Introduction to programming in Python

Arthur Gilly

arthur@makerbox.la
52413104

February 20, 2022 at 1:00 PM
Makerbox Lao

What will you learn today?

Our agenda today

- What programming is and its basic concepts
- How to use variables
- Understand variable types (string, numbers, logical)
- Use common mathematical and logic operators
- Use flow control to evaluate conditions
- Use loops

What is Programming?

- Programming is writing computer programs
- The computer only understands machine code
- Impractical for humans, so **programming languages** were created to be more human-friendly
- There are hundreds of programming languages but they all share the same building blocks
- Python is an **interpreted** language
  - You can edit your code like a normal document and run it in python without additional steps
  - Or even run interactively in a console
- Other languages like C++ are **compiled**
  - They need to be translated into machine code by a **compiler**
  - They are faster, but a bit more tricky to use

What is a variable?

- A variable is a bit of memory that contains a value
- It exists while the program is running, can be changed by the program
  - You can write code to change that value depending on events

- We have variables in real life:
  - The amount of money in my wallet
  - My current average grade at university

Declaration/Initialisation

- In order to use a variable, you have to declare/initialize it
- This is done with the assignation operator “=”

```
i = 8
i ← 8
```

- You can change the value of the variable in the same way

Variable type

- The type of a variable is the class of the information it contains
- In many languages type is very important, you cannot easily change type
- In Python, type is “hidden”, you can switch between types easily
- Visualise the type of your variable with `type(i)`
- What is the type of a variable that contains

```
2
“hello”
3.5
False
```

Basic types

Type	Name	Definition
`int`	Integer	Integer numbers, including negatives
`str`	Strings	Character strings, delimited by double quotes " or single quotes'
`float`	Floating-point numbers	There is a limit both on the precision and value of floats, but they are enough for most uses
`bool`	Booleans	can only be either True or False

Type restrictions

- Try adding two integers together with `+`
- Try adding a `string` with an `integer`
- The operations you can perform on variables depend on their types
- You can convert all integers to strings using `str()` and some strings to integers using `int()`

Other assignation operators

- What do the following operations do?
- `a += 1`
- `b -= 3`

Mathematical operators

- You can use Python like a simple calculator
- It understands the following operators
  - `+`
  - `-`
  - `/`
  - `//`
  - `*`
  - `%` (modulo operator)

    `7 % 3 = 1` because `7 = 2 * 3 + 1`
- And more, which we will not cover today :)

Operand 1 type	Operator	Operand 2 type	Result type
`int`	`+, -`	`int`	
`float`	`+, -`	`int`	
`int`	`/`	`int`	

String operators

- Try the following string operators
- `s1 + s2`
- `s1 in s2`
- `s1[i]`, where `i` is an integer
  - Python numbers indices starting at `0`

Comparison operators

- Like in the previous slide, we have operators that produce Boolean values for number types
- `==`, `!=` (between all types of variables)
- `<`, `<=`, `>`, `>=` (numbers only)

Logical operators

- These operators operate on Booleans

and	O1	O2	Result
	True	True	True
	True	False	False
	False	True	False
	False	False	True

or	O1	O2	Result
	True	True	True
	True	False	True
	False	True	True
	False	False	False

not	O1	Result
	True	False
	False	True

Logical operators

- These operators operate on Booleans

and	O1	O2	Result
	True	True	True
	True	False	False
	False	True	False
	False	False	True

or	O1	O2	Result
	True	True	True
	True	False	True
	False	True	True
	False	False	False

not	O1	Result
	True	False
	False	True

- They can be combined with comparison operators to create “conditions”
- `sex == “male” and age > 20 and not age > 40`

Exercise

- How do you write the following conditions:
  - A number is odd/even
  - A string is empty
  - A string contains the letter “`A`”
  - A number is between `0` and `1`
  - A number is either zero, or between 5 and 10
- When are the following conditions True?
  - `i <= 5`
  - `i < 5 and i > 5`
  - `i < 5 or i > 5`

Flow control: if...elif...else

```
if CONDITION :

 # what to do when CONDITION is True

elif ALTERNATIVE :

 # what to do if CONDITION is False but ALTERNATIVE is True

 # (optional)

else :

 # what to do when all the conditions above are False

 # (optional)
```

Flow control: if...elif...else

```
if age < 12 :

 print(“you are a child”)

elif age < 18 :

 print(“you are a teenager”)

elif age < 60:

 print(“you are an adult”)

else :

 print(“you are a senior”)
```

Change the code above to print both “you are an adult” and “you are a senior” when age is equal or above 60

Flow control: if...elif...else

```
if age < 12 :

 print(“you are a child”)

elif age < 18 :

 print(“you are a teenager”)

else :

 print(“you are an adult”)

if age >= 60 :

 print(“you are a senior”)
```

Loops: the while loop

```
while CONDITION:

 # will be repeated as long as CONDITION is True

 # if CONDITION is not True the first time while is read,

 # then this will never be run

 # the code inside can (and should) change CONDITION
```

Loops: the while loop

```
i=0

while i<=10 :

 print(“i is” + str(i))

 i+=1
```

Loops: the while loop

```
i=0

sum=0

while i<=2 :

 j=input(“Please input an integer (“+str(3-i)+” remaining) : ”)

 j=int(j)

 sum+=j

 i+=1

Print(“thank you, the sum is “+str(sum)+”\n”)
```

Write code that takes in integers from the user until the total is above 20

The for loop

```
for var in VALUES:

 # this will be executed once for each item in VALUES

 # at every iteration, var contains the corresponding item
```

The for loop

```
for i in range(0,10):

 print(i)


for a in [“world”, “Laos”, “Vientiane”]:

 print(“hello, “+a)
```

Exercises

- Takes an integer from the user, prints all odd numbers smaller than this number
- Takes 10 integers from the user, counts the number of odd ones
- Prints all integers from 0 to 10 except 3 and 6
- Chooses a random integer:
  - ```
    import random
    Toguess= random.randint(0,10)
    ```
  - And gives the user 3 tries to guess it
- Prints the following pattern using 2 for loops:

```
*
**
***
****
***
**
*
```

Feel free to send in any questions to workshops@makerbox.la
```

Feedback Questionnaire