

UNIX environment - terms and definitions

Xubuntu - the OS we are using in lab

terminal - text-based interface for working with files, folders, and python

unix commands - ls, cd, mkdir, mv, rm, cp, ssh, man, less

shell - the name of the program which executes our commands in the terminal

command line - the current line in terminal, where we type commands

command prompt - the prefix text at the beginning of the command line

Python

Example of a **programming language**

Programming languages allow us to implement **algorithms** on a computer

Algorithms are sequences of steps that perform a task (ex. recipe, or lego instructions)

Programming languages need to balance **expressiveness** with **unambiguousness**.

Python can run in **shell mode** (e.g. in an interpreter) or **script mode** (e.g. with a file)

Programming - Basic Elements

Data Types

string (str) - text, e.g. "hello"

floats (float) - real numbers, or decimal numbers, e.g. 56.3, -0.0001

integers (int) - whole numbers, 45, 0, -20

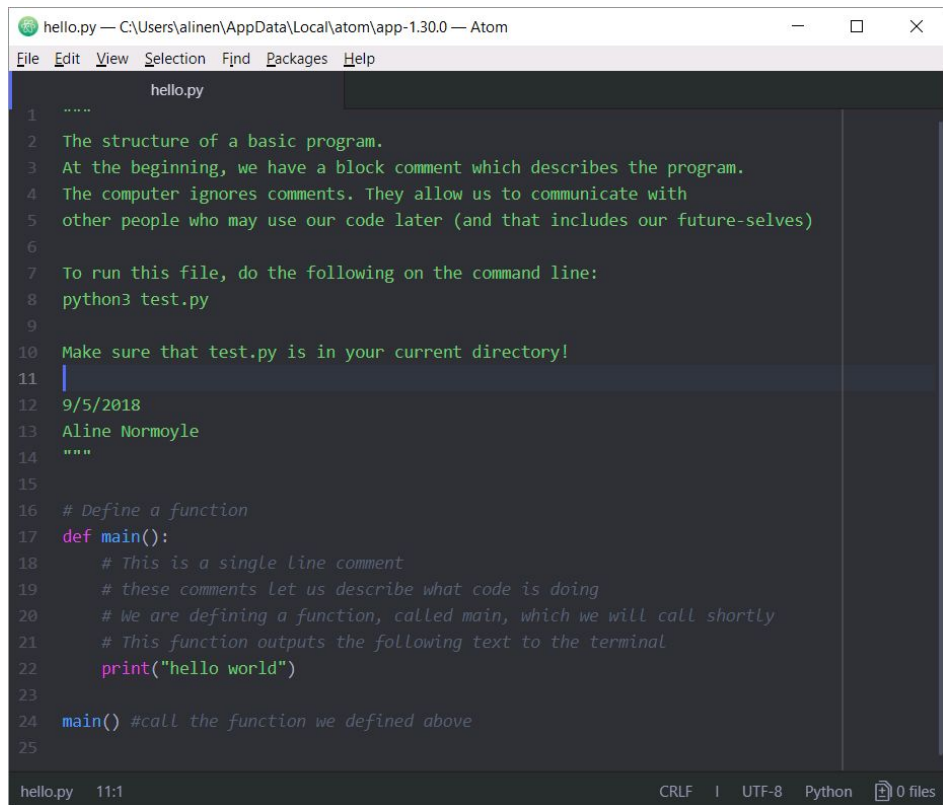
Basic Commands

Printing: output to the screen, e.g. `print("hello world!")`

Arithmetic: add, subtract, multiply, divide, e.g. `+`, `-`, `*`, `/`

Input: ask the user for input, e.g. `input("Give me a value: ")`

Structure of a program



```
hello.py — C:\Users\alinen\AppData\Local\atom\app-1.30.0 — Atom
File Edit View Selection Find Packages Help

hello.py
1  """
2  The structure of a basic program.
3  At the beginning, we have a block comment which describes the program.
4  The computer ignores comments. They allow us to communicate with
5  other people who may use our code later (and that includes our future-selves)
6
7  To run this file, do the following on the command line:
8  python3 test.py
9
10 Make sure that test.py is in your current directory!
11 |
12 9/5/2018
13 Aline Normoyle
14 """
15
16 # Define a function
17 def main():
18     # This is a single line comment
19     # these comments let us describe what code is doing
20     # We are defining a function, called main, which we will call shortly
21     # This function outputs the following text to the terminal
22     print("hello world")
23
24 main() #call the function we defined above
25

hello.py 11:1 CRLF UTF-8 Python 0 files
```

Different types of errors

Whenever we make a mistake the computer is happy to tell us about it

```
>>> print("hello
      File "<stdin>", line 1
        print("hell
              ^
      SyntaxError: EOL while scanning string literal
```

The first part of the error tells us **where** the error occurred and the second part tells us **what went wrong**. Don't worry if these errors don't make sense yet. They will. For now, when you get an error, **check for typos** such as bad indentation, mismatched parentheses, missing punctuation, and misspelled names. Also, make sure you always run with **python3**!

Programming - Basic Elements

Variables

Like a container with a label.

We put data in them and give them useful names,

e.g. `cost = 45.5`, `name = "Baby Boo"`, `country = "Mexico"`

Similar to variables in math

Rules: case sensitive, starts with a letter but can contain `_` or integers, some words such as **`def`**, **`print`**, **`input`** are reserved

Programming - Basic Elements

Assignment: setting a value to a variable

e.g `a = 1`, `c = 0`, `a = c`, `greeting = "hello"`,

Type conversions: changing from one type to another

e.g. `a = int("45")` ← converts the string "45" to the number 45

This is useful for converting input to either integers or floats