CS1011 البرمجة الحاسوبية بلغة Python

المحاضرة 1: مقدمة في علوم الحاسوب والبرمجة بلغة Python

كلمات هذه الصفحة

Classes — كلاسات

Operator — معامل

عملية تطوير البرمجيات

- غالبا ما يتم تقسيم عملية إنشاء البرنامج إلى مراحل بناء على المعلومات التي يتم إنتاجها في كل مرحلة.

كلمات هذه الصفحة

broken down — مقسمة

stages — مراحل

produced — يتم إنتاجها

phase — مرحلة

عملية تطوير البرمجيات

- 1) تحليل المشكلة
  - Figure out exactly the problem to be solved. Try to understand it as much as possible.

كلمات هذه الصفحة

Analyze — تحليل

Figure out — حدد

exactly — بدقة

solved — يتم حلها

عملية تطوير البرمجيات

- **2) تحديد المواصفات**
- صف بدقة ما سيقوم به برنامجك.
  - لا تقلق بشأن كيفية عمل البرنامج، بل ركز على ما سيقوم به.

  - يتضمن ذلك وصف المدخلات والمخرجات، وكيفية ارتباطها ببعضها البعض.

كلمات هذه الصفحة

Determine — تحديد

Specifications — المواصفات

inputs — المدخلات

outputs — المخرجات

relate — ارتباط

عملية تطوير البرمجيات

3 ■) إنشاء تصميم

■ صغ الهيكل العام للبرنامج. ■ هنا يتم تحديد كيفية عمل البرنامج. ■ طور خوارزميتك الخاصة التي تلبي المواصفات.

كلمات هذه الصفحة

Design — تصميم

Formulate — صغ

overall structure — الهيكل العام

worked out — يتم تحديدها

algorithm — خوارزمية

meets — تلبي

عملية تطوير البرمجيات

- 4) تنفيذ التصميم
  - Translate the design into a computer language.
  - In this course we will use Python.

كلمات هذه الصفحة

Implement — تنفيذ

Translate — ترجم

course — دورة دراسية

عملية تطوير البرمجيات

- **5)** اختبار/تصحيح البرنامج
  - جرب برنامجك لترى ما إذا كان يعمل.
- If there are any errors (bugs), they need to be located and fixed. This process is called debugging.
- Your goal is to find errors, so try everything that might “break” your program!

كلمات هذه الصفحة

Test — اختبار

Debug — تصحيح الأخطاء

located — محددة الموقع

fixed — إصلاحها

debugging — عملية تصحيح الأخطاء

goal — هدف

break — تعطيل

عملية تطوير البرمجيات

- 6) صيانة البرنامج
  - Continue developing the program in response to the needs of your users.
- In the real world, most programs are never completely finished – they evolve over time.

كلمات هذه الصفحة

Maintain — صيانة

in response to — استجابة لـ

real world — العالم الحقيقي

evolve — تتطور

برنامج مثال: محول درجات الحرارة

1. التحليل – درجة الحرارة معطاة بالدرجة المئوية (Celsius)، ويريد المستخدم التعبير عنها بالدرجة فهرنهايت (Fahrenheit).
2. المواصفات
   - المدخلات – درجة الحرارة بالدرجة المئوية
   - المخرجات – درجة الحرارة بالدرجة الفهرنهايتية
   - المخرجات = 9/5 × (المدخلات) + 32
   - فهرنهايت = 9/5 × سيلزيوس + 32

كلمات هذه الصفحة

Celsius — الدرجة المئوية

expressed — التعبير عنها

Fahrenheit — الدرجة الفهرنهايتية

برنامج مثال: محول درجات الحرارة

- 3) التصميم
  - الإدخال، المعالجة، الإخراج (IPO)
  - اطلب من المستخدم إدخال البيانات (درجة الحرارة المئوية)
- Process it to convert it to Fahrenheit using F = 9/5*(C) + 32
- أخرج النتيجة عن طريق عرضها على الشاشة

كلمات هذه الصفحة

Prompt — اطلب

displaying — عرض

screen — الشاشة

برنامج مثال: محول درجات الحرارة

- قبل أن نبدأ في البرمجة، دعنا نكتب مسودة أولية للبرنامج باستخدام الشيفرة الوصفية **(pseudocode)**
- الشيفرة الوهمية هي لغة إنجليزية دقيقة تصف ما يفعله البرنامج، خطوة بخطوة.
- باستخدام الكود الزائف، يمكننا التركيز على الخوارزمية بدلا من لغة البرمجة.

كلمات هذه الصفحة

coding — البرمجة

rough draft — مسودة أولية

pseudocode — شيفرة وصفية

precise — دقيقة

concentrate — التركيز

برنامج مثال: محول درجات الحرارة

- الكود الزائف:
  - أدخل درجة الحرارة بالدرجات المئوية (سمها celsius)
  - احسب fahrenheit كـ (9/5)*celsius+32
  - أخرج قيمة fahrenheit
- 4) الآن نحتاج إلى تحويل هذا إلى Python!

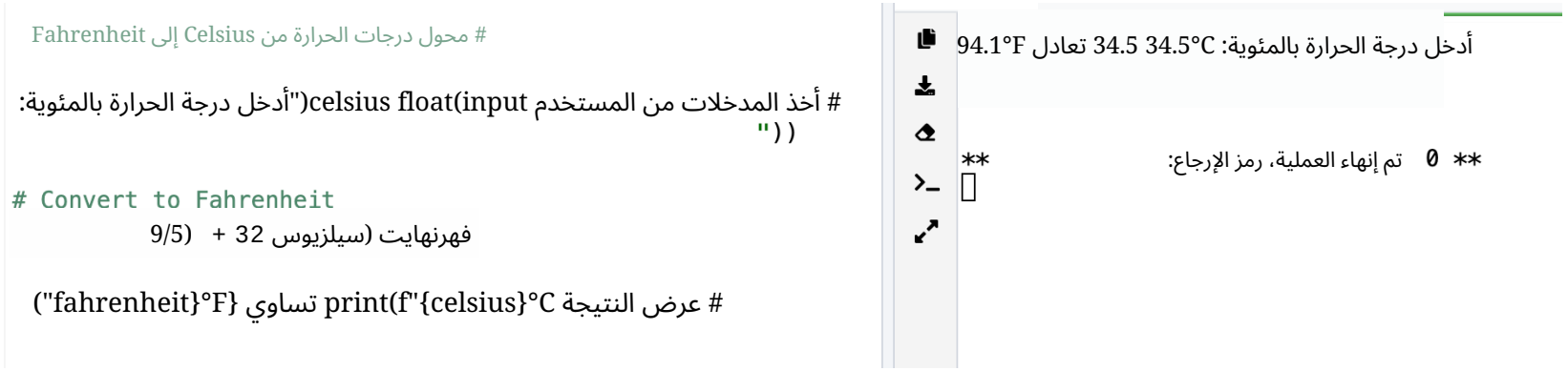
برنامج مثال: محول درجات الحرارة

```
# Celsius to Fahrenheit Converter

# Take input from the user
celsius = float(input("Enter temperature in Celsius: "))

# Convert to Fahrenheit
fahrenheit = (celsius * 9/5) + 32

# Display the result
print(f"{celsius}°C is equal to {fahrenheit}°F")
```

برنامج مثال: محول درجات الحرارة

- 5) بمجرد كتابة البرنامج، يجب علينا اختباره!

```
Enter temperature in Celsius: 34.5
34.5°C is equal to 94.1°F
```



ما هي Python؟

- تعد Python لغة برمجة متعددة الأغراض
- تم إصدارها في عام 1991
- وهي تدعم نماذج برمجة متعددة مثل البرمجة كائنية التوجه (object-oriented)

كلمات هذه الصفحة

general-purpose — متعددة الأغراض

paradigms — نماذج

لماذا Python؟

يعمل على مختلف أنظمة التشغيل

يتمتع بـ بنية لغوية

يمكن لشيفرته البرمجية أن ينفذ بمجرد كتابته (مفسر)

كلمات هذه الصفحة

operating systems — أنظمة التشغيل

syntax — بنية لغوية

executed — ينفذ

interpreter — مفسر

مكونات Python

1) الدوال

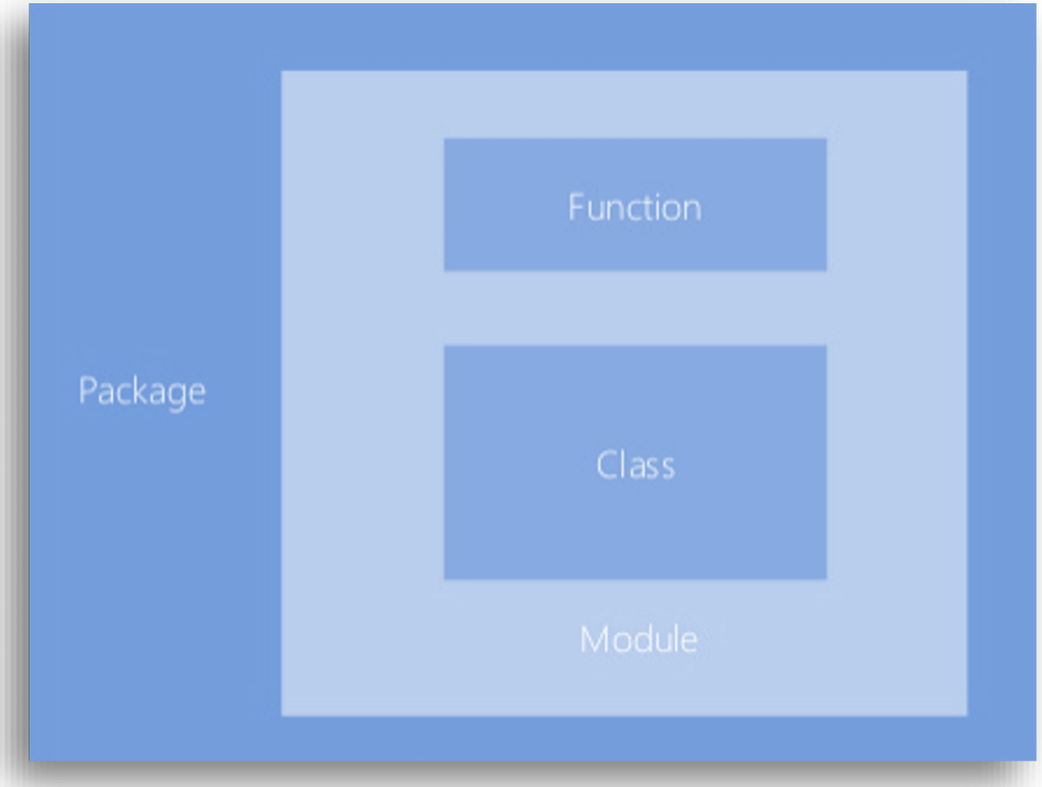
2) الكلاسات

3) الوحدات البرمجية

مجموعة من الدوال والكلاسات

4) الحزم

مجموعة من الوحدات البرمجية



كلمات هذه الصفحة

Components — مكونات

Functions — دوال

Modules — وحدات برمجية

collection — مجموعة

Packages — حزم

العمليات الحسابية

المعامل	المعنى	مثال	Y
+	الجمع	x= 10 y = x + 3	13
-	الطرح	x= 6 y = x – 4	2
*	الضرب	x= 2 y = x * 5	10
/	القسمة	x= 10 y = x / 7	1.4
%	باقي القسمة (remainder)	x= 10 y = x % 7	3
**	الأس	x= 3 y = x ** 3	27

كلمات هذه الصفحة

Addition — الجمع

Multiplication — الضرب

Modulus — باقي القسمة

Exponent — الأس

Subtraction — الطرح

Division — القسمة

reminder — باقي

المتغيرات

- المتغير (variable) هو موقع مسمى في الذاكرة يمكنه تخزين قيمة ما.
- يتم إنشاء المتغير عند تعيين قيمة له لأول مرة.
- لا تحتاج المتغيرات إلى تعريف مسبق لنوعها، بل يمكنها حتى تغيير نوعها بعد تعيينها.
- يمكن استخدام المتغير الذي تم تعيين قيمة له في التعبيرات البرمجية.
- الصيغة:

متغير_الاسم = القيمة

كلمات هذه الصفحة

Variables — متغيرات

named location — موقع مسمى

store — تخزين

assign — تعيين

declared — تعريف مسبق · تحديد نوع المتغير قبل استخدامه

expressions — تعبيرات برمجية

المتغيرات

المتغيرات:قواعد الأسماء

يجب أن يبدأ اسم المتغير بحرف أو _

يمكن أن يحتوي اسم المتغير فقط على أرقام وحروف ورمز الشرطة السفلية _

أسماء المتغيرات حساسة لحالة الأحرف (age ≠Age ≠ AGE)

لا يمكن أن يكون اسم المتغير كلمة محجوزة (انظر أدناه)

كلمات هذه الصفحة

Names Rules — قواعد الأسماء

underscore symbol — رمز الشرطة السفلية

case-sensitive — حساسة لحالة الأحرف

الكلمات المفتاحية

- بعض المعرفات هي جزء من لغة Python نفسها. تعرف هذه المعرفات باسم الكلمات المحجوزة (أو الكلمات المفتاحية). وهذا يعني أنها غير متاحة لك لاستخدامها كأسماء للمتغيرات أو غيرها في برنامجك.

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	خطأ	None		

كلمات هذه الصفحة

Keywords — كلمات مفتاحية

identifiers — معرفات

reserved words — كلمات محجوزة

المتغيرات: قواعد الأسماء

اسم المتغير	الحالة
num_1st	×
مخزن مؤقت _20	√
_المدينة	√
معرف الطالب:	×
المجموع والمتوسط	×
import	×

الثوابت

- الثابت هو متغير لا يمكن تغيير قيمته
- للتمييز بين المتغير والثابت، اعتاد المبرمجون كتابة الثوابت بـحروف كبيرة بالكامل مع استخدام الشرطة السفلية للفصل بين الكلمات

- تتبع الثوابت نفس قواعد تسمية المتغيرات
- الصيغة: CONSTANT_
NAME = القيمة

كلمات هذه الصفحة

Constants — ثوابت

distinguish — للتمييز

capital letters — حروف كبيرة

الثوابت

- في الجدول التالي، أي أسماء الثوابت تخالف/تتوافق مع قواعد التسمية وأسلوب المبرمجين

اسم الثابت	الحالة
num_1st	X
BUFFER-20	X
_المدينة	√
طالب NAME_	√
المجموع والمتوسط	X
RATE	√

كلمات هذه الصفحة

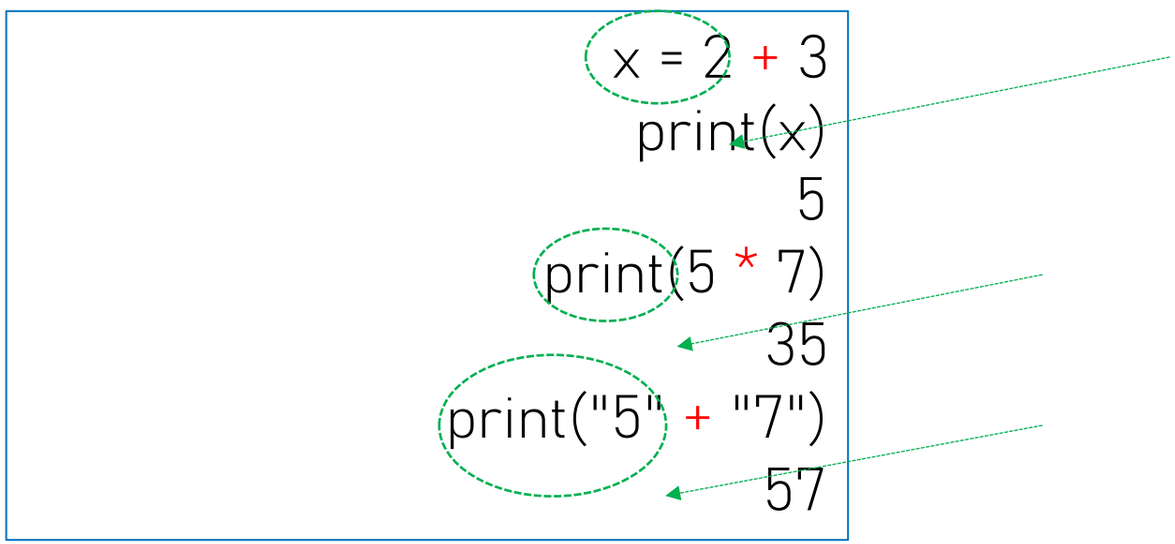
violate — تخالف

conform to — تتوافق مع

تعبيرات

يمكنك إنتاج قيم بيانات جديدة (رقمية أو نصية) في برنامجك باستخدام التعبيرات

```
x = 2 + 3
print(x)
5
print(5 * 7)
35
print("5" + "7")
57
```



§ هذا تعبير يستخدم *(operator)* الجمع

§ هذا تعبير آخر يستخدم الضرب

§ هذا تعبير إضافي آخر يستخدم الجمع كعامل لكن لـ ربط (أو لصق) سلاسل نصية معا

كلمات هذه الصفحة

data — بيانات

concatenate — ربط

glue — لصق

Strings — سلاسل نصية

أنواع البيانات: الأعداد الصحيحة، والأعداد العشرية، والسلاسل النصية

نوع البيانات	مثال
الأعداد الصحيحة	12
الأعداد العشرية	2.75
السلاسل النصية	"مرحبا بالأصدقاء"
القيم المنطقية	واحدة من قيمتين: True أو False

كلمات هذه الصفحة

Data types — أنواع البيانات

integers — أعداد صحيحة

floats — أعداد عشرية

Booleans — قيم منطقية

الإدخال/الإخراج (print, input)

تطلب ميثود (method) الـ **input()** من المستخدم إدخال بيانات وتعيدها.

- من المهم إسناد ميثود الـ **input()** إلى متغير لتخزين بيانات الإدخال.

تقوم دالة الـ **print()** بطباعة الرسالة المحددة إلى جهاز إخراج قياسي.

- تستخدم الـ **print()** الفاصلة (,) لربط الوسائط المنفصلة الممررة إليها، وتدرج مسافة واحدة بينها حتى لا تنتهي برسالة متداخلة.

الصيغة:

- اسم_المتغير = input(“رسالة المطالبة”)

كلمات هذه الصفحة

method — ميثود · دالة تابعة لكائن

standard output device — جهاز إخراج قياسي

comma — فاصلة

arguments — وسائط

squashed — متداخلة

إدخال) إدخال (

- يطبع كل ما يوجد داخل علامتي الاقتباس
- يكتب المستخدم شيئا ما ثم يضغط على مفتاح الإدخال (enter)

- يربط تلك القيمة بمتغير

```
text = input("Type أي شيء... ")
print(5*text)
```

```
Type أي شيء... 8
88888
```

Type
88888

- `input` يعطيك سلسلة نصية، لذا يجب عليك تحويل النوع (cast) إذا كنت

```
num = int(input("اكتب رقما... "))
print(5*num)
```

```
اكتب أي شيء... 8
40
```

كلمات هذه الصفحة

quotes — علامات الاقتباس

Binds — يربط

cast — تحويل النوع

المخرجات (print)

- تستخدم لإخراج البيانات إلى وحدة التحكم (console)
- الكلمة المفتاحية هي print

مخرجات

```
x = 1
print(x)


str
str print("رقمي المفضل")
                    هو",  x, ".", "x =", x)


print("رقمي المفضل")print  is " + x_str + ". " + "x = " + x_str)
```

```
fav num is 1 . x = 1
الرقم المفضلis 1. x = 1
```

كلمات هذه الصفحة

console — وحدة التحكم

مثال

اطلب من المستخدم إدخال اسمه واطبعه

```
your name: daliah
s daliah
```

دوال التحويل: int(), float()

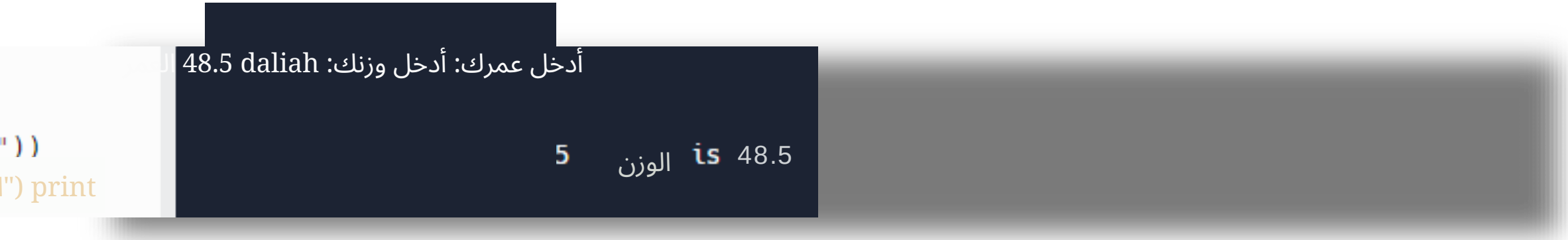
- دالة **int()** : تحول من سلسلة نصية (أو عدد عشري) إلى عدد صحيح
- دالة **float()** : تحول من سلسلة نصية (أو عدد صحيح) إلى عدد عشري
- تعد هذه الدوال مفيدة جدا مع الميثود **input()** . يقوم الميثود input() بإرجاع مدخلات المستخدم كسلسلة نصية (string) حتى لو كانت البيانات أرقاما. لذا، استخدم float() أو int() بعد إدخال البيانات الرقمية.

مثال

- اجمع رقمين واطبع النتيجة.

مثال

- اطلب من المستخدم إدخال اسمه وعمره ووزنه ثم اطبعها



هل لديك أسئلة؟