UNIVERSITÉ DE LORRAINE

Institut des sciences du Digital Management & Cognition
COMPOSANTE DE L'UNIVERSITÉ DE LORRAINE

# Programming

## 1- Introduction, foundations of Python

Those slides will be available on Arche

# Introduction

Gaël Guibon - gael.guibon@univ-lorraine.fr

**Labs** with Nasser-Eddine Monir (PhD Student - Loria - Multispeech)

**Evaluation**

70% exam

30% based on labs: For most labs (not the first one), you will have until the **day before the next lab session** to send me your solution.

Content inspired from Mathieu d'Acquin

# Course Overview

1. Python Basics - *RPG Stammering - Character Creation*
2. Control Flow, Input/Output, Strings - *RPG version 1*
3. Algorithms and Lists - *RPG version 2*
4. Dictionaries, Tuples, Functions - *RPG version 3*
5. Recursion, Exception, and Errors - *RPG version 4*
6. Object Oriented Programming and Advanced Data Structures - *RPG version 5*
7. Libraries and summary - *RPG's Final Form*

# TODAY

What is programming?

What is python and how do we work with Python?

What are the different elements of a Python program?

What are the basic data types in Python and how to manipulate them?

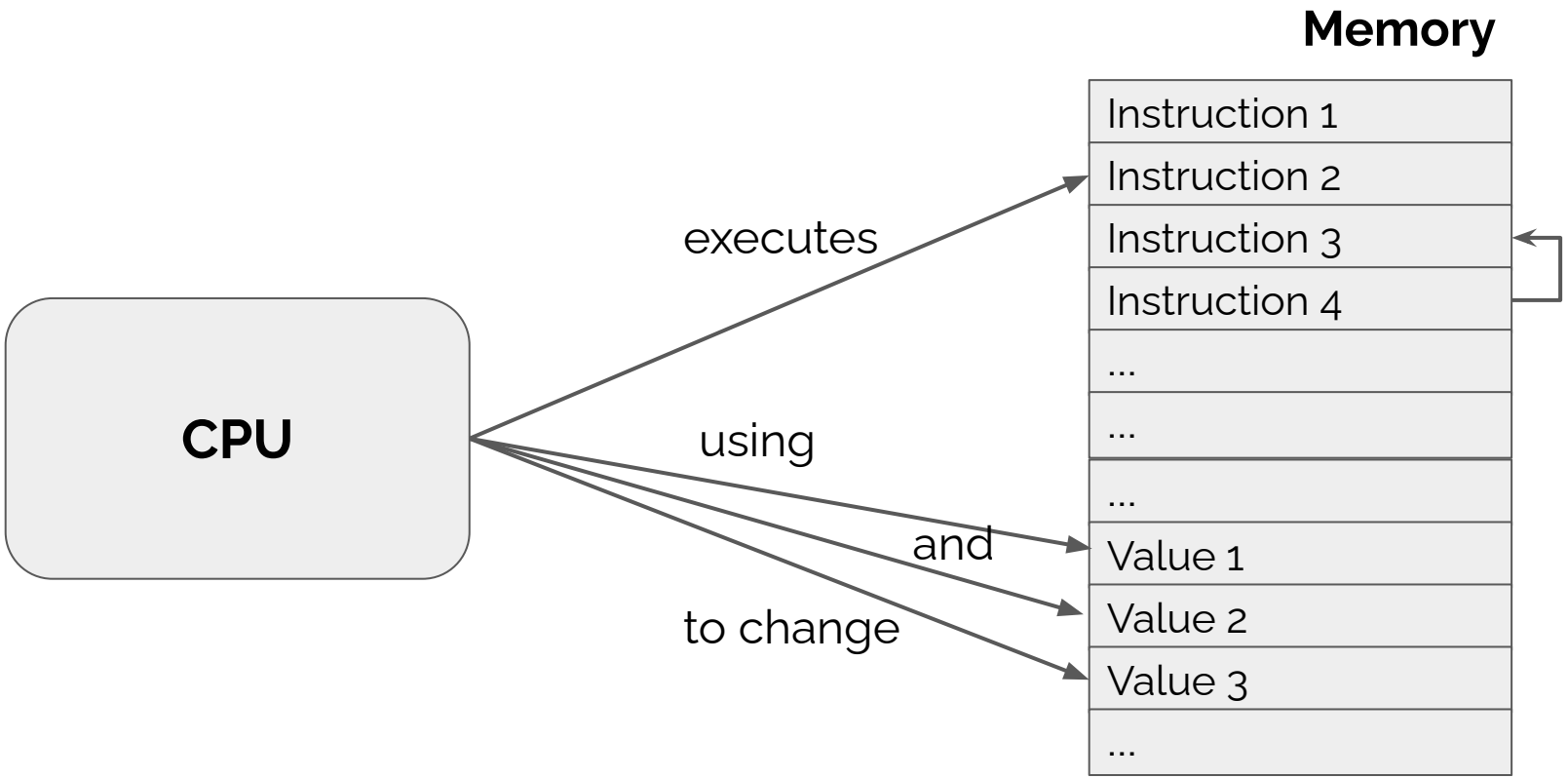
# Programming

*"Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result or to perform a specific task."*

-- Wikipedia

Other words: Coding, developing (computer software)

# A very simplified version of a computer



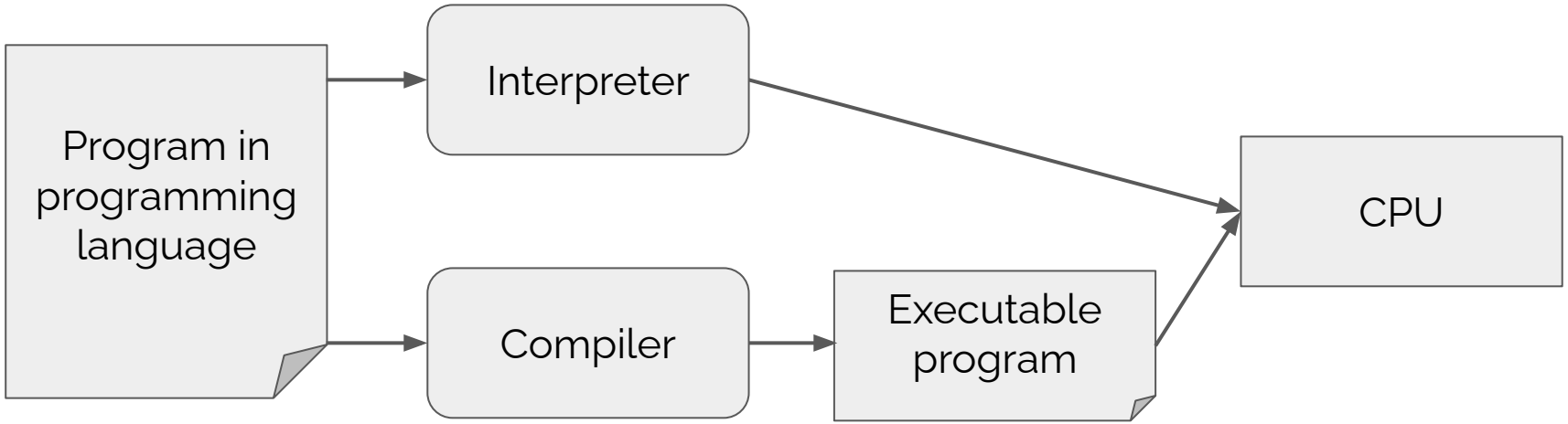
# Programming

Writing a set of instructions (an algorithm) to achieve the desired result.

```
1 Wake up
2 Take a shower
3 Drink coffee
4 Go to class
```

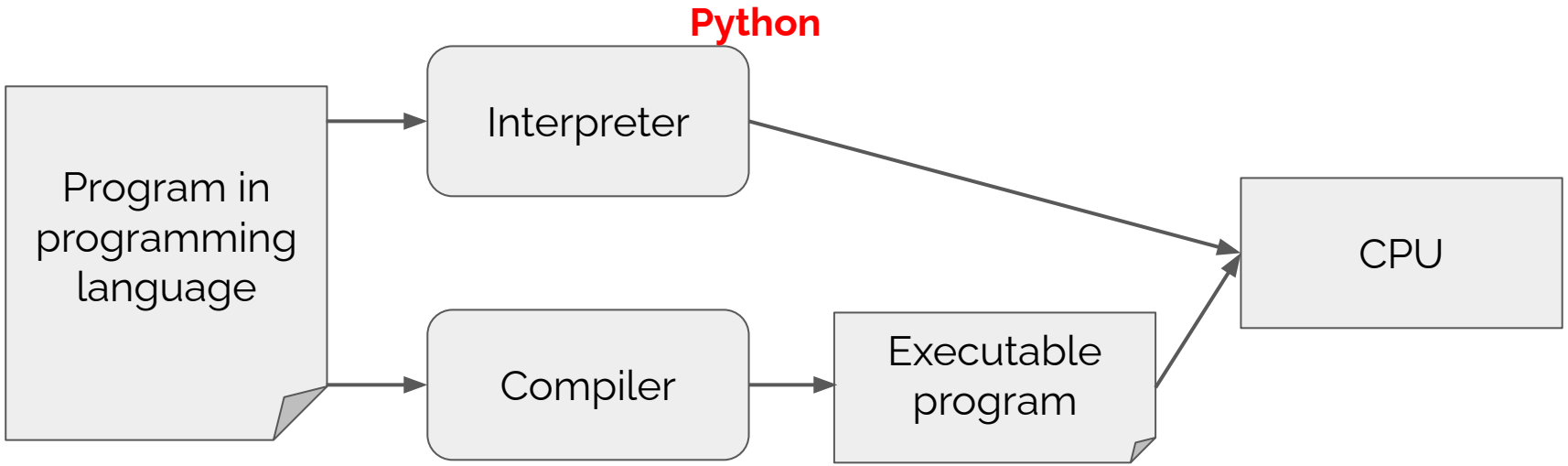
# Programming languages

A way to write instructions that is more convenient than machine code, so to be able to write more complex programs.



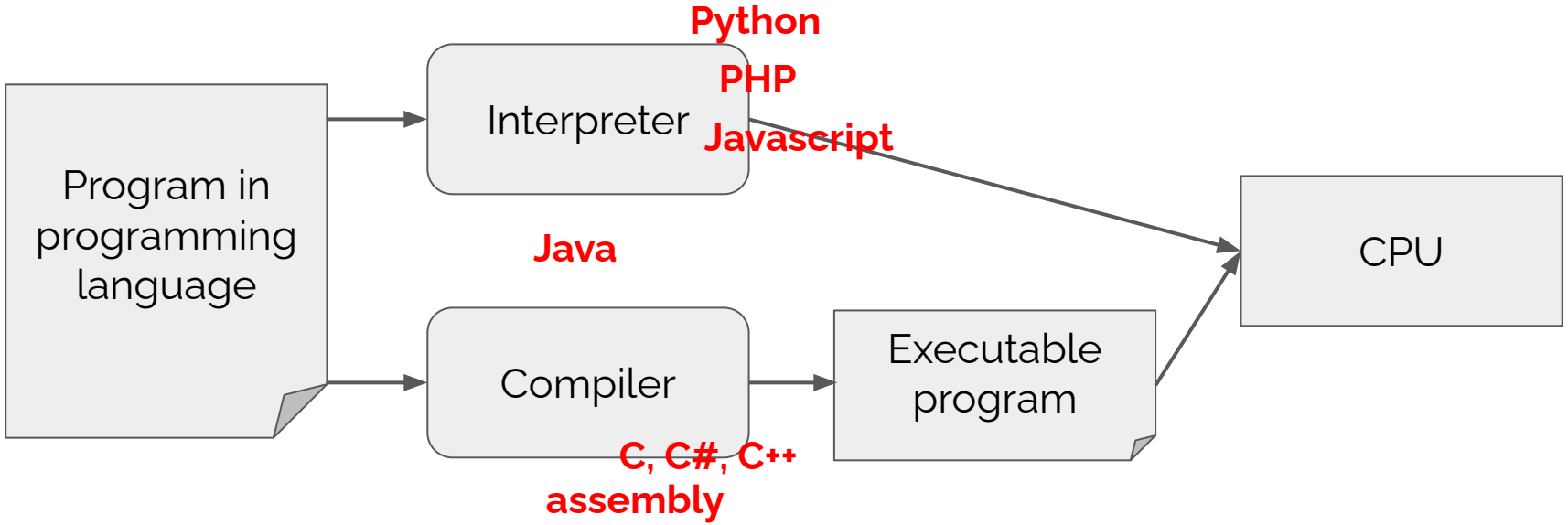
# Programming languages

A way to write instructions that is more convenient than machine code, so to be able to write more complex programs.



# Programming languages

A way to write instructions that is more convenient than machine code, so to be able to write more complex programs.



# Different flavours of programming languages

**Imperative**, for example Python, PHP, Javascript, C

**Functional**, for example Lisp, CAML

**Object-oriented**, for example Python, Javascript, Java, C++, C#

**Scripting**, for example Shell/Bash, Python

# Why learning python is the right choice

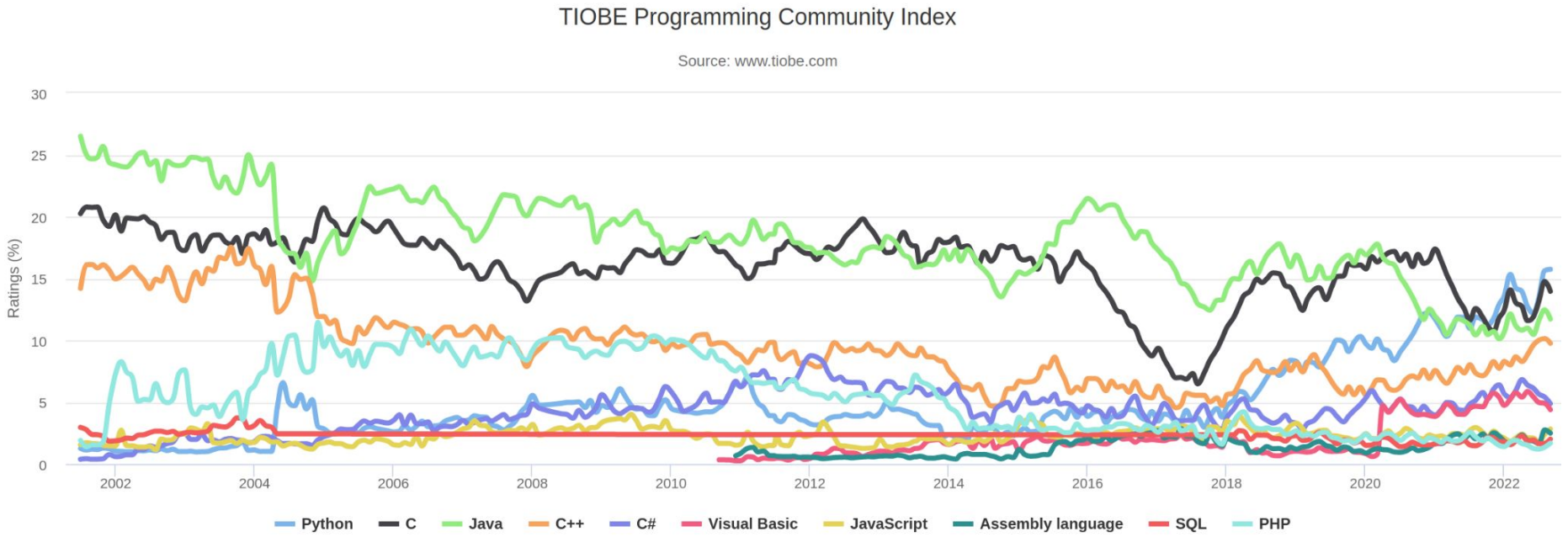
Flexible, intuitive, simple, and popular.



```java
class HelloWorld {
    public static void main(String[] args){
        System.out.println("hello world!");
    }
}
```

```python
print("hello world")
```

# Why learning python is the right choice




Source: https://www.tiobe.com/tiobe-index/ September 2022

# Why learning python is the right choice

| Sep 2023 | Sep 2022 | Change | Programming Language | Ratings | Change |
|---|---|---|---|---|---|
| 1 | 1 | | Python | 14.16% | -1.58% |
| 2 | 2 | | C | 11.27% | -2.70% |
| 3 | 4 | ▲ | C++ | 10.65% | +0.90% |
| 4 | 3 | ▼ | Java | 9.49% | -2.23% |
| 5 | 5 | | C# | 7.31% | +2.42% |
| 6 | 7 | ▲ | JavaScript | 3.30% | +0.48% |

Source: https://www.tiobe.com/tiobe-index/ September 2023

# Python

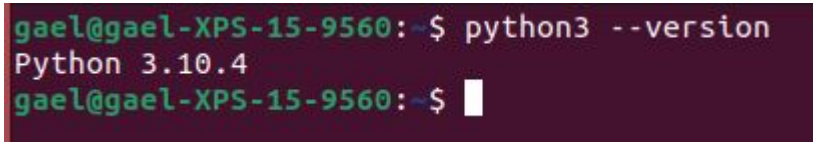
Python is a high level programming language under free licence. It was created by Guido Van Rossum but since 2001 the development is lead by the Python Software Foundation (Guido was the "*benevolent dictator for life*" until july 2018).

Language evolutions are detailed in Python Enhancement Proposals (PEP).

# Working with Python - versions

Python keeps evolving, and has new versions regularly. The most important thing is to choose between Python 2 and Python 3 since some libraries are meant to work for one or the other, but not both.

Python 2 is not supported anymore.




**Here, we will use Python 3**

TL;DR: Python 2 is dead, all hail Python 3

# Working with Python - installation

Command may be python3, python

1. Download the latest Python (Python3 not 2) from here: https://www.python.org/downloads/
2. Install it (if you are using Ubuntu you already have Python installed)

Check if you have the correct Python version

```
gael@gael-XPS-15-9560:~$ python3 --version
Python 3.10.4
gael@gael-XPS-15-9560:~$
```

Check if you have pip

```
gael@gael-XPS-15-9560:~$ pip3 --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

If not: Download https://bootstrap.pypa.io/get-pip.py then execute

```
gael@gael-XPS-15-9560:~$ python3 get-pip.py
```

# Working with Python - installation

- Install Windows' Terminal: 
  https://apps.microsoft.com/store/detail/windows-terminal
  - Support Bash commands
  - Better UI
  - Official app from Microsoft
- Python command is py

```
py mysuperprog.py
```

# Working with Python - environments

Different environments can be used to work with Python:

- A **simple text editor** and the **python interprete**r, e.g. notepad
- A **dedicated python editor**, e.g. SublimeText
- An **IDE (integrated development environment)**, e.g. VSCode, IDLE, PyCharm
- **Jupyter notebooks**., Google Colab, Jupyter Lab, etc.

# Python native functions

To know what does each native Python functions, you need to access the documentation.
Type : help(print)
q to exit the doc

In your browser : https://docs.python.org/3/library/functions.html
or : DevDocs
Famous PEPs: PEP 20 "Zen of Python" ( import this in python console)
PEP 8 "Style Guide for Python Code"

"*There should be one– and preferably only one –obvious way to do it.*"

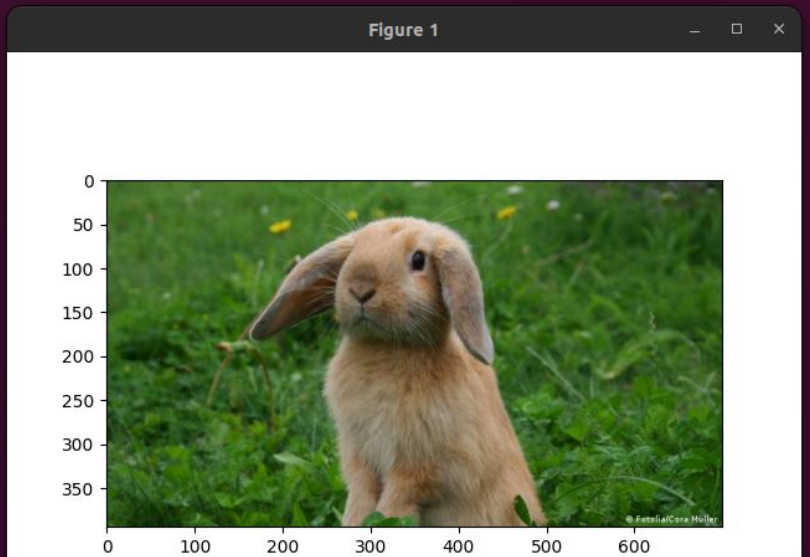
```
# show the code guidelines for Python
import this
```

# Working with Python - example program

```
import sys
from datetime import datetime
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

# tries to parse the DoB and returns
# the number of years (0 if error)
# and whether this is today
def parseDOB(dobString):
  try:
    date = datetime.strptime(dobString,
"%d/%m/%y")
  except:
    return 0,False
  now = datetime.now()
  diff = int((now - date).days) // 365
  return diff,(date.day==now.day and
date.month==now.month)

# displays n candles
def candles(n):
  l1 = l2 = ""
  for i in range(0,n):
    l1 += "()"
    l2 += "||"
  print(l1)
  print(l2)

# programme starts here
name = "bob"
version = 3.5
print(f"Hello, my name is {name}, version {version}")
print("what is your name?")
yourName = input()
print(f"Nice to meet you {yourName}, how old are
you?")
yourAge = int(input()) # this will through an error if
you don't input a number
if yourAge > 40:
  print("That's old! When were you born? (dd/mm/yy)")
else:
  print("Great to talk to a young human! When were you born?
(dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
  yourDOB = input()
  calAge,birthday = parseDOB(yourDOB)
  if calAge <= 0:
    print("please enter a date in the past in the format dd/mm/yy")
if calAge != yourAge:
  print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
  sys.exit(-1)
if birthday:
  print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge}
candles:")
  candles(yourAge)
else:
  print("Happy unbirthday! Here is a rabbit:")
  img = mpimg.imread('rabbit.jpg')
  imgplot = plt.imshow(img)
  plt.show()
```

# Python interpreter

```
gael@gael-XPS-15-9560:~/Documents/nancy/cours/2022-2023/m1_TAL_programming_python$ python3 prog_m1_session_1.py
Hello, my name is bob, version 3.5
what is your name?
Student Maniac
Nice to meet you Student Maniac, how old are you?
22
Great to talk to a young human! When were you born? (dd/mm/yy)
13/09/2000
Happy unbirthday! Here is a rabbit:
```



# Notebook (Google Colab)

## Example program

This is a basic program in python to illustrate components and aspects python programming. It asks a few questions and give messages depending on the answer. You can donwload it as python code (.py) or a Jupyter Notebook (.ipynb) using the File>Download menu.

**Note:** The program requires an image of a rabbit called "rabbit.jpg" to be placed in the directory where the program is run. Here, you can add it using the files section (icon that looks like a folder).

```
[1] # library imports
    import sys
    from datetime import datetime
    import matplotlib.pyplot as plt
    import matplotlib.image as mpimg
```

## Function definitions

This section defines two simple functions that will be used in the code later.

```
[2] # tries to parse the DoB and returns the number of years (0 if error)
    # and whether this is today
    def parseDOB(dobString):
       try:
          date = datetime.strptime(dobString, "%d/%m/%Y")
       except:
           return 0,False
       now = datetime.now()
       diff = int((now - date).days) // 365
       return diff,(date.day==now.day and date.month==now.month)
```

```
[3] # displays n candles
    def candles(n):
      l1 = l2 = ""
      for i in range(0,n):
        l1 += "()"
        l2 += "||"
      print(l1)
      print(l2)
```

## Start of the program

This is where the program actually starts. It greets the user, asks for a name and an age.

```
[7] # programme starts here
    name = "bob"
    version = 3.10
    print(f"Hello, my name is {name}, version {version}")
    print("what is your name?")
    yourName = input()
    print(f"Nice to meet you {yourName}, how old are you?")
    yourAge = int(input()) # this will through an error if you don't input a number
```

```
Hello, my name is bob, version 3.1
what is your name?
Python Maniac
Nice to meet you Python Maniac, how old are you?
22
```

## Birthday

This part of the program asks for the birthdate of the user, check that it is correct, and display something different depending on whether it is today or not.

```
if yourAge > 40:
  print("That's old! When were you born? (dd/mm/yy)")
else:
  print("Great to talk to a young human! When were you born? (dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
    yourDOB = input()
    calAge,birthday = parseDOB(yourDOB)
    if calAge <= 0:
       print("please enter a date in the past in the format dd/mm/yy")

if calAge != yourAge:
  print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
  sys.exit(-1)

if birthday:
  print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge} candles:")
  candles(yourAge)
else:
  print("Happy unbirthday! Here is a rabbit:")
  img = mpimg.imread('rabbit.jpg')
  imgplot = plt.imshow(img)
  plt.show()
```

```
Great to talk to a young human! When were you born? (dd/mm/yy)
10/09/2022
please enter a date in the past in the format dd/mm/yy
10/09/2000
Hey! Today is your birthday! Happy bithday! Here are 22 candles:
()()()()()()()()()()()()()()()()()()()()()()
||||||||||||||||||||||||||||||||||||||||||||
```

# Notebook (Google Colab)

## Example program

This is a basic program in python to illustrate components and aspects python programming. It asks a few questions and give messages depending on the answer. You can donwload it as python code (.py) or a Jupyter Notebook (.ipynb) using the File>Download menu.

**Note:** The program requires an image of a rabbit called "rabbit.jpg" to be placed in the directory where the program is run. Here, you can add it using the files section (icon that looks like a folder).

```
[1]  # library imports
     import sys
     from datetime import datetime
     import matplotlib.pyplot as plt
     import matplotlib.image as mpimg
```

## Function definitions

This section defines two simple functions that will be used in the code later.

```
[2]  # tries to parse the DoB and returns the number of years (0 if error)
     # and whether this is today
     def parseDOB(dobString):
        try:
           date = datetime.strptime(dobString, "%d/%m/%Y")
        except:
            return 0,False
        now = datetime.now()
        diff = int((now - date).days) // 365
        return diff,(date.day==now.day and date.month==now.month)
```

```
[3]  # displays n candles
     def candles(n):
       l1 = l2 = ""
       for i in range(0,n):
         l1 += "()"
         l2 += "||"
       print(l1)
       print(l2)
```

## Start of the program

This is where the program actually starts. It greets the user, asks for a name and an age.

```
[7]  # programme starts here
     name = "bob"
     version = 3.10
     print(f"Hello, my name is {name}, version {version}")
     print("what is your name?")
     yourName = input()
     print(f"Nice to meet you {yourName}, how old are you?")
     yourAge = int(input()) # this will through an error if you don't input a number
```

```
Hello, my name is bob, version 3.1
what is your name?
Python Maniac
Nice to meet you Python Maniac, how old are you?
22
```

## Birthday

This part of the program asks for the birthdate of the user, check that it is correct, and display something different depending on whether it is today or not.

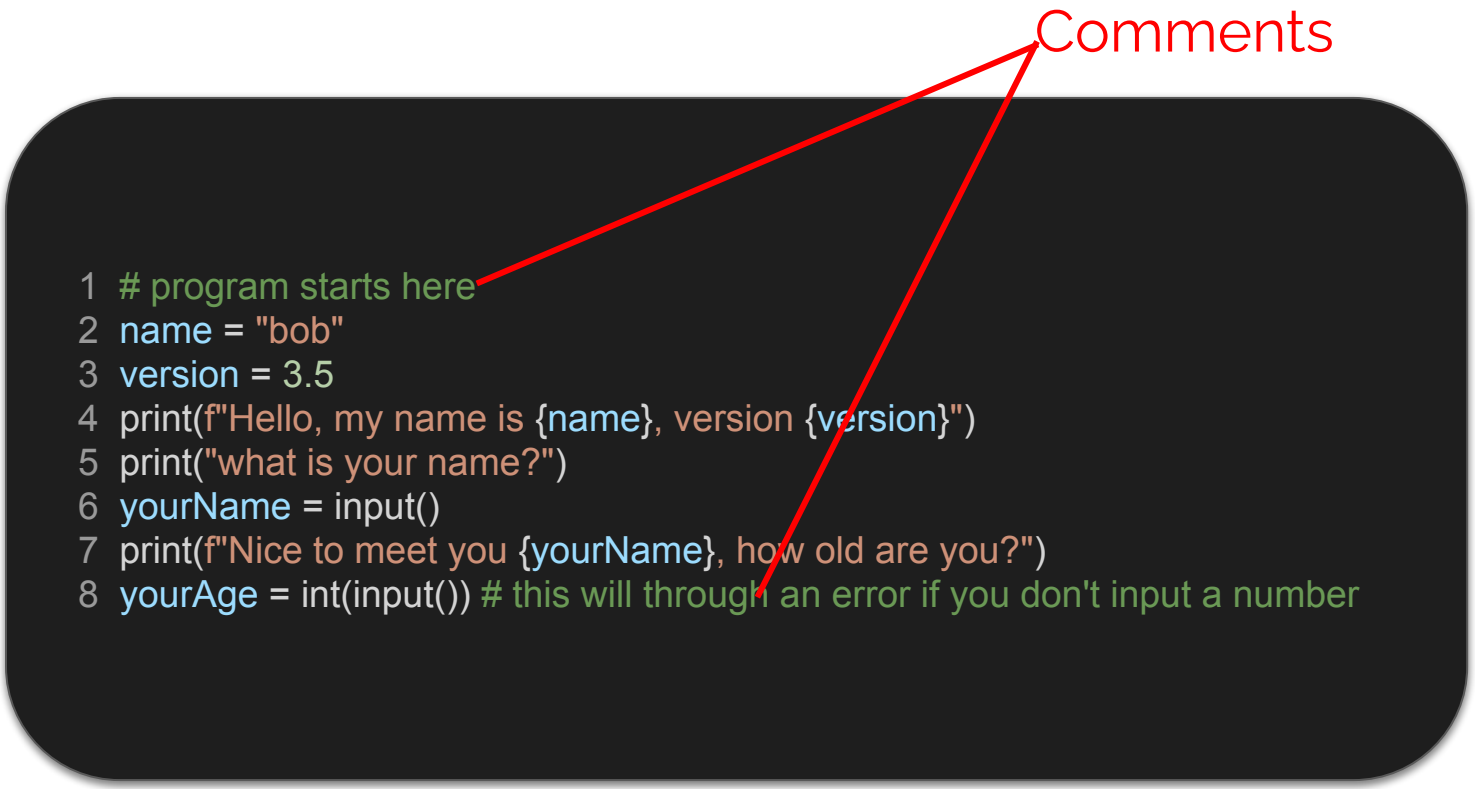
```
if yourAge > 40:
  print("That's old! When were you born? (dd/mm/yy)")
else:
  print("Great to talk to a young human! When were you born? (dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
    yourDOB = input()
    calAge,birthday = parseDOB(yourDOB)
    if calAge <= 0:
       print("please enter a date in the past in the format dd/mm/yy")

if calAge != yourAge:
  print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
  sys.exit(-1)

if birthday:
  print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge} candles:")
  candles(yourAge)
else:
  print("Happy unbirthday! Here is a rabbit:")
  img = mpimg.imread('rabbit.jpg')
  imgplot = plt.imshow(img)
  plt.show()
```

```
Great to talk to a young human! When were you born? (dd/mm/yy)
10/09/2022
please enter a date in the past in the format dd/mm/yy
10/09/2000
Hey! Today is your birthday! Happy bithday! Here are 22 candles:
()()()()()()()()()()()()()()()()()()()()()()
||||||||||||||||||||||||||||||||||||||||||||
```

# Going through the example

```
1  # program starts here
2  name = "bob"
3  version = 3.5
4  print(f"Hello, my name is {name}, version {version}")
5  print("what is your name?")
6  yourName = input()
7  print(f"Nice to meet you {yourName}, how old are you?")
8  yourAge = int(input()) # this will through an error if you don't input a number
```

# Going through the example



# Comments in python

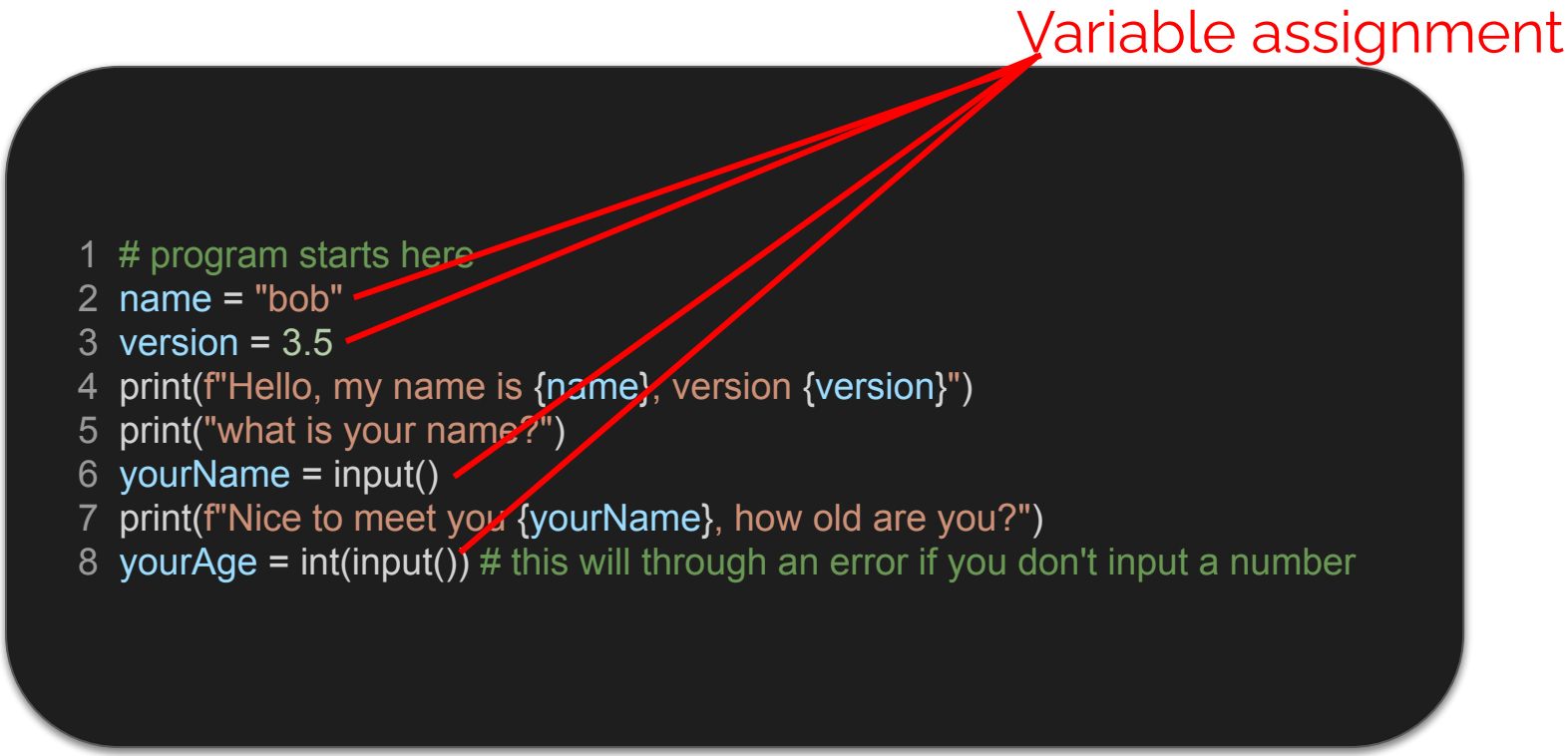
**Single-line comment**: use the # character

```
1 # this whole like gets ignored
2 name = "bob" # anything after # gets ignored
```

**Multi-line comment**: use three double or single quotes

```
1 """
2 Anything in here
3 Gets ignore even if I
4
5 Skip a line
6 """
```

# Going through the example



# Variables in Python

A variable is a name associated with an area in memory, in which we can store values.

```
name = "bob"
```

Creates a variable named “name” and store as its value the string “bob”.

```
version = 3.5
```

Creates a variable named “version” and store as its value the number 3.5.

```
version = 3.6
```

Changes the value stored in the variable version to the number 3.6.

```
x = y = 2
```

Makes both the variables x and y point to the value 2 in memory.

# Variables in Python

A variable can be assign the result of calculating a more complex expression, possibly involving other variables,

```
1  pi = 3.14
2  radius = 5
3  perimeter = 2 * pi * radius
```

or of the value returned by a function (tackled in later course)

```
1  yourName = input()
2  diff = int((now - date).days) // 365
```

# Variable types in Python

A variable has a type, which can be known using the type function

```
1 X = 2.3
2 type(x) # <class ‘float’>
3 x = "I'm sure this will come out as a string"
4 type(x) # <class ‘str’>
```

You can convert from one type to the other using **typecasting**

```
1 x = 2.3
2 type(x) # <class 'float'>
3 y = str(x)
4 type(y) # <class 'str'>
```

# Basic (builtin) types in Python

Types that can be used “out-of-the box” in python include:

*Simple types:*

- **Boolean (bool)**: `True` or `False`
- **Integer (int)**: `3, 687, -12987, 0`
- **Float (float)**: `3.14, -12.4561, 1.0, 35e-8`
- **String (str)**: `“bob”, ‘python is a great language’`

*More complex types:*

- **Complex (complex)**: `3j, 5+2j`
- **List (list):** `[1,2,3], [“bob”, “likes”, “chips”] [1.2, x, “jeff”]`
- **Tuple (tuple)**: `(1,2, 1,x,True)`
- **Range (range)**: `range(10), range(0,10), range(3, 12, -1)`
- **Dictionary (dict):** `{‘name’: “Player One”, ‘age’: 22}`

# Booleans (bool)

Indicate if something is True or False. Examples:

```
1 courseTopic = "python"
2 goodCourse = True
3 courseOnPython = courseTopic == "python"
4 goodCourseOnPython = goodCourse and courseOnPython
5 print(goodCourseOnPython) >>> True
```

From the example programme, with `date.day` and `date.month` being the birthday's day and month, and now.day/now.month being the day and month of today:

```
date.day==now.day and date.month==now.month
```

is True if the birthday is today.

# Operations on booleans

## Logical operations

- A *and* B - True if both booleans A and B are true
- A *or* B - True if A is true, B is true or both are true
- *not* A - True if A is false

## Comparisons producing booleans

- >, <, <=, >= - between numbers and other types
- ==, != - between objects, compare their values (value (in)equality)
- is, is not - between objects, compare the objects (object (in)equality)

# Numbers (int, float, complex)

**Operations on numbers:**

- `x+y, x-y` - Addition and subtraction (`3+4, 3.5-7`)
- `x*y, x/y` - multiplication, division (`12*46, 3.4/8`)
- `x//y, x%y` - Integer (or floor) division, modulo (`3//4, 12%5`)
  -> `x//y` is exactly the same as `int(x/y)`
- `-x` - changes the sign (x=2, y=-x ⟶ y == -2)
- `x**y` (or pow(x,y)) - power (`2**4`, pow(2,4))
- `abs(x)` - absolute value (x=-36, y=abs(x) ⟶ y == 36)
- ...

# Strings

Strings (str) in python are sequences of characters. They can be declared using single or double quotes:

```
1  x1 = "I'm a string"
2  x2 = 'I am a "better" string'
```

**Common operations on strings:**

- `x+y` - concatenation (“bob”+’ is the best’ → “bob is the best”)
- `x[y]` - get the character at index y in the string x (x=”bob”, x[1] == “o”)
- `len(x)` - length of the string x (x=”bob”, len(x) == 3)

We will see more complex operations on strings later, including multi-line strings, substring, search, etc.

# Going through the example

```
1 # program starts here
2 name = "bob"
3 version = 3.5
4 print(f"Hello, my name is {name}, version {version}")
5 print("what is your name?")
6 yourName = input()
7 print(f"Nice to meet you {yourName}, how old are you?")
8 yourAge = int(input()) # this will through an error if you don't input a number
```

## Things we have seen:

- variables
- strings
- floats
- typecast

```
1 # program starts here
2 name = "bob"
3 version = 3.5
4 print(f"Hello, my name is {name}, version {version}")
5 print("what is your name?")
6 yourName = input()
7 print(f"Nice to meet you {yourName}, how old are you?")
8 yourAge = int(input()) # this will through an error if you don't input a number
```

**Things we will see:**

- print
- input
- string formatting

```
1 # program starts here
2 name = "bob"
3 version = 3.5
4 print(f"Hello, my name is {name}, version {version}")
5 print("what is your name?")
6 yourName = input()
7 print(f"Nice to meet you {yourName}, how old are you?")
8 yourAge = int(input()) # this will through an error if you don't input a number
```

```
if yourAge > 40:
 print("That's old! When were you born? (dd/mm/yy)")
else:
 print("Great to talk to a young human! When were you born? (dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
  yourDOB = input()
  calAge,birthday = parseDOB(yourDOB)
  if calAge <= 0:
    print("please enter a date in the past in the format dd/mm/yy")

if calAge != yourAge:
 print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
 sys.exit(-1)

if birthday:
 print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge} candles:")
 candles(yourAge)
else:
 print("Happy unbirthday! Here is a rabbit:")
 img = mpimg.imread('rabbit.jpg')
 imgplot = plt.imshow(img)
 plt.show()
```

```
if yourAge > 40:
 print("That's old! When were you born? (dd/mm/yy)")
else:
 print("Great to talk to a young human! When were you born? (dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
  yourDOB = input()
  calAge,birthday = parseDOB(yourDOB)
  if calAge <= 0:
    print("please enter a date in the past in the format dd/mm/yy")

if calAge != yourAge:
 print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
 sys.exit(-1)

if birthday:
 print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge} candles:")
 candles(yourAge)
else:
 print("Happy unbirthday! Here is a rabbit:")
 img = mpimg.imread('rabbit.jpg')
 imgplot = plt.imshow(img)
 plt.show()
```

## Things we have seen:

- comparisons
- int
- bools

```
if yourAge > 40:
 print("That's old! When were you born? (dd/mm/yy)")
else:
 print("Great to talk to a young human! When were you born? (dd/mm/yy)")
calAge = birthday = 0
while calAge <= 0:
  yourDOB = input()
  calAge,birthday = parseDOB(yourDOB)
  if calAge <= 0:
    print("please enter a date in the past in the format dd/mm/yy")

if calAge != yourAge:
 print(f"Liar, you said your were {yourAge}! I'm out of here! {name} out.")
 sys.exit(-1)

if birthday:
 print(f"Hey! Today is your birthday! Happy bithday! Here are {yourAge} candles:")
 candles(yourAge)
else:
 print("Happy unbirthday! Here is a rabbit:")
 img = mpimg.imread('rabbit.jpg')
 imgplot = plt.imshow(img)
 plt.show()
```

## Things we will see:

- if statements
- loops
- tuples
- packages and libraries

# Textual Role Playing Game

```
1  # Get the input function and display a greeting message
2  print("Hello Player One, what's your name?")
3  playerName = input()
4  print("Welcome", playerName)
5
6  # Example room pattern
7  print('''Your are in a tiny room. Humidity fills the air but your stomach, reminds you that you are very hungry. You are in front of two doors. Behind the first one you can hear muffled voices. Behind the second one you can smell something intriguing.''')
8  print('Which door do you choose? Type "1" for the first room and "2", for the second room.')
9  selection = input()
10 if selection == '1':
11         print('Game Over! Try again', playerName)
12 elif selection == '2':
13         print('Upon opening the door, you can see a huge fest with exquisite meals, everywhere.')
```

# A word on errors

3 types of errors:

- **Syntactic errors**: What you have written is not proper python

```
1  x = "left
2       hello = 2+3
```

- **Semantic errors (including exceptions)**: The syntax is correct but what you have written does not make sense to the interpreter

```
1  x = "bob"+3.5
2  x = int("notanumber")
```

- **Bugs**: What you have written makes sense to the interpreter, but does not do what it is supposed to do.

# To be seen in labs

Familiarizing yourself with the python environment

Do some calculations

Create a character creation mockup

**To do in advance of the lab:**

Install Python3 on your own device

Install an editor or IDE if you want to use one

Install Jupyter Notebook