

الحادي عشر تكنولوجيا -مراجعة عامة – لمادة علوم الحاسب

1- ما المقصود بهياكل البيانات وماهي أنواعها؟

هياكل البيانات: هي وسيلة لتخزين وتنظيم البيانات في الذاكرة بحيث يمكن استخدامها بكفاءة، ولها أمثلة متعددة، أبرزها القوائم والصفوف والمصفوفات والقاموس وغيرها.

2- عدد أنواع البيانات في لغة بايثون؟

a. Numbers

b. البيانات نصية String

c. بيانات على شكل قائمة List

d. بيانات على شكل صفوف Tuple

e. قاموس بيانات Dictionary

3- ما المقصود بالمتغير؟

المتغير هو اسم رمزي يشير لمكان في ذاكرة الحاسوب لتخزين البيانات أثناء تنفيذ البرنامج.

4- عرف المصطلحات التالية:

1- القائمة List : القائمة في بايثون عبارة عن سلسلة من العناصر من نفس النوع أو من أنواع مختلفة مثل

النصوص والأعداد الصحيحة والأعداد العشرية وغيرها.

- مثال لبيانات على شكل قائمة List :

`list = ["banana", "pineapple", "cherry"]`

5- الصف Tuple : يعتبر Tuple مجموعة مرتبة من البيانات بحيث يمكن أن تكون القيم المخزنة بداخله

من أي نوع ولكن لا يمكن تغييرها.

- مثال لبيانات على شكل صف Tuple :

`PersonInfo = ("Khaled", "khaled@edu.qa", 1234)`

6- المصفوفة الأحادية: تشبه هذه المصفوفة القوائم، فكل عنصر في المصفوفة لديه فهرس يقوم بتحديد موقعه داخل المصفوفة، ويمكننا الوصول الى عناصر كل مصفوفة عن طريق كتابة اسم المصفوفة والرقم التسلسلي للعنصر بين قوسين مربعين. المصفوفة يجب أن تحتوي على بيانات من نفس النوع.

مثال لبيانات على شكل مصفوفة Array :

الرموز المحددة لنوع عناصر المصفوفة	
نوع البيانات	رمز النوع
(عدد صحيح)	i
(عدد عشري)	d
(حرف أو رمز)	u

```
from array import *  
my_array = array ('i', [85,92,80,81])
```

7- اذكر وظيفة كل دالة من الدوال التالية:؟

- Len() : عدد العناصر الموجودة في قائمة أو عدد الحروف لمتغير نصي أو عدد أرقام لمتغير عددي
- Sum() : إيجاد مجموع عدة عناصر.
- Max() : إيجاد أكبر عنصر موجود في قائمة.
- Min() : إيجاد أصغر عنصر موجود في قائمة.

8- مزيد من الدوال الجاهزة المستخدمة مع القوائم: ص 19

الدوال المستخدمة مع القوائم	
الوصف	العملية
إضافة العنصر x في آخر القائمة.	listName.append(x)
يزيل العنصر x من القائمة.	listName.remove(x)
تحسب عدد مرات ظهور العنصر x داخل القائمة.	listName.count(x)
فرز العناصر في القائمة.	listName.sort()
عكس عناصر القائمة.	listName.reverse()
حذف جميع العناصر من القائمة.	listName.clear()

9- قارن بين وظيفة دالة `listName.append(x)` ووظيفة دالة `listName.remove(x)` عند استخدامهم مع القوائم.
(انظر الجدول السابق في الفقرة 8)

10- قارن بين وظيفة دالة `listName.sort()` ووظيفة دالة `listName.reverse()` عند استخدامهم مع القوائم.

11- اختر الإجابة الصحيحة في كل مما يلي:

<input checked="" type="checkbox"/>	0	1. ترقيم القائمة يبدأ من:
<input type="checkbox"/>	1	
<input type="checkbox"/>	-1	

<input type="checkbox"/>	<code>sum()</code>	2. الدالة التي تُرجع أدنى عنصر في القائمة هي:
<input checked="" type="checkbox"/>	<code>min()</code>	
<input type="checkbox"/>	<code>max()</code>	

<input type="checkbox"/>	<code>sum()</code>	3. الدالة التي تُرجع أكبر عنصر في القائمة هي:
<input type="checkbox"/>	<code>min()</code>	
<input checked="" type="checkbox"/>	<code>max()</code>	

<input type="checkbox"/>	<code>sum()</code>	4. لإضافة عنصر إلى القائمة نستخدم الدالة:
<input checked="" type="checkbox"/>	<code>append()</code>	
<input type="checkbox"/>	<code>count()</code>	

12- `foods = ["salad", "fruit", "vegetables", "dairy"]`

طباعة أول عنصر في القائمة `Print(foods[0])`

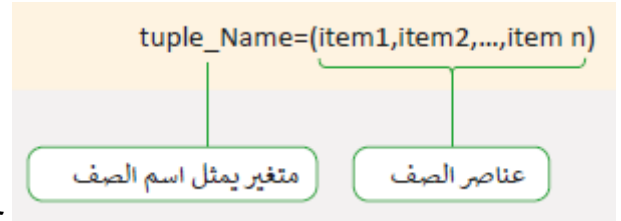
طباعة العنصر الرابع والأخير في القائمة `Print(foods[3])`

طباعة عدد عناصر القائمة مستخدماً دالة `len` `Print(len(foods))`

13- ما المقصود بالصفوف Tuples ؟

يعتبر Tuple مجموعة مرتبة من البيانات بحيث يمكن أن تكون القيم المخزنة بداخله من أي نوع ولكن لا يمكن تغييرها.

14- اكتب الصيغة العامة للصف Tuple ؟



حيث يتم وضع عناصر الصف بين القوسين () ويفصل بينهما

بفاصلة.

15- ما هو وجه الاختلاف بين القوائم Lists والصفوف Tuples ؟ (الكتاب ص 37)

← يتم تعريف الصفوف بإحاطة العناصر داخل القوسين () بدلاً من [].

← تعتبر الصفوف Tuples غير قابلة للتغيير.

16- صغ علامة ✓ أمام العبارة الصحيحة وعلامة ✗ أمام العبارة الخاطئة:

<input checked="" type="checkbox"/>	1. يجب أن تكون جميع العناصر في الصف Tuple من نفس نوع البيانات.
<input checked="" type="checkbox"/>	2. الصفوف Tuple هو من نوع هياكل البيانات غير القابلة للتغيير.
<input checked="" type="checkbox"/>	3. يتم ترقيم مواقع العناصر في الصف بدءاً بالرقم (1).
<input checked="" type="checkbox"/>	4. يتم إحاطة العناصر داخل الصف Tuple بأقواس مربعة.
<input checked="" type="checkbox"/>	5. لا يمكننا إضافة عناصر جديدة في Tuple بعد إنشائه.

17- ما المقصود بالمصفوفة أحادية الأبعاد 1D ؟

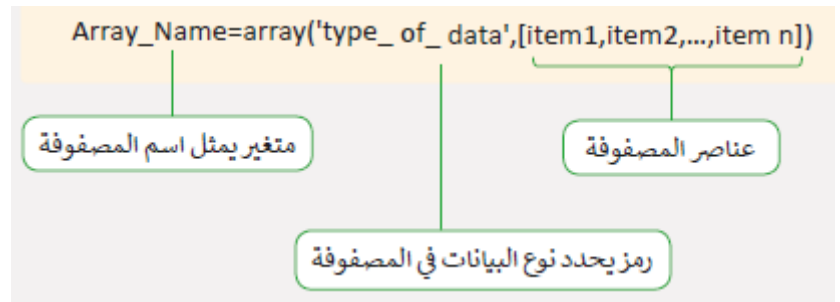
- المصفوفة الأحادية: تشبه هذه المصفوفة القوائم، فكل عنصر- في المصفوفة لديه فهرس يقوم بتحديد موقعه داخل المصفوفة، ويمكننا الوصول الى عناصر كل مصفوفة عن طريق كتابة اسم المصفوفة والرقم التسلسلي للعنصر بين قوسين مربعين. المصفوفة يجب أن تحتوي على بيانات من نفس النوع.

مثال لبيانات على شكل مصفوفة Array :

```
from array import *  
my_array = array('i', [85,92,80,81])
```

رمز النوع	نوع البيانات
i	(عدد صحيح)
d	(عدد عشري)
u	(حرف أو رمز)

18 – اكتب الصيغة العامة لتعريف المصفوفة؟



حيث يتم وضع عناصر المصفوفة بين قوسين مربعين [] ويفصل بين كل عنصر وآخر بالفاصلة .

19- كيف يتم تمثيل البيانات في المصفوفة ثنائية الأبعاد 2d ؟ ص 52

في المصفوفة ثنائية الأبعاد 2d Array ، يتم تمثيل البيانات داخل صفوف وأعمدة، وتتم فهرسة كل عنصر في المصفوفة ثنائية الأبعاد بواسطة فهرسين أحدهما يشير إلى الصف والآخر للعمود.

20- ضع علامة ✓ أمام العبارة الصحيحة وعلامة ✗ أمام العبارة الخطأ:

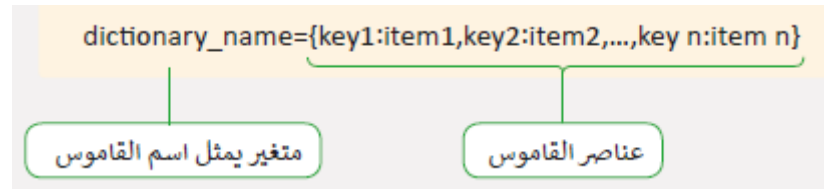
<input type="checkbox"/>	1. يمكن أن تحتوي المصفوفة على بيانات مختلفة النوع.
<input checked="" type="checkbox"/>	2. يستخدم Python القوائم لتمثيل المصفوفات.
<input checked="" type="checkbox"/>	3. عند إنشاء مصفوفة يجب علينا تحديد نوع البيانات.
<input type="checkbox"/>	4. تتم فهرسة كل عنصر في المصفوفة ثنائية الأبعاد برقم فهرس واحد.

21- ما المقصود بالقاموس ؟ وما هي الصيغة العامة لتعريف القاموس ؟

القاموس Dictionary: هو هيكل بيانات يقوم بتخزين البيانات على شكل أزواج، وكل زوج عبارة عن قيمتين

هما المفتاح والقيمة.

الصيغة العامة لتعريف القاموس:



- ❖ تستخدم الأقواس المعقوفة { } عند تعريف القاموس، ويتم الفصل بين العنصر والمفتاح باستخدام علامة :
- ❖ لا يمكن وجود عنصرين في القاموس لهما نفس المفتاح، كل مفتاح يسمح لك للوصول لقيمة واحدة من القيم الموجودة بالقاموس.

22 ما وجه الاختلاف بين القائمة List والقاموس Dictionary ؟ الكتاب ص 58

- ← القائمة هي سلسلة من العناصر المتتابعة، في حين أن القاموس يضم أزواجاً من العناصر غير المرتبة.
- ← الفرق الرئيس يتلخص في كيفية الوصول إلى العناصر، فعناصر القائمة يتم وضعها في قائمة يتم الوصول إليها عن طريق رقم الموقع، بينما يتم الوصول إلى عناصر القاموس من خلال المفاتيح.

23- عدد الدوال الجاهزة المستخدمة مع القاموس ووظيفة كل منها.؟ الكتاب ص 60

الدوال	
الوصف	الدالة
ترجع القيمة المرتبطة بالمفتاح x، وإذا لم يتم العثور على المفتاح في القاموس فإنها ترجع None.	dictName.get(x)
تضيف زوج/أزواج جديدة من العناصر إلى القاموس إذا كانت المفاتيح غير موجودة مسبقاً فيه. أو تقوم بتحديث محتوى القيم المرتبطة بالمفاتيح الموجودة.	dictName.update(x)
تُرجع جميع القيم الموجودة في القاموس.	dictName.values()
تُرجع جميع المفاتيح الموجودة في القاموس.	dictName.keys()
تُحذف جميع العناصر داخل القاموس.	dictName.clear()

24- عدد الاستخدامات المختلفة لهياكل البيانات. ؟ الكتاب ص 63

استخدام هياكل البيانات	
الاستخدام	هياكل البيانات
عندما نحتاج إلى مجموعة يتم تعديلها بشكل متكرر.	القائمة List
عندما نريد تخزين بيانات من نفس النوع.	المصفوفة Array
عندما نحتاج إلى تخزين بيانات لا نريد تغييرها.	الصفوف Tuple
عندما نحتاج إلى وجود ارتباط منطقي ما بين (المفتاح:القيمة).	القاموس Dictionary
عندما نحتاج إلى البحث عن البيانات بناءً على مفتاح مخصص.	

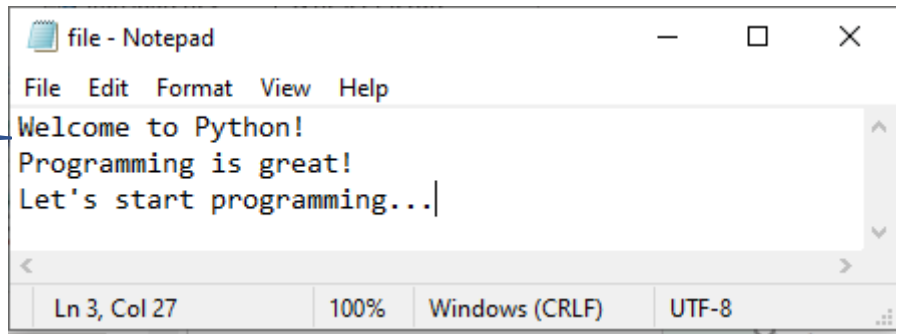
الوحدة الثانية: تقنيات البرمجة ملفات البيانات



• إن أبسط نوع من أنواع حاويات وحفظ البيانات هو الملف النصي Text File.

-1

ملف نصي



2- ما المقصود بالملفات النصية (تعريف الملفات النصية):

الملفات النصية Text Files:

الملف النصي عبارة عن سلسلة من النصوص (تشمل الحروف والأرقام والرموز) ، يمكن إجراء عمليات مختلفة على الملفات النصية كالحذف والإضافة والتعديل ، ويتم ذلك من خلال أوامر برمجية محددة .

3- ما هي العمليات الأساسية لمعالجة ملف البيانات؟ اشرح وظيفة كل منها؟

فتح الملف Open

لفتح ملف، يتعين علينا تحديد موقع الملف في وحدة التخزين، ثم نختار ما إذا كنا نريد أن نقرأ من الملف أو نقوم بالكتابة فيه .

القراءة من الملف Read

قراءة البيانات الموجودة في الملف إلى متغيرات وهياكل بيانات داخل برنامجنا الموجود في الذاكرة لمزيد من عمليات المعالجة.

الكتابة إلى الملف Write

نقل قيم المتغيرات وهياكل بيانات المستخدمة في البرنامج إلى الملف المحفوظ في وحدة التخزين .

إغلاق الملف Close

عندما نغلق ملفاً ، فإن نظام التشغيل يتأكد من انتهاء جميع عمليات القراءة والكتابة .

توضيح:

دالة فتح الملف Open ودالة إغلاق الملف Close.



التعامل مع الملفات في Python

العملية	حرف العملية
قراءة البيانات من ملف موجود مسبقاً.	r
الكتابة إلى ملف.	w
إلحاق البيانات في نهاية ملف موجود.	a

دالة إغلاق الملف Close

عند الانتهاء من عمليات الكتابة والقراءة من الملف يجب علينا إغلاقه وذلك باستخدام Close وتؤمن هذه الدالة حفظ جميع التغيرات إذا قمنا بإجرائها على الملف.

الصيغة العامة لدالة إغلاق الملف : `object.close()`

دالة () read
قراءة جميع محتويات الملف.

دالة () readline
قراءة سطراً واحداً من الملف.

كتابة مقطع برمجي لقراءة محتويات ملف نصي باستخدام دالة Read

الملف النصي

```
file - Notepad
File Edit Format View Help
Welcome to Python!
Programming is great!
Let's start programming...|
```

```
example-1-P81-readFile.py - C:/Users/r.mohamed0401/Desktop/2021-2ف-علوم الحاسب-11/تحضير/
File Edit Format Run Options Window Help
f=open("file.txt" , "r")
print(f.read())
f.close()
```

جملة read
تقرأ محتويات
الملف بالكامل

```
RESTART: C:/Users/r.mohamed0401/Desкто
2021-2ف-علوم الحاسب-11/ير/week7/example
Welcome to Python!
Programming is great!|
Let's start programming...
>>>
```

نتيجة التنفيذ

كتابة مقطع برمجي لقراءة سطراً واحداً من الملف باستخدام دالة Readline

```
f=open("file.txt","r")  
print(f.readline())  
print(f.readline())  
f.close()
```

دالة readline() تقرأ سطراً
واحداً من الملف

نتيجة التنفيذ:

```
Welcome to Python!  
  
Programming is great!  
  
>>>
```

4- ما المقصود بالإلحاق إلى ملف نصي؟ :

هو إضافة نص جديد في نهاية الملف النصي وذلك بفتح ملف نصي موجود مسبقاً بوضعية الإلحاق a .

```
f=open("file.txt", "a")  
f.write("\nThis is the new text.")
```

5- اكتب الصيغة العامة للإلحاق؟

الصيغة العامة للإلحاق: <object>=open("<file_path>","a")

6- ماذا يحدث إذا تم فتح الملف النصي بوضعية "W" مع استخدام دالة write ؟
تقوم دالة write بالكتابة على الملف الحالي بحيث يتم حذف المحتوى السابق

```
f=open("file.txt", "w")  
f.write("The previous text has been deleted...")
```



```
f=open("file.txt", "w")  
f.write("This is a new file")
```

7- ما الفرق بين دالة readline() ودالة readlines() ؟
◀ تقرأ دالة readline() في كل مرة سطرا واحدا من الملف.
◀ أما دالة readlines() فتقوم بقراءة جميع أسطر الملف وترجع قائمة تحتوي على جميع أسطر الملف.

```
f=open("cities.txt", "r")  
print(f.readlines())
```

نتيجة التنفيذ

```
['Doha\n', 'Paris\n', 'London\n', 'Rome\n', 'Berlin']  
>>>
```

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

-8

✗	1. عندما نقرأ من ملف txt. فإننا نضيف أسطرًا جديدة.
✗	2. تقوم عملية الإلحاق إلى ملف بإضافة المحتوى إلى بداية الملف.
✓	3. عملية الكتابة write تقوم باستبدال النص الموجود داخل الملف.
✗	4. يمكننا فتح الملف النصي فقط إذا ما كان بنفس المجلد الموجود به ملف Python البرمجي.
✓	5. لإلحاق نص جديد في الملف فإننا نستخدم معامل "a".

9- ما المقصود بالتكرار المتداخل؟

- ❖ تسمى عملية وضع تكرار داخل تكرار آخر بعملية التداخل Nesting
- ❖ كما يمكن أي نوع تكرار داخل نوع آخر. وأكثر التكرارات المتداخلة شيوعاً هي تكرارات For

القواعد المطبقة على التكرارات المتداخلة:

10

← يجب أن تكون كامل الجملة التكرارية الداخلية داخل الجملة التكرارية الخارجية.

← ينبغي استخدام اسم مختلف للعداد في كل من التكرار الداخلي والخارجي.

← انتبه إلى أن التكرار الداخلي يكتمل أولاً.

← في كل دورة من التكرار الخارجي، يتم تنفيذ جميع دورات التكرار الداخلي.

11- اذكر مثالا على التكرارات المتداخلة في حياتنا اليومية ؟

توجد العديد من الأمثلة على التكرارات المتداخلة في حياتنا اليومية، فمثلاً تعتبر الساعة الرقمية إحدى الأمثلة البسيطة عليها.

في الساعة الرقمية نحتاج 3 تكرارات

- ← الأول خاص بتتبع الساعة.
- ← والثاني يختص بتتبع الدقائق.
- ← والثالث يختص بتتبع الثواني.



12- ما المقصود بالقوائم المتداخلة؟

يمكن أن تحتوي القائمة على أنواع مختلفة من العناصر وقد تحتوي على قائمة، ويطلق عليها حينها مسمى "القوائم المتداخلة"

مثال: list1=[3,["a","b","c"],7.5,-2,"Doha"] العنصر الثاني ["a","b","c"] في القائمة list1 عبارة قائمة.

13- ما المقصود بالمصفوفة ثنائية الأبعاد [2D] Array ؟

- ↪ المصفوفات الثنائية يتم تمثيلها في بايثون باستخدام القوائم المتداخلة
- ↪ كما يتم تمثيل البيانات في المصفوفة ثنائية الأبعاد في صفوف وأعمدة، وتتم فهرسة كل عنصر في المصفوفة ثنائية الأبعاد بفهرسين أحدهما للصف والآخر للعمود.

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

14-

1.	يقوم Python بإنشاء مصفوفة ثنائية الأبعاد 2D array باستخدام القوائم.	✓
2.	يتم تمثيل البيانات في المصفوفة ثنائية الأبعاد 2D array فقط من خلال الصفوف.	✗
3.	تتم فهرسة كل عنصر في المصفوفة ثنائية الأبعاد 2D array بواسطة فهرسين.	✓
4.	إن الفهرسين في عنصر المصفوفة ثنائية الأبعاد 2D array، أحدهما للعمود والآخر لحجم المصفوفة.	✗
5.	لطباعة جميع عناصر المصفوفة، نحتاج إلى استخدام التكرارات المتداخلة.	✓

15- استخدم المصفوفة ثنائية الأبعاد لملء الفراغات.

رقم الصف

رقم العمود

```
nums[0][2]=2
nums[1][1]=5
nums[2][3]=11
nums[3][2]=14
nums[0][3]=3
nums[1][0]=4
nums[1][3]=7
nums[2][1]=9
```

nums	0	1	2	3
0	0	1	2	3
1	4	5	5	7
2	8	9	10	11
3	12	13	14	15

16- ما الذي ستتم طباعته بواسطة المقطع البرمجي التالي؟

☐ 3
 ☐ 4
 ☐ a
 ☒ b

```
list=[3,4,["a","b"]]
print(list[2][1])
```

1.

☐ 1
 ☒ a
 ☐ 0
 ☐ b

```
list=[1,"a",[0,"b"]]
print(list[1])
```

2.

☐ a
 ☒ d
 ☐ 3
 ☐ c

```
list=[3,["c","d"],["a","b"]]
print(list[1][1])
```

3.

☐ 7
 ☒ a
 ☐ c
 ☐ b

```
list=[[7,"a"],4,["c","b"]]
print(list[0][1])
```

4.

17- ما المقصود بالدوال في لغة بايثون؟

الدالة هي عبارة عن مقطع برمجي قابل لإعادة الاستخدام بحيث يستخدم لتنفيذ إجراء معين في البرمجة، ويتم استدعائها بواسطة البرنامج الرئيس أو بواسطة دوال أخرى في البرنامج.

18- اذكر تعريف دالة الاستدعاء الذاتي؟

الاستدعاء الذاتي هي العملية التي تستدعي بها الدالة نفسها.

19- عدد مكونات دالة الاستدعاء الذاتي؟

تتكون دالة الاستدعاء الذاتي من حالتين:

← الحالة الأساسية (Base case).

هي الحالة التي تتوقف فيها الدالة عن استدعاء ذاتها، ويتم التحقق من الوصول للحالة الأساسية من خلال جملة شرطية، وبدون الحالة الأساسية ستكون عملية الاستدعاء الذاتي لا نهائية.

← حالة الاستدعاء الذاتي (Recursive case).

هي الحالة التي تستدعي فيها الدالة ذاتها عند عدم تحقق شرط التوقف، وتبقى الدالة في حالة الاستدعاء الذاتي لحين وصولها إلى الحالة الأساسية.

20- أذكر إيجابيات وتحديات دالة الاستدعاء الذاتي؟

الإيجابيات والتحديات لاستخدام دوال الاستدعاء	
التحديات	الإيجابيات
في بعض الأحيان يكون من الصعب تتبع المنطق المتبع في دوال الاستدعاء الذاتي	دوال الاستدعاء الذاتي تختزل المقطع البرمجي إلى عدد أصغر من التعليمات
الاستدعاء الذاتي يتطلب المزيد من الذاكرة والوقت	يمكن تقسيم المهمة الصعبة إلى مجموعة من المشاكل الفرعية باستخدام الاستدعاء الذاتي
ليس من السهل تحديد الحالات التي يمكن فيها استخدام دوال الاستدعاء الذاتي	في بعض الأحيان يكون من السهل استخدام الاستدعاء الذاتي بدلاً من التكرارات المتداخلة

21- ما أوجه الاختلاف بين التكرار والاستدعاء الذاتي؟

الاستدعاء الذاتي والتكرار	
التكرار	الاستدعاء الذاتي
تنفيذه أسرع	البطء في التنفيذ مقارنة بالتكرار
يحتاج إلى ذاكرة أقل	يحتاج لمزيد من الذاكرة
حجم المقاطع البرمجية أكبر	حجم المقاطع البرمجية أقل
ينتهي باستكمال عدد التكرارات المحددة أو تحقق شرط	ينتهي عند الوصول إلى حالة الأساس

22- متى نستخدم الاستدعاء الذاتي؟

- ← في كثير من الحالات تعتبر كطريقة أكثر سهولة عند التعامل مع مشكلة ما.
- ← بعض تراكيب البيانات من السهل استكشافها باستخدام الاستدعاء الذاتي.
- ← بعض خوارزميات الفرز مثل الفرز السريع **quick sort** تستخدم الاستدعاء الذاتي.

23- ما المقصود بدالة الاستدعاء الذاتي اللانهائي؟

عند تنفيذ استدعاء دالة الاستدعاء الذاتي ، فيجب أن توفر طريقة لإيقاف التكرار بإيجاد شرط معين لتفادي حدوث التكرار غير المحدود (الانهائي).