

أسلوب الكتابة بلغة بايثون

CODE LAY-OUT

أهداف الدرس



أهداف الدرس

- 1- يتعرف أسلوب كتابة الأوامر للغة Python.
- 2- يتقن بناء الأوامر البرمجية للغة Python.

قواعد كتابة الكود في بايثون هي التالية:

1- مبادئ كتابة الكود

- لإضافة تعليق ابدء السطر بالرمز #: التعليق لا ينفّذ البرنامج، بل يقوم المبرمج بإضافته ليتذكّر شيئاً خاصاً بالكود الذي أدخله وليكون الكود واضحاً بالنسبة له.

هذا السطر يحتوي على تعليق #

- عند إدخال تعليق على عدّة أسطر استخدم الرمز " " في بداية التعليق ونهايته.

هذا السطر الأول من التعليق " "

هذا السطر الثاني من التعليق

" " هذا السطر الثالث من التعليق

- لا تستخدم المفتاح TAB لترك المسافات الفارغة لأن المسافة التي يعطيها هذا الزر غير مسموح استخدامها في لغة بايثون إذا كانت تتعدّى الأربع فراغات.
- أترك سطر فارغ على الأقل بين كل دالتين.
- أترك مسافة فارغة حول جمل التحكم وجمل الشرط.
- أترك 4 مسافات فارغة Spaces عند كتابة الكود بشكل متداخل، ويقصد بذلك كتابة مجموعة من الأوامر تنتمي إلى جملة برمجية أخرى مثل الجملة الشرطية أو التكرارية، مع العلم أنّ معظم المحرّرات الآن تقوم بهذا الأمر بصورة تلقائية.

if age>20:

4 فراغات print("")

- عدد الأحرف القصوى التي يمكن كتابتها السطر الواحد هو 79 حرفاً.

ملاحظة: سوف تتمكن عزيزي الطالب من استيعاب هذه القواعد عند البدء الفعلي في البرمجة بلغة Python.

2- حساسية حالة الأحرف Case Sensitivity

أي أن لغة البايثون تميّز حالة الأحرف، وخاصّة بالنسبة لأسماء المتغيّرات أو الدوال أو غيرها من المعرّفات، فمثلاً تعيين المتغيّر **A** والمتغيّر **a** لا يمثل نفس المتغيّر فهما مختلفان.

3- أسماء المتغيّرات

إذا كان المتغيّر مكوّنًا من أكثر من كلمة استخدم الرمز (_) للفصل بين الكلمات.

first_name مثال 1

4- كتابة أكثر من تعيين في نفس السطر

لكتابة أكثر من تعيين في نفس السطر، أفصل بينها بالنقطة الفاصلة.

a=1 ; b=2 ; c=3 مثال 2

5- كتابة أمر واحد على أكثر من سطر.

لكتابة أمر واحد على عدّة سطور، أضيف الرمز \ (backslash) نهاية السطر

مثال 3

result = numb1 + \

numb2 + \

numb3

ملاحظة: الأوامر التي تحتوي على الرموز [] أو () أو { } لا تحتاج \ في نهاية كل سطر.

مثال 4

```
cars = ['TOYOTA',
        'NISSAN',
        'MAZDA']
```

6- الكلمات المحجوزة في لغة البرمجة Python: هي كلمات مفتاحية خاصة بلغة البايثون والتي لا يمكن توظيفها كمتغيرات أو معرفات في الحل البرمجي.

and	def	exec	global	lambda	pass	assert
del	false	if	none	print	break	elif
finally	import	nonlocal	raise	class	else	for
in	not	return	continue	except	from	is
or	true	try	while	with	yield	

مثال 5

لو حاولنا استخدام الكلمة المفتاحية if كمتغير وأسندنا له القيمة 5 مثلاً، ستكون النتيجة رسالة خطأ الآتية:

```
if=5
SyntaxError: invalid syntax
```

7- إدراج بعض العلامات الخاصة عند العرض

أ- إضافة المسافة TAB: استخدام الرمز \t ←

```
print("Welcome to \t bahrain")
Welcome to         bahrain
```

ب- إضافة علامة تنسيق مفردة: بما أنَّ علامة التنسيق المفردة ضمن الأمر `print` تقوم بحصر النص الذي أريد إدراجه، فماذا لو أردت أن أظهر هذه العلامة بذاتها مع النص الذي سيظهر على الشاشة؟ للقيام بذلك أدرج الرمز `'`

```
print("\' WELCOME TO BAHRAIN \'")
' WELCOME TO BAHRAIN '
```

ت- إضافة علامة تنسيق مزدوجة: بما أنَّ علامة التنسيق المزدوجة ضمن الأمر `print` تقوم بحصر النص الذي أريد إدراجه، فماذا لو أردت أن أظهر هذه العلامة بذاتها مع النص الذي سيظهر على الشاشة؟ للقيام بذلك أدرج الرمز `"`

```
print("\\" WELCOME TO BAHRAIN \")
" WELCOME TO BAHRAIN "
```

ث- إدراج سطر جديد: للقيام بذلك أدرج الرمز `\n`

```
>>> print("WELCOME TO BAHRAIN \n")
WELCOME TO BAHRAIN
>>>
```

السطر الجديد