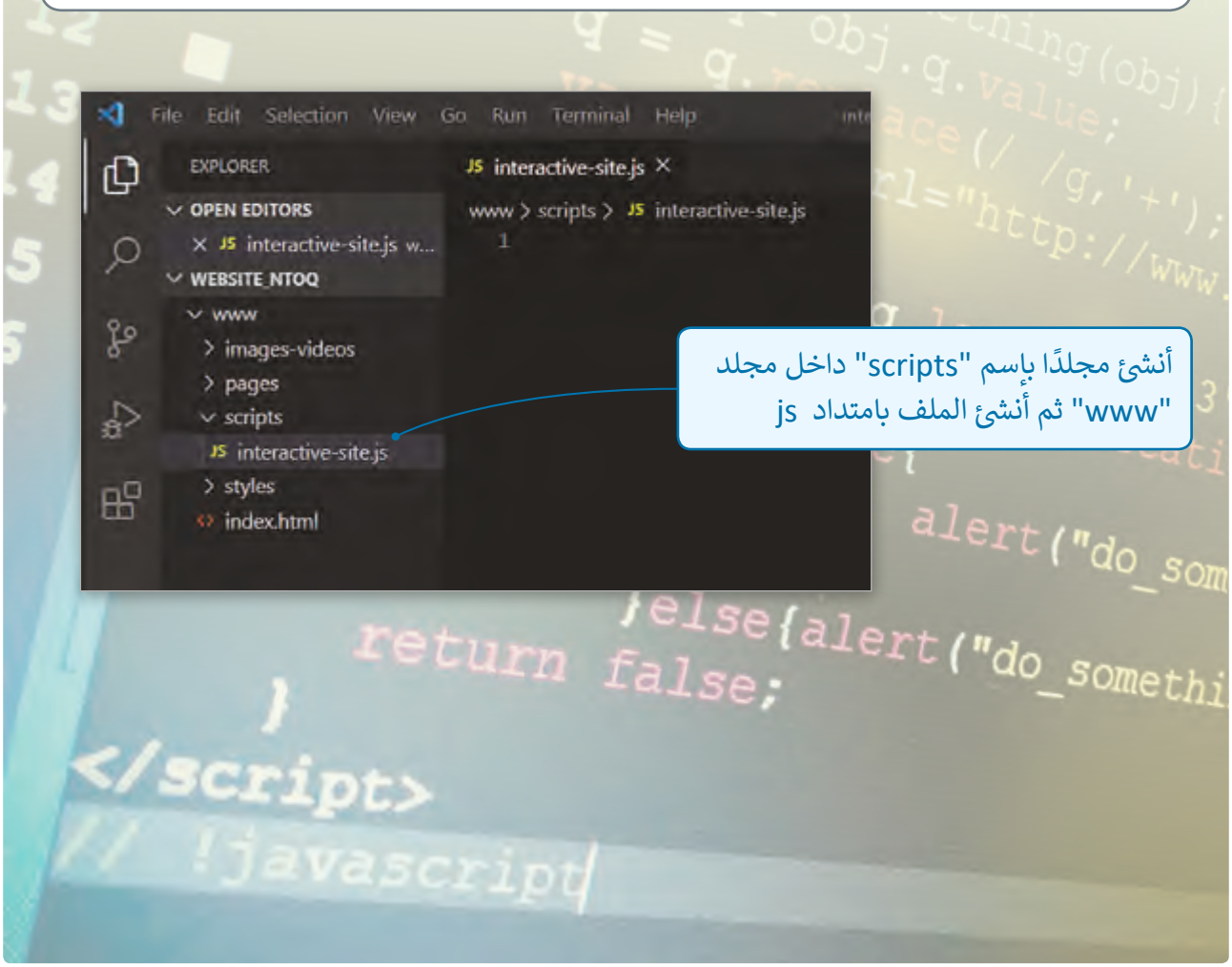
سنقوم بكتابة script لتنفيذ الآتي:

> جعل شريط التصفح واضحًا للمستخدم عند استعراض الموقع على الهاتف المحمول.

> عرض رسالة ترحيبية من الموقع للمستخدم تتناسب مع الوقت.

يمكننا أيضًا إنشاء أيقونة تظهر عندما يصبح عرض الشاشة صغير نسبيًا.

يتشابه هذا مع وظيفة واجهة المستخدم الموجودة في الهواتف المحمولة حيث تكون الشاشة صغيرة. للقيام بذلك سننشئ ملفًا خارجيًا يحتوي على كود JavaScript بنفس الطريقة التي أنشأت بها ملف HTML وCSS ولكن بالامتداد js



.html

```
<div class="nav_bar" id="myNav_bar">
  <ul>
    <li><a href="#top">الصفحة الرئيسة</a></li>
    <li><a href="#about"> تاريخ المنتخب القطري</a></li>
    <li><a href="#history"> ألبوم الصور</a></li>
    <li><a href="#gallery">من نحن</a></li>
    <li><a href="pages/contact-form.html" target="_blank">اتصل
    بنا<a></li>
    <li><a href="javascript:void(0);" class="icon"
    onclick="myFunction()"><i class="fa fa-bars"></i></a></li>
  </ul>
</div>
```

ينشئ هذا السطر البرمجي أيقونة تظهر لنا كقائمة خاصة بالشاشات صغيرة الحجم.

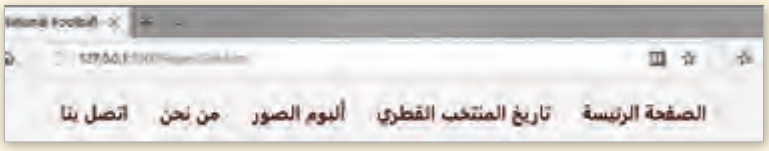
.js

```
// A function that adjusts the site properties depending on the
screen width of the user's device
function myFunction() {
// Picking out the element with the corresponding Id from the
 Html structure and storing it in the variable 'x'
var x = document.getElementById("myNav_bar");
// Adding a new CSS class to the selected element if the
 condition is true
if (x.className === "nav_bar") {
  x.className += " responsive";
  }
  else {
    x.className = "nav_bar";
  }
}
```

لاحظ أن هذه الأيقونة تحتاج إلى التصميم بواسطة **CSS** والربط مع المقطع البرمجي **script** الذي كنا قد كتبناه مسبقا.

.css

```
/*Hide the link that should open and close the navigation bar
on small screens */
.nav_bar .icon {
  display:none;
}
/* When the screen is less than 700 pixels wide, hide all
links. Show the link that should open and close the navigation
bar (.icon) */
@media screen and (max-width: 700px) {
  .nav_bar a{display: none;}
  .nav_bar a.icon {
    float: left;
    display: block;
  }
}
/* The "responsive" class is added to the nav_bar with
JavaScript when the user clicks on the icon. */
/* This class makes the nav_bar look good on small screens
(display the links vertically instead of horizontally) */
@media screen and (max-width: 700px) {
  .nav_bar.responsive {position: relative;}
  .nav_bar.responsive .icon {
    position: absolute;
    left: 0;
    top: 0;
  }
  .nav_bar.responsive a {
    float: none;
    display: block;
    text-align: right;
  }
}
```



Qatar National Football
127.0.0.1:5500/Pages
منتخب قطر لكرة القدم
ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، وبحلول عام 1948 لعب عمال النفط المغتربين أول مباراة رسمية لكرة القدم في قطر، ثم تأسس الاتحاد القطري لكرة القدم في عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIFA في عام 1970.
خاض المنتخب الوطني القطري مبارياته الأولى في تصفيات FIFA المؤهلة لكأس العالم عام 1977، وشارك الفريق لأول مرة في كأس آسيا عام 1980 والذي استضافته دولة قطر، وقد حقق الفريق نجاحات ومراكز مختلفة خلال تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على المستوى الإقليمي، حيث فاز بكأس الخليج 1992 للمرة الأولى في تاريخه، وتوج بطل الخليج للمرة الثانية في تاريخه في عام 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر كمستضيف لكأس العالم 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى العالم.
حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب كأس آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداءً راقيًا ليكون العنابي أول فريق يتوج بالكأس الآسيوية في نسختها الجديدة.
ألبوم الصور

Qatar National Football
127.0.0.1:5500/Pages
الصفحة الرئيسة
تاريخ المنتخب القطري
ألبوم الصور
من نحن
اتصل بنا
FARS
منتخب قطر لكرة القدم
ظهرت كرة القدم في دولة قطر بالتزامن مع اكتشاف النفط عام 1940، وبحلول عام 1948 لعب عمال النفط المغتربين أول مباراة رسمية لكرة القدم في قطر. ثم تأسس الاتحاد القطري لكرة القدم في عام 1960 وانضم الاتحاد القطري لكرة القدم إلى منظمة FIFA في عام 1970.
خاض المنتخب الوطني القطري مباراته الأولى في تصفيات FIFA المؤهلة لكأس العالم عام 1977، وشارك الفريق لأول مرة في كأس آسيا عام 1980 والذي استضافته دولة قطر. وقد حقق الفريق نجاحات ومراكز مختلفة خلال
تسعينيات القرن الماضي ومطلع القرن الحالي، وذلك على المستوى الإقليمي، حيث فاز بكأس الخليج 1992 للمرة الأولى في تاريخه، وتوج بطل الخليج للمرة الثانية في تاريخه في عام 2004، ومرة ثالثة عام 2014، وفي ديسمبر 2010، تم الإعلان عن قطر كمستضيف لكأس العالم 2022، لتلعب دولة قطر للمرة الأولى في البطولة الأهم على مستوى العالم.
حقق المنتخب إنجازه الأكبر على الإطلاق في مسيرته الرياضية عام 2019 حين توج بلقب كأس آسيا 2019 بعد أن تغلب على نظيره الياباني بنتيجة 3-1 في مباراة قدم فيها أداء راقيًا ليكون

الرسالة الترحيبية

الخاصية الجديدة هي عبارة عن مقطع برمجي **script** آخر يعرض رسالة تتغير وفق وقت تسجيل المستخدم لدخوله.

سنضيف رسالة ترحيبية في الوسم header.
سننشيء قسمًا باسم
وبداخلها سنضيف المقطع البرمجي الخاص بنا.

.html

```
<div class="welcome">
<h1 style="text-align:center; color:white; text-shadow:rgb(94, 9, 9); font-size:
40px;">
<script src="scripts/interactive-site.js"></script> </h1> </div>
```

.js

```
// This block gets the time to use in the following code
var today = new Date();
var hourNow = today.getHours();
// Declaring the greeting
var greeting;
// We do not assign a value to manipulate it later
// Changing the content of the greeting message depending on
 the time that was given by the Date() block
if (hourNow < 12) {
greeting="صباح الخير!";
}
else if ( hourNow > 12) {
greeting="مساء الخير!";
}
else {
greeting="أهلًا وسهلًا";
}
// Give a simple expanation of the document
document.write(greeting);
// A built-in function that writes on the document
```

تحسين محرك البحث (Search Engine Optimization-SEO)

يقصد بمصطلح تحسين محرك البحث جميع الإجراءات التي نحتاجها في هيكلية وتركيب محتوى موقع الويب لأجل تحسين الترتيب العالمي للموقع وزيادة معدل زيارات المستخدمين، وذلك من خلال الوصول إلى الموقع عن طريق نتائج البحث باستخدام الكلمات المفتاحية المتعلقة بمحتواه وليس عن طريق الإعلانات المدفوعة.

توجد عدة تقنيات من السهل تنفيذها وعادة ما تأتي بنتائج جيدة لزيادة معدل تصفح الموقع:

تتغير تقنيات تحسين محركات البحث بشكل دائم، نستعرض هنا بعضًا منها:

- قم بتحسين العناوين الرئيسة، حيث أن وسم العنوان له وزنه الخاص في محرك البحث. يجب أن تكون الكلمة المفتاحية المهمة في البداية، مع إضافة الكلمات المفتاحية المناسبة إلى الرأس قدر الإمكان.
- اختر الصور ذات الأحجام المناسبة: تتطلب الصور مساحة تخزينية أكبر على الخادم، مما يعني أنها ستتطلب مساحة تخزينية أكبر على جهاز المستخدم، و مزيدا من الوقت للتحميل. إن الصور التي تستغرق وقتًا طويلًا لتحميلها تؤثر بشكل سلبي على ترتيب الموقع.
- قم بتشغيل أدوات تتبع زيارات موقع الويب الخاص بك. إن التتبع هو مصطلح مهم جدًا في SEO، فتتبع نتائج تصفح الموقع الغير مدفوعة تخبرنا إذا ما كنا قد حققنا أي تقدم أم لا في جذب المزيد من الزوار.

1 اختر الإجابة الصحيحة:

1. ما هو وسم HTML الصحيح الذي يُمكننا من كتابة مقطع Jacascript؟	○ <javascript> ○ <scripted> ○ <script> ○ <js>
2. ما هي صيغة JavaScript الصحيحة لطباعة القيم في لوحة تحكم Console؟	○ print(5) ○ console.log(5); ○ console.print(5); ○ print.console(5);
3. ما هي الصيغة الصحيحة للرجوع إلى مقطع برمجي خارجي Script باسم "example.js"؟	○ <script src=”example.js”> ○ <script href=”example.js”> ○ <script ref=”example.js”> ○ <script name=”example.js”>
4. أي من التالي هي الصيغة الصحيحة لعرض جملة "This is an example" في صندوق تحذيري باستخدام JavaScript؟	○ alertbox(“This in an example”); ○ msg(“This in an example”); ○ msgbox(“This in an example”); ○ window.alert(“This in an example”)

5. ما هي الصيغة الصحيحة لإنشاء دالة في JavaScript باسم SayHello()؟	○ function = sayHello() ○ function sayHello() ○ function := sayHello() ○ function : sayHello()
6. كيف يُمكن استدعاء الدالة في JavaScript؟	○ call sayHello(); ○ call function sayHello(); ○ sayHello(); ○ function sayHello();
7. ناتج تنفيذ مقطع JavaScript التالي هو: <script type="text/javascript"> a = 8 + "8"; document.write(a); </script>	○ 16 ○ Compilation Error ○ 88 ○ Run Time Error
8. اختر صيغة JavaScript الصحيحة لتغيير محتوى مقطع HTML التالي. Qatari Students	○ document.getElement(“student”).innerHtml=”I am a from student”; ○ document.getElementById(“student”).innerHtml=”I am a student”; ○ document.getId(“geek”)=”I am a student”; ○ document.getElementById(“student”).innerHtml=I am a student;

9. نتيجة عرض لوحة تحكم Console الخاصة بمقطع JavaScript التالي:

```
<script>
function student() {
   if(true) {
      var a = 5;
   }
   document.write(a);
} student();
</script>
```

- Compilation Error
- Nothing will be printed as output
- 5
- Runtime error

10. ناتج تنفيذ مقطع JavaScript التالي:

```
<script type="text/javascript">
<!--
document.write("Hello");
//-->
</script>
```

- Nothing
- Compilation Error
- Hello
- <!–document. w r i t e (“ H e l l o ”) ; / / – >

2

افتح المجلد الخاص بموقع Adventure_Site وقم باستكمال التصميم كالتالي:

> افتح ملف HTML المركزي وقم بإنشاء نص تحية للترحيب بالزائرين.

> لقد قمت سابقًا بإنشاء شريط تنقل والآن يمكنك إضافة رمز القائمة ليتناسب المظهر مع الجهاز المحمول. لتحقيق ذلك، قم بإضافة نص برمجي في HTML ومن بعدها قم بالتغييرات المناسبة في استعلامات الوسائط في ملف CSS الخاص بك.

> احفظ التغييرات التي أجريتها.

الدرس الثالث

المعالجة المعتمدة على الخادم

يحتوي موقع الويب الذي أنشأناه على صفحة اتصل بنا، والتي تحتوي بدورها أربعة حقول لإدخال البيانات من قبل الزوار، ولكن يمكنك ملاحظة أن ضغط زر الإرسال submit لا يُحدث نتيجة وذلك لأننا لم نبرمج هذا الزر لأداء أي وظيفة بعد، ويفترض أن يقوم بتعبئة البيانات المدخلة في قواعد بيانات خاصة في جهة الخادم، وسنقدم في هذا الدرس بعض مهارات المعالجة المعتمدة على الخادم.

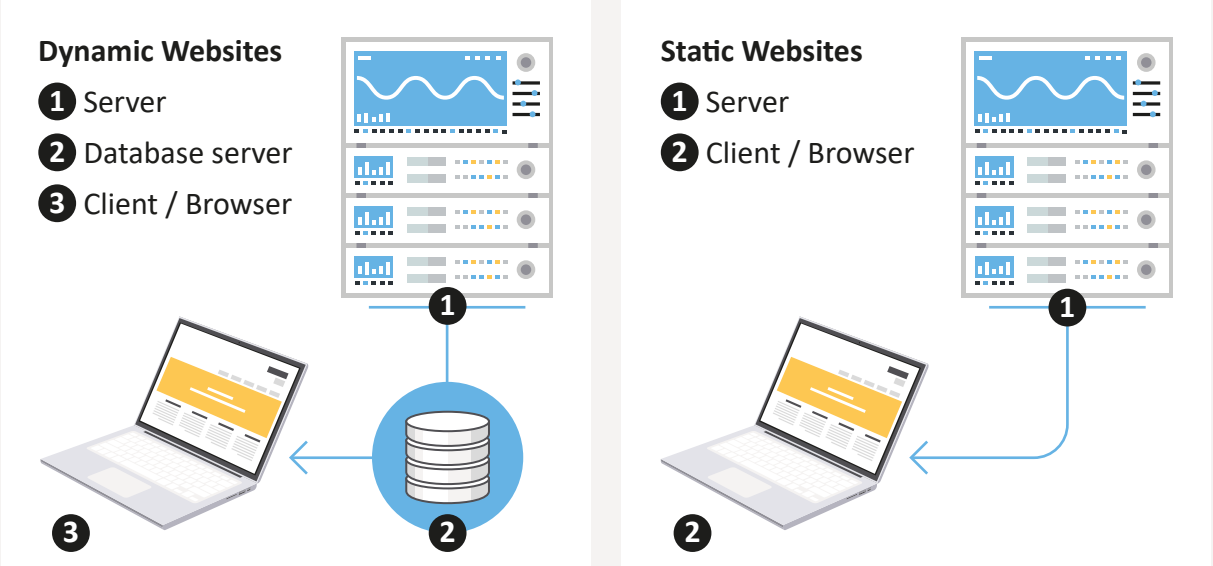
قبل الانتقال لهذه الخطوة سنستعرض بعض المفاهيم التي توفر فهمًا أفضل لكيفية عملها.

مواقع الويب الساكنة والتفاعلية

يتم استضافة مواقع الويب على خادم أو أكثر بحيث يتم الوصول لصفحة الموقع الرئيسة باستخدام شبكة الحاسوب، يوجد نوعان من مواقع الويب: المواقع الساكنة **Static Websites** والمواقع التفاعلية **Dynamic Websites**. تعرض مواقع الويب الساكنة نفس المحتوى لجميع المستخدمين، وتكتب عادةً بلغة **HTML وCSS**، أما المواقع التفاعلية فتعرض محتوى متغير مع دعم لتفاعل المستخدمين وباستخدام البرمجة المتقدمة مع قواعد البيانات.

> المواقع الساكنة **Static Websites** تختص بعرض المعلومات في المقام الأول، وتأتي بعدد ثابت من الصفحات وبتخطيط محدد.
> عند عرض صفحة من هذا الموقع على متصفح الويب يكون المحتوى ثابتًا ولا يتغير وفق استجابة المستخدم.
> يتم تصميم مثل هذه المواقع بتنسيق HTML و CSS.
فعلى سبيل المثال، يعتبر موقع الويب الخاص بفريق قطر الوطني.

مواقع الويب التفاعلية **Dynamic Websites**:
> أكثر فاعلية نظرًا لسماحها للمستخدمين بالتفاعل مع المعلومات المعروضة في كل صفحة ويب.
> يتم تصميمها باستخدام تنسيقات HTML و CSS بالإضافة إلى لغة برمجة تسمح بتغيير المعلومات وفق استجابة المستخدمين.
> يتم جلب محتوى موقع الويب من قاعدة بيانات باستخدام البرمجة النصية.



المقارنة بين مواقع الويب الساكنة ومواقع الويب التفاعلية		
	مواقع الويب الساكنة	مواقع الويب التفاعلية
إنشاء الصفحات	تحتوي على عدد معين من الصفحات بتخطيط ثابت.	يمكنها إنشاء صفحات تلقائيًا مع إمكانية تغير التخطيط وفق الحاجة.
سرعة التحميل	سريعة في التحميل نظرًا لاحتوائها فقط على مقطع برمجي بسيط بلغة HTML والذي لا يتطلب أي معالجة تعتمد على الخادم.	قد تأخذ وقتًا في التحميل نظرًا لاعتمادها على معالجة البيانات من قِبل الخادم وإنشاء محتوى متغير.
التعامل مع البيانات	لا يتم استخدام أي قواعد بيانات.	يتم استخدام نظام إدارة قاعدة البيانات DBMS.
آلية إضافة وتحديث المحتوى	لإضافة محتوى جديد إلى موقع الويب أو لإضافة صفحات ويب جديدة يجب على المطور إنشاء وتحميل صفحات الويب الجديدة على الخادم.	لا يتم إنشاء كل صفحة ويب بشكل فردي ولكن من خلال التعليمات البرمجية والبيانات بالاعتماد على الخادم ومن قاعدة البيانات

نماذج الاستضافة Hosting Models

هناك أنواع من خوادم الاستضافة، نذكر منها: خادم الاستضافة الداخلي **On-premise Server**، وخادم الاستضافة السحابي **Cloud-based server**.

استخدام خادم الاستضافة الداخلي **On-premise server** يتطلب وجود خادم فعلي في إحدى منشآت الشركة، بينما يقدم خادم الاستضافة السحابي **Cloud Server** استضافة افتراضية سحابية تعتمد على وجود نظام تشغيل وخدمات أخرى مثبتة على خادم افتراضي، في الواقع تتطلب الاستضافة السحابية وجود مركز بيانات فعلي يختص بتوفير دعم للأجهزة والبرمجيات والصيانة للشركات التي لا تمتلك خوادم فعلية خاصة بها، وسيتم التطرق إلى ذلك تفصيلًا في دروس لاحقة.

المقارنة بين أنواع الاستضافة		
	خادم الاستضافة الداخلي	خادم الاستضافة السحابي
التكلفة	تتطلب الاستضافة استثمارا ماديًا كما تحتاج إلى الدعم والصيانة المتخصصة.	أسرع في الإعداد والتشغيل، مقابل رسوم لخدمات الاستضافة والصيانة، وعادة ما تكون أقل تكلفة.
الأمان	توفر درجة أكبر من التحكم بأمان البيانات وطرق حمايتها.	يقدم مزود الخدمة حلول لضمان أمن البيانات وحمايتها متمثلة بعمليات الصيانة والنسخ الاحتياطي، ... وغيرها
الموثوقية	لا تعتمد الخوادم الداخلية على توفر الاتصال بالإنترنت، وهو اعتبار مهم للعديد من الشركات عند الرغبة في الوصول إلى البيانات داخل المبنى نفسه.	تعتمد الاستضافة السحابية على الاتصال بالإنترنت اعتمادًا تامًا.
قابلية التوسع	تتطلب توفير الأجهزة والبرامج لتوسيع نطاقها عند الضرورة حسب متطلبات التوسع، وتحمل التكاليف الخاصة بذلك.	توفر المرونة في خيارات التوسع من حيث إمكانية وزيادة المساحة التخزينية والقدرة الاستيعابية للموقع فيما يتعلق بعدد الزوار وحجم المحتوى، وذلك مقابل رسوم الخدمة.

الخادم Server

الخادم هو جهاز حاسوب يقوم بتشغيل برنامج معين لخدمة المستخدمين المرتبطين به لغرض ما، واعتمادًا على الغرض المطلوب منه يقوم الخادم أيضًا بتشغيل الخدمات المناسبة. تطلق على الخوادم أسماء محددة تتعلق بالغرض الذي يتم استخدامها لأجله، فعلى سبيل المثال:

- يطلق على الخادم الخاص بصفحات الويب اسم خادم الويب **Web Server**، والذي يقوم بتشغيل البرنامج المناسب لخدمة الملفات الخاصة بموقع الويب المستضاف.
- أما إذا كان الخادم خاص بحسابات البريد الإلكتروني للمستخدمين فيطلق عليه اسم خادم البريد الإلكتروني **Mail Server**، والذي يقوم بتشغيل البرنامج المناسب لإرسال واستقبال رسائل البريد الإلكتروني.

يمكن إعداد الخوادم لإدارة وظيفة أو أكثر على الشبكة اعتمادًا على حجم شبكة المؤسسة، وعلى عدد المستخدمين، ومتطلبات إمكانية الوصول، وسعة التخزين... وما إلى ذلك.

وظائف الخادم	الوصف
خادم الملفات File server	يُمكن المستخدمين من تخزين الملفات في موقع مركزي، ويسمح بمشاركة الملفات والمجلدات بين مستخدمي الشبكة.
خادم الطباعة Print server	إدارة الاتصالات بين الطابعات وأجهزة حواسيب العملاء.
خادم التطبيقات Application server	يستضيف مواقع الإنترنت وتطبيقات الويب ويجعلها متاحة للمستخدمين، كما يتيح كذلك استضافة التطبيقات القائمة على الويب.
خادم البريد الإلكتروني Mail server	يوفر خدمات البريد الإلكتروني للمستخدمين لإرسال واستقبال الرسائل، ومن الضروري وجود مثل هذا الخادم ليتمكن المستخدمون من تبادل رسائل البريد الإلكتروني.
خادم المحطة الطرفية Terminal server	خادم المحطة الطرفية هو خادم أو جهاز شبكة يمكّن الاتصالات بأنظمة عميل متعددة من الاتصال بشبكة LAN دون استخدام مودم أو واجهة شبكة.
خادم اسم المجال (DNS) Domain Name Server	ينشئ أسماءً وفق تسلسل هرمي ويحولها إلى عناوين IP، و يوفر أيضًا وسيلة لتسمية مضيفات عناوين IP وتحديد موقع مضيفات IP عندما يتم الاستعلام عنها باسمها.
خادم بروتوكول التكوين الديناميكي للمضيف (DHCP) Dynamic Host Configuration Protocol Server	هو خادم شبكة يقوم بتوفير عناوين IP والبوابات الافتراضية وغيرها من معاملات الشبكة تلقائيًا وتعيينها لأجهزة العميل، ويعتمد على البروتوكول القياسي المعروف بـ DHCP للرد على طلبات العملاء.
خادم تدفق الوسائط Streaming Media Server	يوفر خدمات الوسائط بحيث يمكن للعملاء الوصول إلى تدفق الصوت والفيديو.

قاعدة البيانات Database

كما تعرفنا مسبقًا فإن قاعدة البيانات تستخدم لتخزين البيانات ومعالجتها، فهي نظام متكامل يتكون من معلومات وبرامج منظمة تجعل من عملية إدارة البيانات سهلة وآمنة.

في الماضي تكونت قواعد البيانات من ملفات بسيطة تتضمن عدة صفوف من البيانات المحفوظة، ولكنها أصبحت الآن أكثر تقدمًا مما سمح بإجراء استعلامات معقدة على جداول قاعدة البيانات المتعددة، كما يتم التحكم بها عن طريق نظام إدارة قاعدة البيانات (DBMS).

نظام إدارة قاعدة البيانات (DBMS): هو مجموعة من البرامج التي توفر للمستخدمين إمكانية الوصول بأمان إلى قاعدة البيانات ومعالجة البيانات واستخراج التقارير.

إيجابيات نظام إدارة قواعد البيانات (DBMS):

- يوفر مجموعة متنوعة من الخيارات ووظائف متعددة لتخزين واسترجاع البيانات بكفاءة.
- يسمح بتكاملية وأمان البيانات.
- يسمح للمستخدمين بالوصول المتزامن إلى البيانات ولكن يمكن لمستخدم واحد فقط تحرير نفس البيانات في الوقت نفسه.

تحديات نظام إدارة قواعد البيانات (DBMS):

- تكلفة الأجهزة والبرمجيات لنظام (DBMS) مرتفعة جدًا.
- تعتبر معظم نظم إدارة قواعد البيانات (DBMS) أنظمة معقدة وقد تتطلب كوادر مؤهلة ومدربة جيدًا للتعامل معها.

ما الذي نعنيه بالمعالجة الداخلية المعتمدة على الخادم Server-side Processing ؟

كما ذكرنا مسبقًا فإن مواقع الويب التفاعلية تستخدم البرمجة النصية والتي تعتمد بدورها على توفر خادم الاستضافة الداخلي أو الاستضافة السحابية، أو كليهما، في البرمجة النصية المعتمدة على خادم الاستضافة الداخلي، تتم معالجة المقاطع البرمجية من قبل هذا الخادم، ومن ثم يتم إنشاء صفحات الويب التي تُرسل إلى المستخدم.

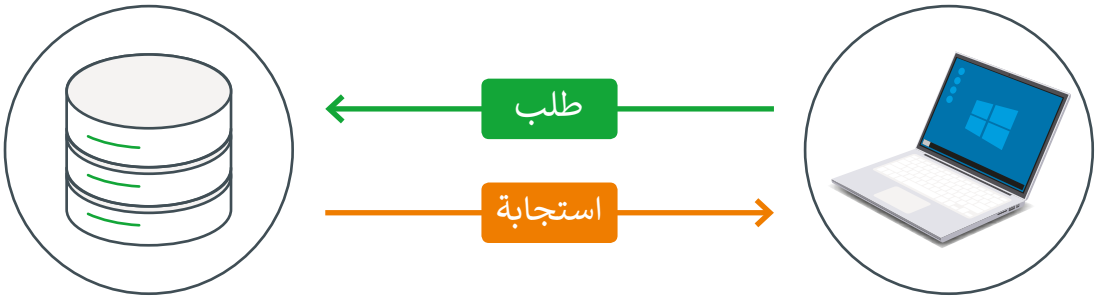
نموذج الاتصال الذي تم إنشاؤه يتكون من جزأين:

1. المعالجة الأمامية (Front-end) وهي خاصة بلغة HTML.
2. المعالجة الخلفية (Back-end) لمعالجة محتويات النموذج.

تتعامل المعالجة الأمامية مع عرض المعلومات للمستخدمين بينما تتعامل المعالجة الخلفية مع النماذج المرسلة، وفي حالتنا سنقوم بحفظ النماذج المُرسلة داخل خادم قاعدة البيانات MS SQL Server.

خادم MS SQL هو أداة لإدارة قاعدة البيانات تم تصميمها للخوادم.

يتطلب موقع الويب التفاعلي وظائف أكثر مما يمكن للصفحات الثابتة تقديمه، لذا فإن إنشاء صفحة لنموذج الاتصال تجعل منه تفاعليًا وبالتالي يتطلب وجود معالجة تعتمد على الخادم، حيث يقوم خادم الويب بتمرير معلومات النموذج إلى تطبيق صغير يتم تشغيله بواسطة نظام التشغيل ليقوم بمعالجة البيانات ثم يرجع صفحة ويب أو رسالة تأكيد.



لغة البرمجة النصية المعتمدة على الخادم

البرمجة النصية المعتمدة على الخادم هي تقنية يتم فيها تشغيل برنامج نصي لمقاطع برمجية script على خادم ويب لإنشاء صفحات الويب بشكل تلقائي، وهذا يساعد في إنشاء صفحة ليتم إرسالها إلى المتصفح الخاص بالمستخدم، وفي هذه الحالة يتم تنفيذ البرامج النصية على الخادم فقط وليس إلى جانب المستخدم.

بيئة الخادم البرمجية Node.js

Node.js هي بيئة خادم برمجية تسمح لنا بإنشاء تطبيقات الخادم والشبكات باستخدام مقاطع JavaScript، وتسمح هذه البيئة بإنشاء محتوى ديناميكي للصفحة من خلال ربط أوامر JavaScript بقاعدة بيانات على خادم الويب.

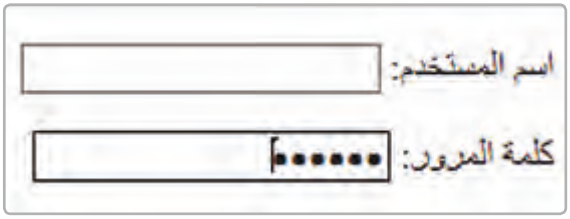
عادةً ما تخدم **Node.js** مجموعة من الإجراءات والكائنات الخاصة بـ **JavaScript**، فهي في الواقع بيئة عمل أو مترجم خاص لتشغيل أي برنامج **JavaScript**، وتستخدم عادةً مجموعة من المكتبات أو الوحدات القياسية **Modules** التي تنفذ وظائف أساسية مختلفة.

لاحظ النموذج التالي:

يقوم بتحديد مكان إرسال بيانات النموذج عند إرسال النموذج.

```
<form action="http://www.myserver.qa/comments.php" method="get" >
   <p>اسم المستخدم:
      <input type="text" name="username" /></p>
   <p>كلمة المرور:
      <input type="password" name="password"/></p>
</form>
```

يقوم بتحديد إرسال بيانات النموذج كمتغير URL.

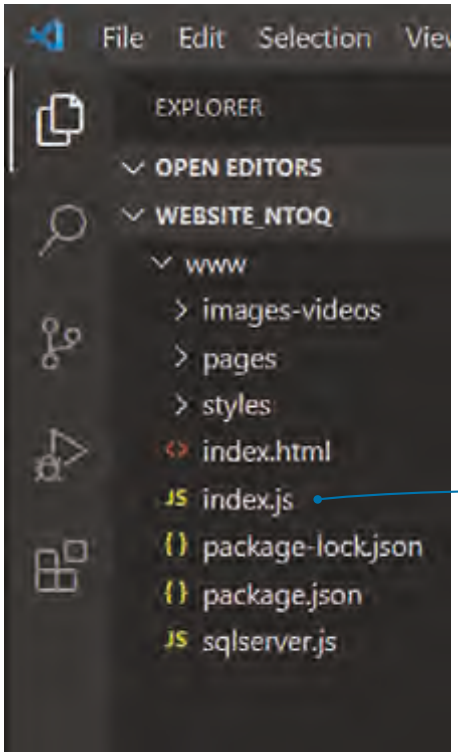


كيف يعمل نموذج HTML؟

- يقوم المستخدم بزيارة صفحة ويب تحتوي على نموذج.
- يعرض متصفح الويب نموذج HTML.
- يقوم المستخدم بتعبئة النموذج وإرسال البيانات.
- يرسل المتصفح بيانات النموذج المعبأة إلى خادم الويب.
- يقوم البرنامج الموجود على خادم الويب بمعالجة بيانات النموذج.
- يتم إرسال صفحة استجابة إلى المتصفح.

لقد جهزنا مقطعًا برمجيًا يحتوي على دالة سنستخدمها في ملف **JavaScript** الرئيس **(index.js)** والذي سينشئ اتصال إلى الخادم ويدخل بيانات النموذج في قاعدة البيانات.

لإنشاء كود مثل هذا فإنك تحتاج إلى مهارات تطوير خلفية متقدمة.



قبل الاستمرار في كتابة المقاطع البرمجية المعتمدة على الخادم فلنضف الحدث **action** وسمة الإجراء **method** إلى نموذج **HTML** الخاص بنا.

تستخدم سمة الإجراء method مع وسم النموذج form للإشارة إلى الطريقة التي سيتم بها تمرير البيانات المدخلة إلى الخادم، أما قيمة سمة الإجراء "get" فتجعل بيانات النموذج مرئية إلى الرابط التشعبي URL.

.html

```
<h1> اتصل بنا</h1>
<form action="http://www.myserver.qa/comments.php" method="get" >
  <fieldset>
    <legend>اترك رسالتك</legend>
    <label for="name">الاسم الأول</label><br/>
    <input id="name" name="name" type="text" placeholder="اكتب اسمك الأول ..."><br/>
    <label for="Last name"> الاسم الأخير</label><br/>
    <input id="lname" name="lname" type="text" placeholder="اكتب اسمك الأخير..."><br/>
    <label for="email">البريد الإلكتروني</label><br/>
    <input id="email" name="email" type="email" placeholder="أدخل بريدك الإلكتروني..."><br/>
    <label for="message">الر سالة</label><br/>
    <textarea placeholder="رسالتك.." ></textarea><br/>
    <input type="submit" value="Submit"/>
  </fieldset>
</form>
```

تحدد سمة id اسمًا فريدا خاصًا بعنصر HTML.

سننشئ الآن المقطع البرمجي الخاص بالملف **(index.js)** والذي سيقوم بما يلي:

إعداد موقع الويب الخاص بنا وإرسال البيانات المدخلة من المستخدم إلى قاعدة بيانات بحيث يتم حفظها وتنظيمها.

في البداية علينا استيراد وحدة قياسية **(Module)**، والتي هي عبارة عن مكتبة أو ملف بلغة **JavaScript** يمكننا استيراده إلى المقطع البرمجي باستخدام الدالة **require()** في بيئة **node.js**.

يتم استدعاء الوحدة القياسية **Module** باسمها باستخدام **Express framework** وهو تطبيق ويب خاص بـ **Node.js** يوفر مجموعة من الميزات لتطوير مواقع وتطبيقات الويب والهواتف الذكية.

نحتاج إلى استخدام express من node.js لكي نتمكن من إرسال البيانات إلى الخادم.

يخبر هذا السطر البرنامج باستخدام المقاطع البرمجية من ملف sqlserver.js الموجود والمتاح للاستخدام.

.javascript

```
var express = require('express');

var sqlserver = require('./sqlserver');

var app = express();

app.use(express.static('www'));
```

باستخدام هذه الطريقة نقوم بإنشاء إطار express جديد الذي يمكننا تعيينه إلى متغير والتفاعل معه.

يبحث التطبيق في هذا المجلد عن ملفات بصيغة HTML أوCSS أو أي ملفات JavaScript موجودة.

لنستكمل العمل مع الدوال.

يقوم هذا السطر البرمجي باستدعاء الدالة GET عندما نضغط على زر الإرسال في النموذج.

.javascript

```
app.get('/requests/form', function (req, res) {
```

هذه هي بيانات الدخول إلى الخادم للتمكن من تخزين المدخلات إلى قاعدة البيانات الموجودة عليه.

إذا كانت قاعدة بيانات SQL تعمل على نفس الجهاز، فاستخدم localhost كملقم وإلا ستحتاج إلى تغييرها إلى اسم خادم SQL المناسب.

```
    var user = "User01";
    var password = "123456";
    var server = "localhost";
    var database = "Qatar-Web-App"
```

قد تختلف بيانات الدخول من خادم لآخر حسب إعدادات الأمان الخاصة به، سيساعدك معلمك في الحصول على هذه البيانات داخل المعمل.

باستخدام الوظيفة التالية من sqlserver.js يتم إعداد اتصال بقاعدة البيانات باستخدام بيانات الدخول المذكورة أعلاه.

.javascript

```
    var server_config = sqlserver.configure_server(user,
password, server, database);
```

باستخدام الوظيفة الثانية من **sqlserver.js** يتم إدراج بياناتك في قاعدة البيانات.

.javascript

```
    sqlserver.insert_data_to_database(server_config, req.query.
name, req.query.lname, req.query.email, req.query.message);

    res.send('GET request successfully read by back end');
});
```

هذه رسالة يتم عرضها عند اكتمال العملية المذكورة بنجاح.

```
app.listen(80, function () {
    console.log('listening at port 80');
});
```

هذه هي نقطة بداية التطبيق.

يحتوي الملف المعطى للطالب الذي تم إعداده sqlserver.js على اسم جدول قاعدة البيانات ويجب تغييره وفقًا لجدول قاعدة بيانات الطالب المحدد. ابحث عن سطر الاستعلام إلى الجدول وقم بتغييره وفقًا لذلك لكل طالب. على سبيل المثال، نستخدم Table_01 للطالب الأول.

ثم سنبدأ التطبيق باستخدام **Terminal** داخل برنامج **Visual Studio Code** لتنفيذ الأوامر التي تسمح لك بالقيام بمهمة تشغيل الأدوات المذكورة سابقًا وإنشاء الاتصال مع الخادم وتشغيل تطبيق الويب لإدخال البيانات.

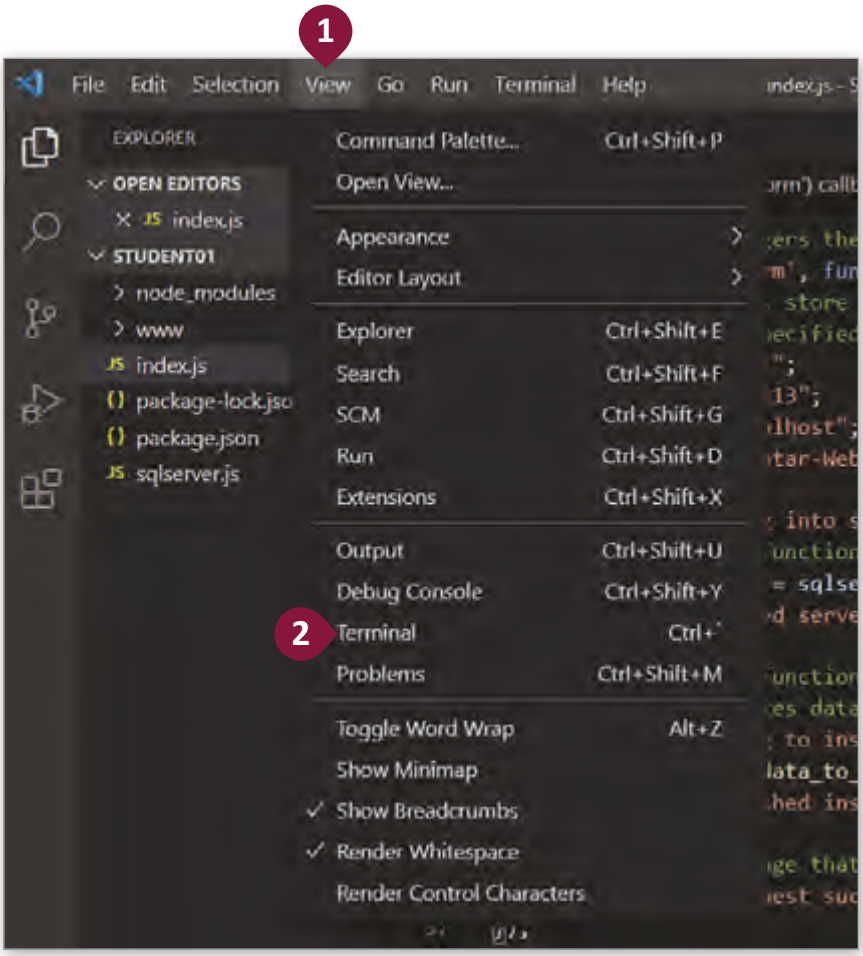
لتشغيل تطبيق الويب وبدأ إدخال البيانات:

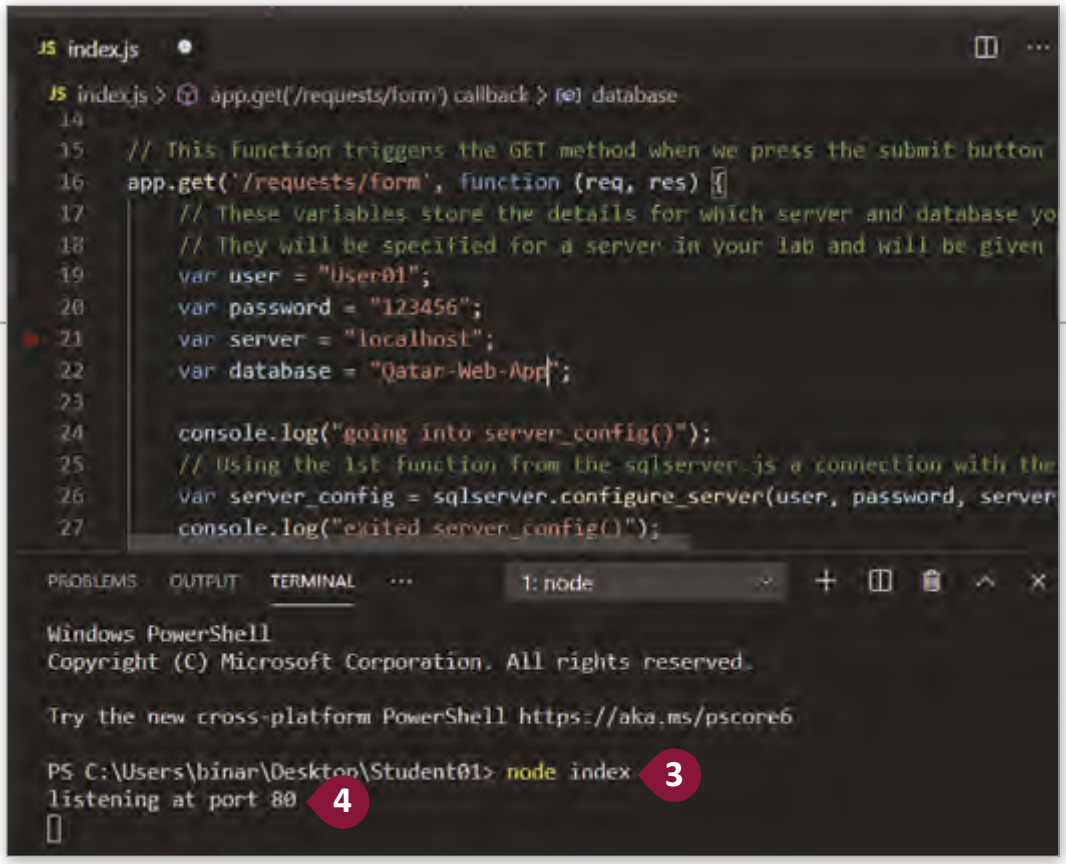
> من قائمة الأوامر اضغط **View** (عرض) ❶ ثم اضغط **Terminal**.❷

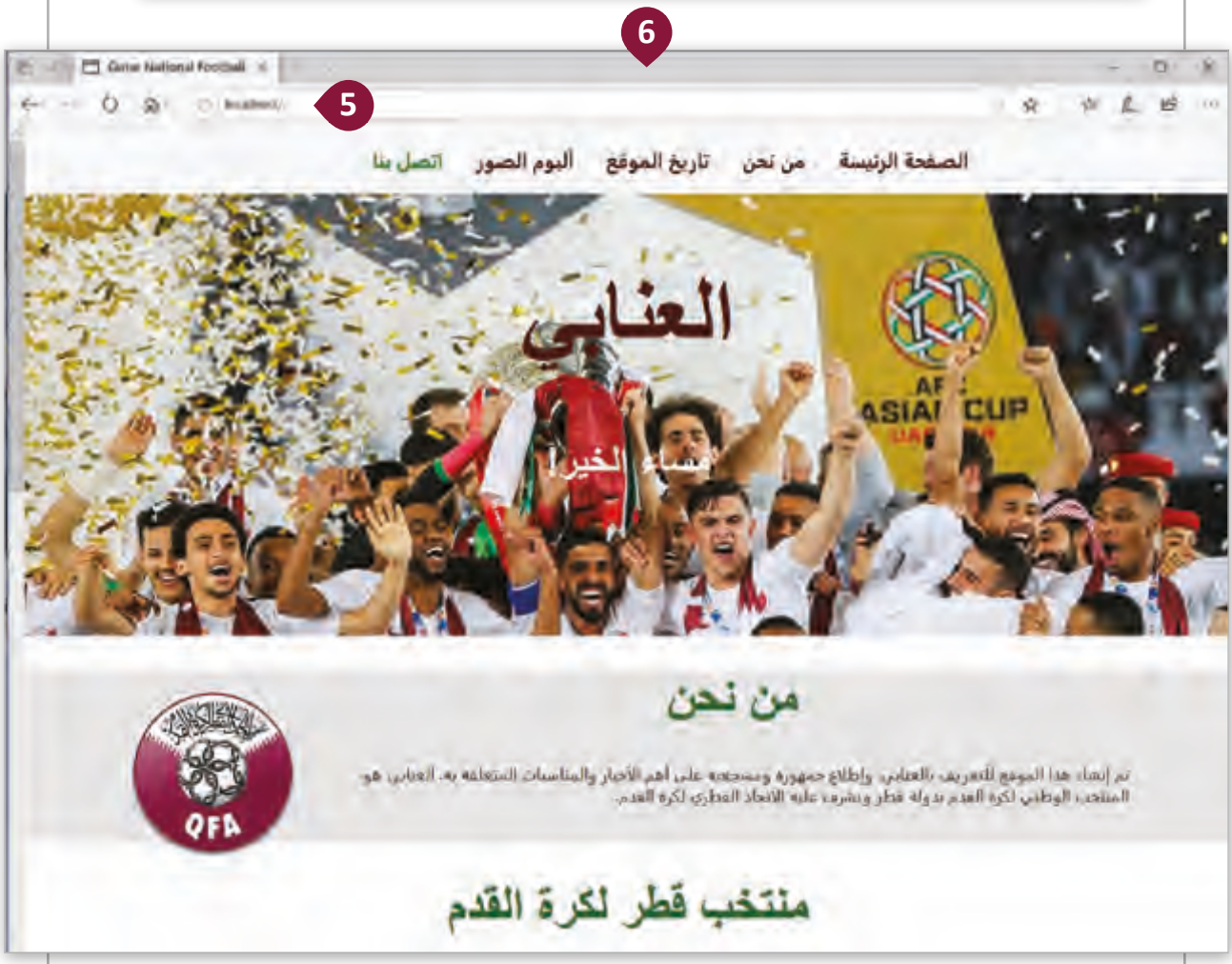
> في **Terminal** اكتب: **node index**.❸

> عند ظهور الرسالة "**listening at port 80**", ❹ نتقل إلى متصفح الويب واكتب في مربع بحث عنوان URL كلمة: **localhost**. ❺

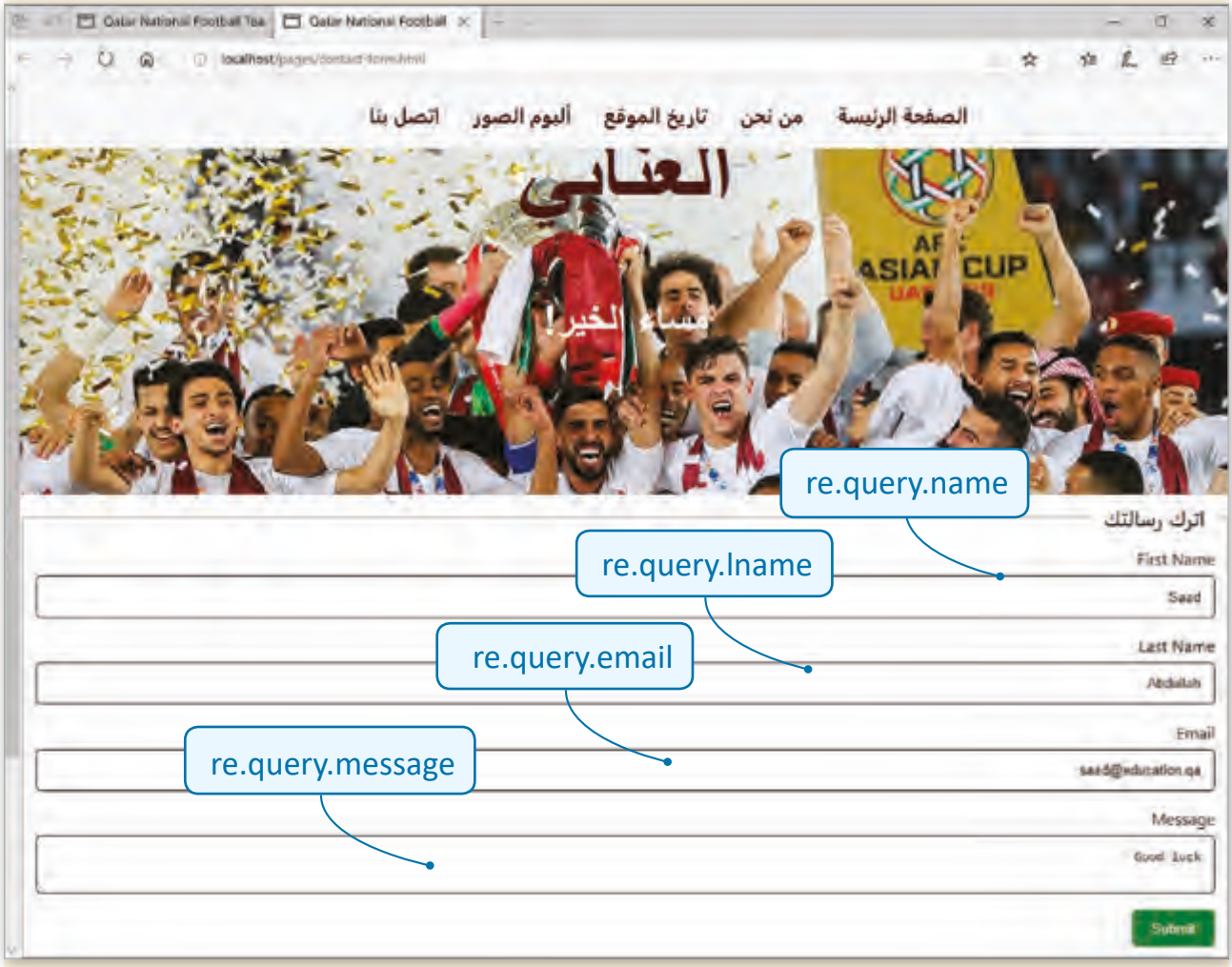
> يفتح موقع الويب الخاص بك. ❻

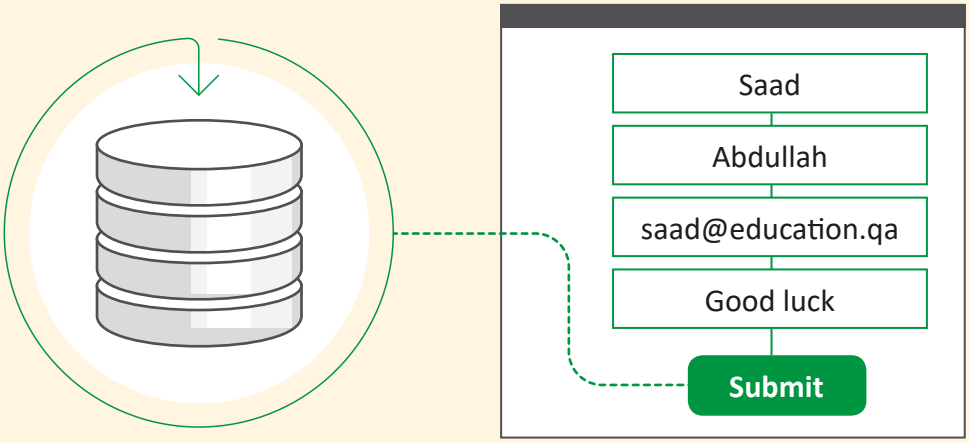



index.js
index.js > app.get('/requests/form') callback > database
// This function triggers the GET method when we press the submit button
app.get('/requests/form', function (req, res) {
// These variables store the details for which server and database yo
// They will be specified for a server in your lab and will be given
var user = "User01";
var password = "123456";
var server = "localhost";
var database = "Qatar-Web-App";
console.log("going into server_config()");
// Using the 1st function from the sqlserver.js a connection with the
var server_config = sqlserver.configure_server(user, password, server
console.log("exited server_config()");
PROBLEMS
OUTPUT
TERMINAL
1: node
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\binar\Desktop\Student01> node index
3
listening at port 80
4

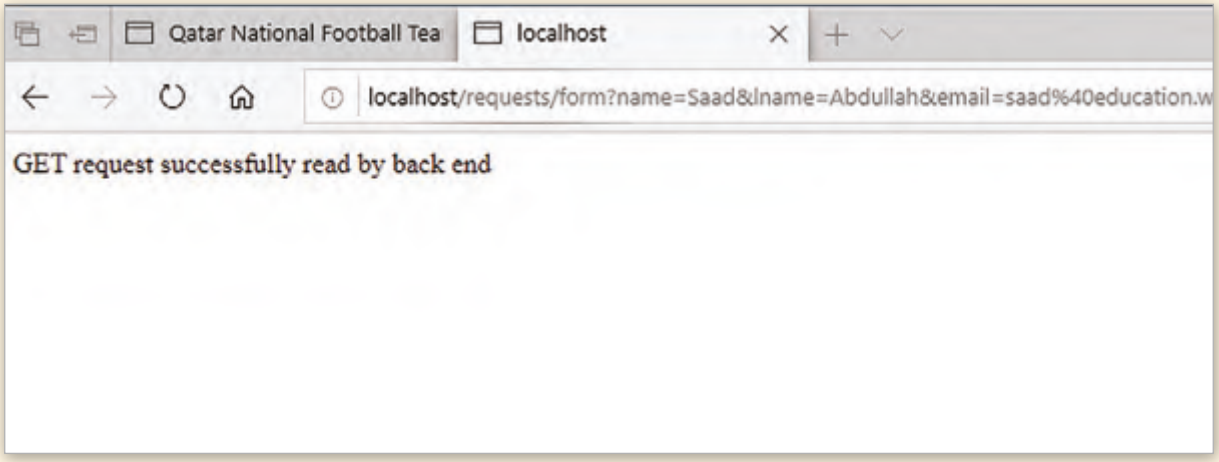

6
5
الصفحة الرئيسة
من نحن
تاريخ الموقع
اليوم الصور
اتصل بنا
العنابي
مساء الخير!
من نحن
منتخب قطر لكرة القدم

الآن وبعد فتح النموذج وتعبئة بياناته، سيتم حفظ تلك البيانات في قاعدة البيانات الخاصة بنا.





بعد ضغط زر الإرسال **submit** الخاص بالنموذج سيتم إرسال البيانات لتحفظ في قاعدة البيانات وستظهر الرسالة التالية.



سيتم عرض ما يلي في **VSC** وفي **Terminal**:

```
going into server_config()
exited server_config()
going to insert form data
tedious deprecated The default value for `config.options.enableArithAbort` will
change from `false` to `true` in the next major version of `tedious`. Set the va
lue to `true` or `false` explicitly to silence this message. node_modules\mssql\
lib\tedious\connection pool.js:61:23
finished inserting form data
Inserting the following values into the table...
name: Saad
lname: Abdullah
email: saad@education.qa
message: Good luck
```

إذا أردنا التحقق من البيانات المخزنة في قاعدة البيانات سنقوم بفتح **SQL Server Management Studio**.

للتحقق من البيانات المخزنة في قاعدة البيانات:

> قم بتشغيل **SQL Server Management Studio** واختر **SQL Server Authentication**. ❶

> أدخل بيانات اعتماد تسجيل الدخول الخاصة بك، على سبيل المثال **User01** كاسم مستخدم و **123456** ككلمة مرور ❷ واضغط **Connect**. ❸

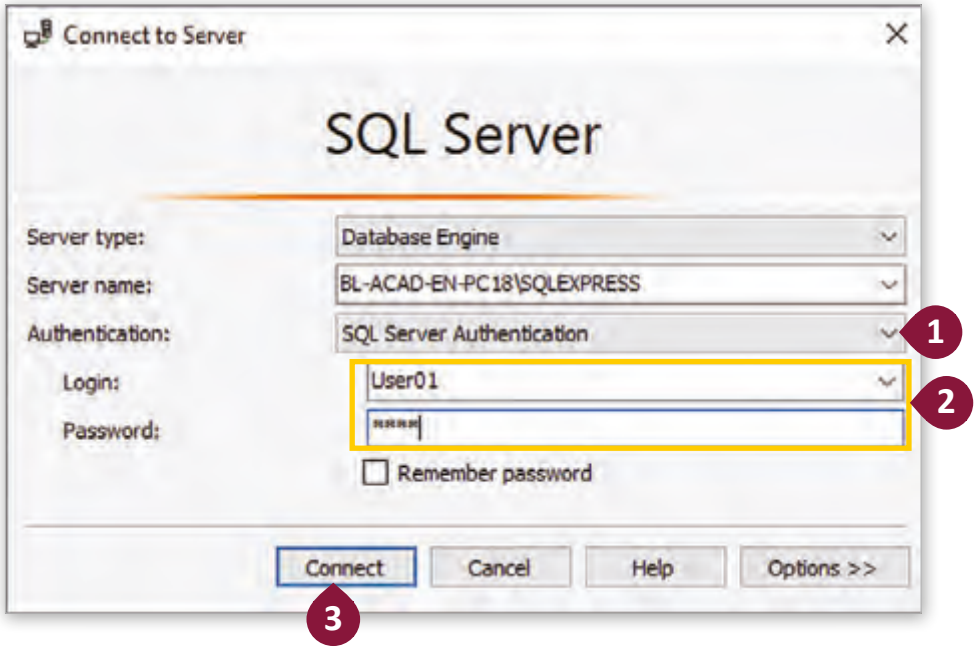
> من **Object Explorer**، اضغط ضغطًا مزدوجًا على **Databases**. ❹

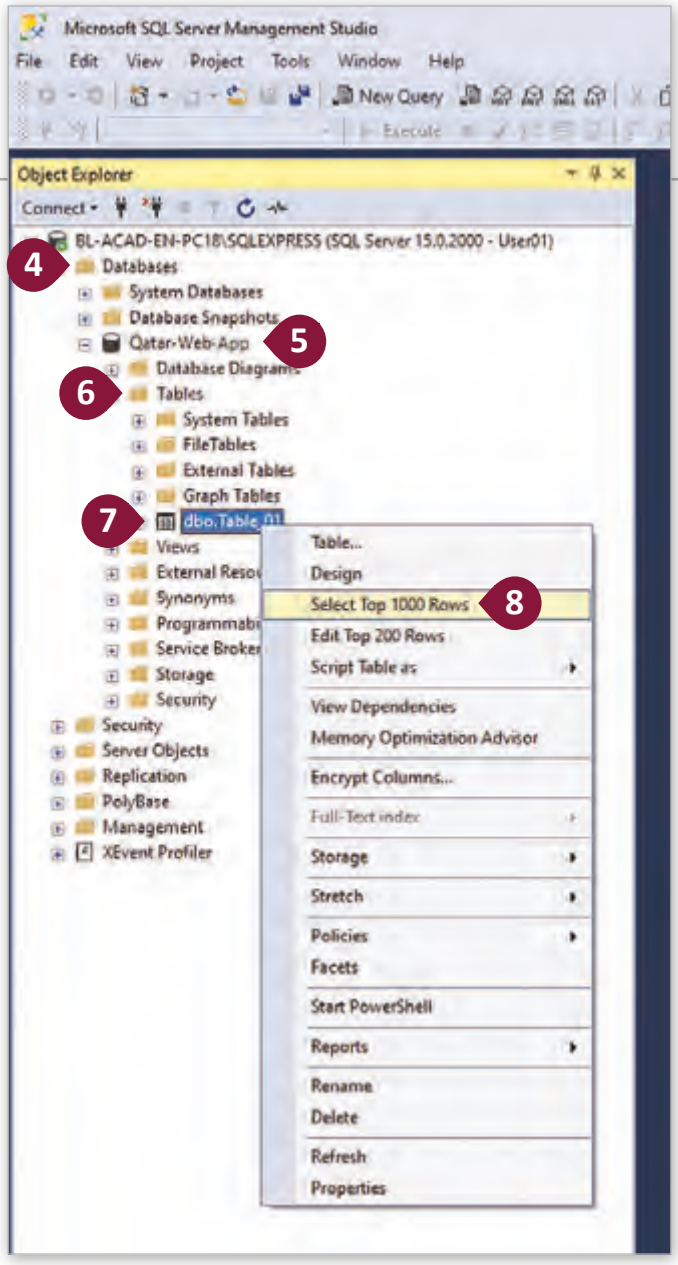
> اضغط ضغطًا مزدوجًا على قاعدة بيانات **Qatar-Web-App**. ❺

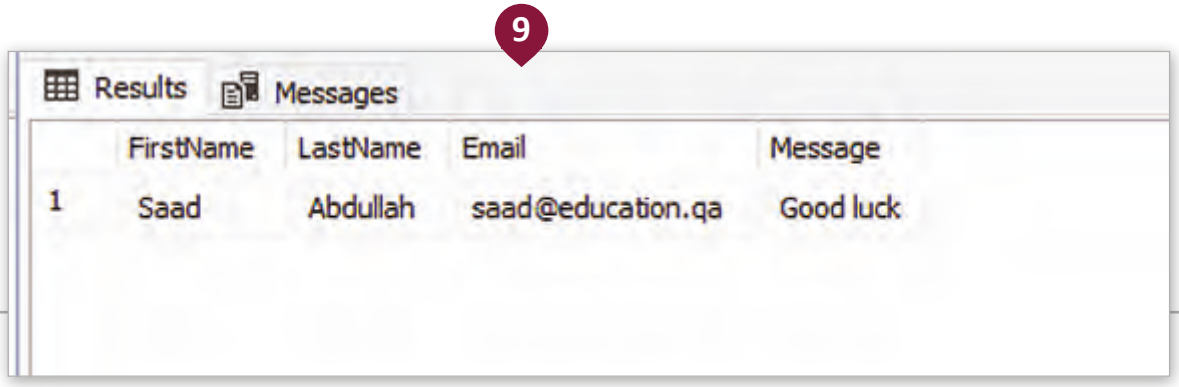
> اضغط ضغطًا مزدوجًا على **Tables**. ❻

> اضغط بزر الأيمن على **dbo.Table_01** ❼ واضغط **Select Top 1000 Rows**. ❽

> ستجد أن البيانات التي أدخلتها في نموذج اتصل بينا قد تم تخزينها في قاعدة البيانات. ❾




Microsoft SQL Server Management Studio
File Edit View Project Tools Window Help
New Query
Object Explorer
Connect
BL-ACAD-EN-PC18\SQLEXPRESS (SQL Server 15.0.2000 - User01)
Databases
System Databases
Database Snapshots
Qatar-Web-App
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Table_01
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
Table...
Design
Select Top 1000 Rows
Edit Top 200 Rows
Script Table as
View Dependencies
Memory Optimization Advisor
Encrypt Columns...
Full-Text index
Storage
Stretch
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
4
5
6
7
8


9
Results
Messages
FirstName
LastName
Email
Message
1
Saad
Abdullah
saad@education.qa
Good luck

1 ضع علامة ✓ أمام العبارة الصحيحة وعلامة ✗ أمام العبارة الخطأ.

.1	لموقع الويب الساكن عدد معين من الصفحات ويستخدم HTML وCSS والبرمجة النصية.	○
.2	من نماذج استضافة المواقع، الاستضافة على الخوادم الداخلية.	○
.3	نظام إدارة قاعدة البيانات DBMS هو لغة تعتمد على الخادم تسمح لنا بالتحكم وإدارة البيانات في قاعدة بيانات.	○
.4	خادم الطباعة يوفر خدمات الوسائط بحيث يمكن المستخدم من الوصول إلى تدفق الصوت والفيديو.	○

2 حدد الأخطاء في المقاطع البرمجية التالية وقم بتصحيحها.

1. يحدد هذا السطر موقع إرسال بيانات النموذج عند الضغط على زر الإرسال submit.

```
<form action=http://www.myserver.qa/comments.php method="get">
```

_____

2. يخبر هذا السطر البرنامج لاستخدام حقيبة من node.js.

```
var express = request('express');
```

_____

3. يستخدم هذا السطر لعرض رسالة عند اكتمال العملية بنجاح.

```
require.send('GET request successfully read by back end');
```

_____

3

أجب عن الأسئلة التالية.

1. ما أوجه الاختلاف بين مواقع الويب الساكنة والتفاعلية؟

2. لماذا نستخدم نظام إدارة قاعدة البيانات DBMS؟

3. ما هو الخادم Server؟ ولماذا يُستخدم؟

_____

_____

_____

_____

_____

_____

_____

4

افتح المجلد الخاص بموقع الويب الخاص بالمغامرة Adventure وقم بإضافة نموذج تفاعلي كالتالي:

> افتح المجلد في برنامج Visual Studio Code ثم افتح صفحة "Contact_form.html".

> أجرِ التغييرات اللازمة لتحديد موقع إرسال بيانات النموذج عند الضغط على زر Submit.

> أنشئ مقطعًا برمجيًا باستخدام node.js لحفظ البيانات في قاعدة البيانات.

> احفظ التغييرات ليصبح موقع الويب التفاعلي الخاص بك جاهزًا.

مشروع الوحدة

العنوان: إنشاء موقع ويب تعريفي لشخصية من اختيارك

الوصف: أنشأنا سابقا موقع ويب خاص بأحد الشخصيات المشهورة كعالم أو كرياضي، في هذا المشروع سنستكمل عملية تصميم هذا الموقع.

الأدوات: محرر Visual Studio Code

خطوات التنفيذ:

أنشئ مجلدًا في مستنداتك باسم "Tribute_website".

افتح محرر Visual Studio Code ثم افتح المجلد الذي أنشأته.

> طبق قواعد CSS التي تعلمتها لجعل موقعك جذابًا ومتجاوبًا بإضافة استعلامات الوسائط اللازمة.

> استخدم JavaScript لإضافة أيقونة القائمة الخاصة بالتصفح.

> في النهاية، استخدم مقطعًا برمجيًا يسمح لنا بحفظ البيانات الخاصة بالنموذج عند الضغط على زر الإرسال submit إلى قاعدة بيانات.

> احفظ التغييرات.

ماذا تعلمت

تعلمت في هذه الوحدة:

- المقصود بـ JavaScript وأهميتها في موقع الويب.
- استخدام JavaScript في تعديل محتوى عناصر HTML.
- العرض باستخدام مقاطع JavaScript .
- المقصود بتقنيات تحسين موقع البحث SEO.
- المقصود بقاعدة البيانات والخوادم.
- أوجه الاختلاف بين مواقع الويب الساكنة والتفاعلية.
- المقصود بالبرمجة النصية المعتمدة على الخادم.
- استخدام Node.js لحفظ البيانات في قاعدة البيانات.

المصطلحات:

الدرس 1	تصميم الويب المتجاوب Responsive web-design	حجم الشاشة Screen-size	استعلامات الوسائط Media queries
	إطار العرض Viewport	شريط التصفح Navigation bar	شبكة نظير لنظير Peer-to-Peer
	محاكاة الجهاز Emulation Device		

الدرس 2	المتغيرات Variables	الجمل Statements	الدوال Functions
	تحسين محرك البحث SEO	كائن Object	

الدرس 3	خادم Server	قاعدة بيانات Database	نظام إدارة قاعدة البيانات Database Management System
	نظام تشغيل Operating System	موقع ويب ساكن Static Website	موقع ويب تفاعلي Dynamic Website
	لغة برمجة نصية تعتمد على الخادم Server-side Scripting Language	الاستضافة Hosting	وحدة قياسية Module
	خادم الملفات File server	خادم الطباعة Print server	خادم التطبيقات Application server
	خادم البريد الإلكتروني Mail server	خادم المحطة الطرفية Terminal server	خادم اسم المجال (DNS) Domain Name Server
	خادم بروتوكول التكوين الديناميكي للمضيف (DHCP) Dynamic Host Configuration Protocol Server	خادم تدفق الوسائط Streaming Media Server	

الصف	الصفحة	الملاحظة

الصف	الصفحة	الملاحظة

الصف	الصفحة	الملاحظة

الصف	الصفحة	الملاحظة

تكنولوجيا المعلومات المستوى الحادي عشر المسار التكنولوجي
الفصل الدراسي الأول 2021 - 2020
كتاب الطالب

تم النشر بواسطة: **دار النشر MM Publications**
www.mmpublications.com
info@mmpublications.com
المكاتب
المملكة المتحدة، الصين، قبرص، اليونان، كوريا، بولندا، تركيا، الولايات المتحدة الأمريكية، الشركات المنتسبة والممثلين في جميع أنحاء العالم.

حقوق التأليف والنشر © 2020 لشركة **Binary Logic SA**
تم النشر بواسطة دار النشر **MM Publications** بموجب اتفاقية مُبرمة مع شركة **Binary Logic SA**.

جميع الحقوق محفوظة. لا يجوز نسخ أي جزء من هذا المنشور أو تخزينه في أنظمة استرجاع البيانات أو نقله بأي شكل أو بأي وسيلة إلكترونية أو ميكانيكية أو بالنسخ الضوئي أو التسجيل أو غير ذلك دون إذن كتابي من الناشرين وفقًا للعقد المُبرم مع وزارة التعليم والتعليم العالي بدولة قطر.

يُرجى ملاحظة ما يلي: يحتوي هذا الكتاب على روابط إلى مواقع ويب لا تُدار من قبل شركة **Binary Logic**. ورغم أنَّ شركة **Binary Logic** تبذل قصارى جهدها لضمان دقة هذه الروابط وحداثتها وملائمتها، إلا أنها لا تتحمل المسؤولية عن محتوى أي مواقع ويب خارجية.

إشعار بالعلامات التجارية: أسماء المنتجات أو الشركات المذكورة هنا قد تكون علامات تجارية أو علامات تجارية مُسجَّلة وتُستخدم فقط بغرض التعريف والتوضيح ولا توجد أي نية لانتهاك الحقوق. تنفي شركة **Binary Logic** وجود أي ارتباط أو رعاية أو تأييد من جانب مالكي العلامات التجارية المعنيين. تُعد **Microsoft** و **Windows** و **Windows Live** و **Outlook** و **Access** و **Excel** و **PowerPoint** و **OneNote** و **Skype** و **OneDrive** و **Bing** و **Edge** و **Internet Explorer** و **Kodu Game Lab** و **MakeCode** و **Office 365** علامات تجارية أو علامات تجارية مُسجَّلة لشركة **Microsoft Corporation**. وتُعد **Google** و **Gmail** و **Chrome** و **Google Docs** و **Google Drive** و **Google Maps** و **Android** و **YouTube** علامات تجارية أو علامات تجارية مُسجَّلة لشركة **Google Inc**. وتُعد **Apple** و **iPad** و **iPhone** و **Pages** و **Numbers** و **Keynote** و **iCloud** و **Safari** علامات تجارية مُسجَّلة لشركة **Apple Inc**. تم تطوير **Scratch** من قبل مجموعة **Lifelong Kindergarten Group** في مختبر **MIT Media Lab**، كما أن اسم **Scratch** وشعار **Scratch** و **Scratch Cat** علامات تجارية مُسجَّلة مملوكة من قبل **Scratch Team**. وتُعد **LEGO®** و **MINDSTORMS®** علامات تجارية أو علامات تجارية مُسجَّلة لشركة **The LEGO Group**. وتُعد **Python** وشعارات **Python** علامات تجارية أو علامات تجارية مُسجَّلة لمؤسسة **Python Software Foundation**. وتُعد **LibreOffice** علامة تجارية مُسجَّلة لشركة **Document Foundation**.

تم الإنتاج في الاتحاد الأوروبي

الرقم الدولي المعياري للكتاب: **978-618-05-4672-9** N2005002013-17464