

# COMP 204: Computer Programming for Life Sciences

Writing and Running Python Program

Mathieu Blanchette,  
based on material from Yue Li, Carlos Oliver and Christopher  
Cameron

# What is a computer program?

**Computer program:** Simple text file containing one or more instructions, written using a programming language

**Programming language:** A language used by programmers to give instructions to a computer (Python, Java, C++, etc.)

**Syntax:** The rules of the programming language.

- ▶ Every language has its own syntax.
- ▶ A program that contains syntax errors cannot be interpreted by the interpreter, so it cannot be executed

**Semantics:** Logic of the sequence of steps taken to obtain the desired output.

- ▶ A program that is syntactically correct but semantically incorrect will run but will produce the wrong output
- ▶ Correcting semantic errors is called debugging. Most of the time doing programming is spent doing this

**Execution of a program:** A program gets executed one line at a time, starting with the first line.

- ▶ However there are ways to control this using control flow statements: conditional statements, loops, etc.

# The Python programming language

Python is a popular and very powerful programming language

Python is an interpreted language

- ▶ This means that the computer reads and executes each line of code one by one.
- ▶ there exists many Python interpreters: Cython, PyPy, Jython,

Python is used everywhere:

- ▶ Large and small software companies: Google, Dropbox, etc.
- ▶ Science: NASA, pharmaceutical companies, academic research
- ▶ Deep learning libraries: Tensorflow, PyTorch, Keras, Theano

Advantages:

- ▶ Succinct and easy to pick up syntax.
- ▶ Lots of built in functionality (sorting, max, etc.)
- ▶ Tons of specialized modules for science, math, databases, visualization, AI, games, web development, etc.
- ▶ Free, open source, community driven

Main drawback: Relatively slow execution

# Writing a program

A program can be written using any simple text editor.

- ▶ Sublime, Emacs, Vi, etc.
- ▶ NOT RECOMMENDED: Notepad (Windows), TextEdit (Mac), Word (simple text)

Or using an Integrated Development Environment (IDE)

- ▶ Wing 101 (simple to use, limited functionality). Used for this course <https://wingware.com/downloads/wingide-101>
- ▶ Eclipse, PyCharm, Spyder, etc.
- ▶ IDEs have many advantages:
  - ▶ Specialized editor (syntax highlighting, auto-completion, etc.)
  - ▶ Integrated running environment
  - ▶ Debugging tools
- ▶ Or using a Python notebook (more on this later)

# Installing Python

See detailed instructions in Tutorial slides available on MyCourses ( Content )

Download Anaconda from <https://www.anaconda.com/download>

- ▶ Selection Python 3.7
- ▶ Select the appropriate version for your computer (Mac vs Windows vs Linux)

Once downloaded, install the package (double-click the downloaded package, follow instructions)

# Getting started with Spyder

Live demo on Spyder!

Creating a new Python program:

- ▶ File → New. Then, File → SaveAs hello\_world.py
- ▶ Note about file names:
  - ▶ should always end with .py
  - ▶ Avoid using spaces or special characters ! @ # % & \* ( ) in file names

Enter text editor window:

---

```
print("Hello World")
```

---

Running your program: several options

- ▶ Run in normal mode: Run → Run (or Green Play button):  
Executes the entire program
- ▶ Step mode: Debug → Debug (or blue Play/pause button),  
then Debug → Step : pauses before the execution of each line  
of the program, to let you observe the execution step by step

Output appears in iPython Console: Hello World

A multi-line program:

---

```
print("Hello World!")  
print("Welcome to COMP 204!")  
print("This isn't too hard!")
```

---

Step-through the program, look at output in the iPython Console.  
Note: To test a line a code without executing the whole program:  
Paste it the iPython console window.

# Errors

Syntax errors: Try the code below, see the iPython Console

---

```
print("Hello World")  
printi("Welcome to COMP 204")  
print("This isn't too hard!")
```

---



# Data types

In Python, data comes in different native types:

- ▶ Strings (called str): sequence of zero or more characters.
- ▶ Integers (called int): Any positive or negative integer: 17, 0, -53, 64729237463928
- ▶ Decimal numbers (called float): Any decimal number: 3.1416, -2.43, 0.0
- ▶ Boolean (called bool): True or False
- ▶ and many more we will encounter later

To know the type of an object, use the type function:

---

```
type("Yue") # returns <class 'str'>
type(29.34) # returns <class 'float'>
```

---

In Python, data types are automatically handled by the interpreter. However, in other languages such as Java or C, we will need to declare the specific type of variable before we use it.

# Comments

Comments are pieces of text that are present in the program but are disregarded by the interpreter

Any text that follows `#` is considered a comment (except within strings)

---

```
# Author: Mathieu Blanchette
# This is my first Python program
print("Hello World")
print("Welcome to COMP 204") # Fall 2019
print("Do not dial # 911")
```

---

Comments are useful to:

- ▶ Provide explanation about what codes does, how it works
- ▶ Help you remember how your program works, and other understand it
- ▶ Indicate authorship

# Strings

A string is a sequence of zero or more characters delimited by single or double quotes.

---

```
"Hello" # This is a string
'My name is Mathieu' # This is also a string
'4236' # Yes, this is a String, because of the quotes
4236 # This is a number, not a String
"" # an empty string
" " # a string consisting of a space

# A string defined with '...' can contain " but not '
'He said: "The answer is 42"'
```

  

```
# A string defined with "... " can contain ' but not "
"He said: it's 42"
```

  

```
# use backslash to use quote between quotes
"He said: \"yes, it's 42\""
```