Intro to programming 3

Henri Vandendriessche
henri.vandendriessche@ens.fr

2022-09-27

Terminal cheat sheet reminder

- Bash commands to navigate directories
  - Print Working Directory. Print the path of the current directory

```
pwd
```

  - List all files of the current directory

```
ls folder
```

  - Moving into folder1 and subfolder2 at once.

```
cd folder1/subfolder2
```

  - Moving out of a directory

```
cd ..
```

  - Going back and forth in the directory tree

```
cd ../../folder1/subfolder1
```

  - Going back to the root directory

```
cd ~
```

- "**Tab**" to use the auto-completion
- **Ctrl + C** to stop a program execution
- Many more bash commands to use...

So far

- Python
- Data types:
  - integer
  - float
  - string
  - boolean
- **If**, **For** and **While** loops:
  - syntax
  - indentation
- Data collections:
  - list
  - tuple
  - set
  - dictionary

Clarification 1

- A **for** loop is used used to iterate over the elements of a sequence (such as a string, tuple, set, list or dictionnary) or other iterable object:
- It differs from other **for** keyword in other programming languages and works more like an iterator.

```
list1 = [1,2,3,0]
for x in list1:
  print(x)
```

```
## 1
## 2
## 3
## 0
```

- It is almost similar to (which is much closer to the "traditional" for in programming)

```
list2 = [1,2,3,0]
for x in range(0,len(list2)):
  print(list2[x])
```

```
## 1
## 2
## 3
## 0
```

Clarification 2

- A **While** loop cannot directly iterate over the elements of a sequence like the **for** loop

```
list1 = [1,2,3,0]
while x in list1:
  print(x)
```

NameError: name ‘x’ is not defined

Today

- Python standard library
- Random numbers and number choices
- Exercises

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below
  - https://docs.python.org/3/library/index.html

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below
  - https://docs.python.org/3/library/index.html
- The "Python library" contains several different kinds of components.

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below
  - https://docs.python.org/3/library/index.html
- The "Python library" contains several different kinds of components.
- It contains data types that would normally be considered part of the "core" of a language, such as numbers and lists.

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below
  - https://docs.python.org/3/library/index.html
- The "Python library" contains several different kinds of components.
- It contains data types that would normally be considered part of the "core" of a language, such as numbers and lists.
- The library also contains built-in functions and exceptions — objects that can be used by all Python code without the need of an import statement. Some of these are defined by the core language, but many are not essential for the core semantics and are only described here.

Python standard library 1/3

- Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below
  - https://docs.python.org/3/library/index.html
- The "Python library" contains several different kinds of components.
- It contains data types that would normally be considered part of the "core" of a language, such as numbers and lists.
- The library also contains built-in functions and exceptions — objects that can be used by all Python code without the need of an import statement. Some of these are defined by the core language, but many are not essential for the core semantics and are only described here.
- This manual (https://docs.python.org/3/library/index.html) is organized "from the inside out:" it first describes the built-in functions, data types and exceptions, and finally the modules, grouped in chapters of related modules.

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.
- Many of the internet protocols are supported

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.
- Many of the internet protocols are supported
- Module for relational databases

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.
- Many of the internet protocols are supported
- Module for relational databases
- It has also modules to create graphical interface

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.
- Many of the internet protocols are supported
- Module for relational databases
- It has also modules to create graphical interface
- It comes with a lot built-in functions (https://docs.python.org/3/library/functions.html)

Python standard library 2/3

- Python's large standard library is one of the greatest strength of Python.
- Many of the internet protocols are supported
- Module for relational databases
- It has also modules to create graphical interface
- It comes with a lot built-in functions (https://docs.python.org/3/library/functions.html)

Built-in Functions

A	E	L	R
abs()	enumerate()	len()	range()
aiter()	eval()	list()	repr()
all()	exec()	locals()	reversed()
any()			round()
anext()	F	M	
ascii()	filter()	map()	S
	float()	max()	set()
B	format()	memoryview()	setattr()
bin()	frozenset()	min()	slice()
bool()			sorted()
breakpoint()	G	N	staticmethod()
bytearray()	getattr()	next()	str()
bytes()	globals()		sum()
		O	super()
C	H	object()	

Python standard library 3/3

- Alongside the standard library, Python Package Index (PyPi), the official repository for third-party Python software contains more than 380 000 packages (as of June 2022)
- Example of famous third-party Python packages on PyPi:
  - Pandas
  - Matplotlib
  - Seaborn
  - Scikit-learn

Python module - example of **Random** 1/4

- Python incorporates in its standard library a multitude of modules for a variety of subjects and problem (network, text processing, mathematics, file and directory access, cryptography. . .)

https://docs.python.org/3/library/index.html

- The standard library include in particular a specific module for random (pseudo-random) number generation

https://docs.python.org/3/library/random.html

Python module - example of **Random** 2/4

- Several ways to import a python module

```
import random # import random
int_list =[1,2,3]
random.shuffle(int_list)# from that object you have to access all the functions
print(int_list)
```

```
## [2, 1, 3]
```

```
import random as rand # import random using a custom local name
rand.shuffle(int_list) # from that object you have to access all the functions
print(int_list)
```

```
## [2, 3, 1]
```

```
from random import shuffle,randint,choice # import only needed function
shuffle(int_list) # use the function directly without object before
print(int_list)
```

```
## [2, 3, 1]
```

```
from random import * # import all the functions bundled inside Random at once
shuffle(int_list)
print(int_list)
```

Python module - example of **Random** 3/4

```
from random import *

print(randint(1, 100))    # Pick a random integer between 1 and 100.
```

```
## 6
```

```
print(uniform(1, 100))    # Pick a random float between 1 and 100.

# prints a random value from the list
```

```
## 45.919101920578804
```

```
list1 = [1, 2, 3, 4, 5, 6]
print(choice(list1))
```

```
## 6
```

```
items = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = sample(items, 4)    # Pick 4 random items from the list
print(y)
```

```
## [7, 5, 6, 8]
```

Python module - example of **Random** 4/4

```
# using randrange() to generate in range from 20
# to 50. The last parameter 3 is the step
# from 20 to 50 with a step of 3 (20 - 23 - 26 ....  47 - 50)
print("A random number from range is : ", end="")
```

```
## A random number from range is :
```

```
print(randrange(20, 50, 3))
```

```
## 47
```

Exercises

- Exercise 1: Lottery pick. Generate 100 random lottery tickets (one ticket is a sequence of 5 digits) and pick one winner out of it.
- Exercise 2: write a program that generates a random 10 character long password including 6 letters with 2 of them uppercase, 1 digit and 1 special symbol.
- Exercise 3: Monte Carlo estimation of Pi: one way to estimate the value of the pi is to generate a large number of random points in the unit square and see how many fall within the unit circle; their proportion is an estimate of the area of the circle. See https://academo.org/demos/estimating-pi-monte-carlo. Implement the proposed algorithm to estimate the value of pi.
- Exercise 4: Write a program that prints the first N rows of Pascal's triangle (see https://www.youtube.com/watch?v=XMriWTvPXHI).