Deep Learning
Exercise "Introduction to Python"

Prof. Dr. Jürgen Brauer

Exercise 1 - Preparing to work with Python

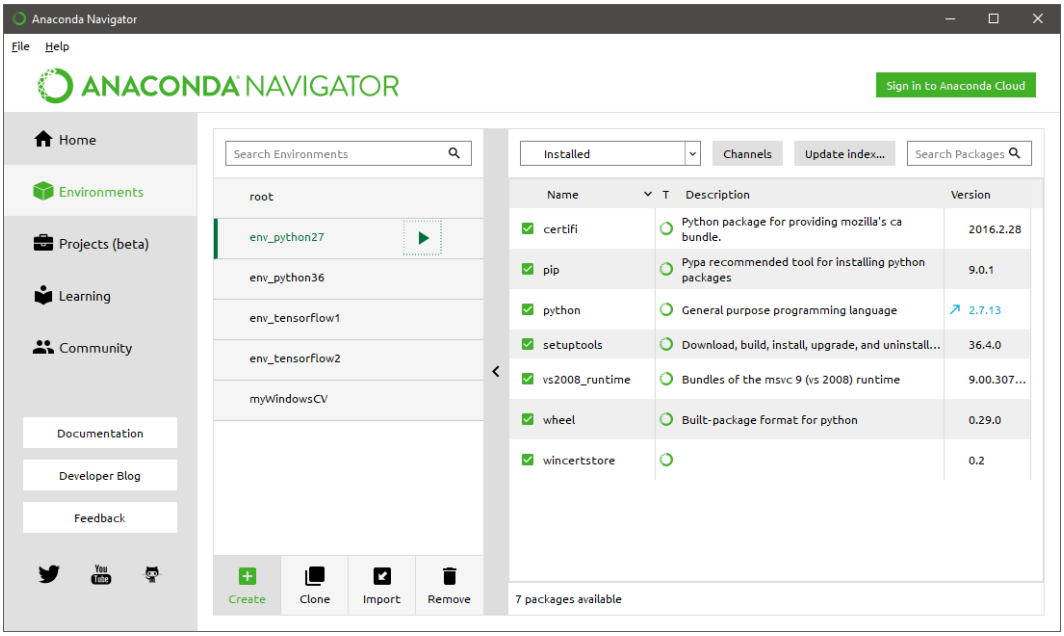
Go to the Python download page:
`https://www.python.org/downloads/`

You will see that there are two different versions of Python that are available: Python 3.6.3 and Python 2.7.14. Which one should I download?

It is important to know that code written in Python 3.x is not compatible with code written in Python 2.x. On the one side Python 2.7 still provides a larger set of packages which has the effect that some programmers choose to remain with Python 2.7. On the other side Python 3 is said to be the future of the language.

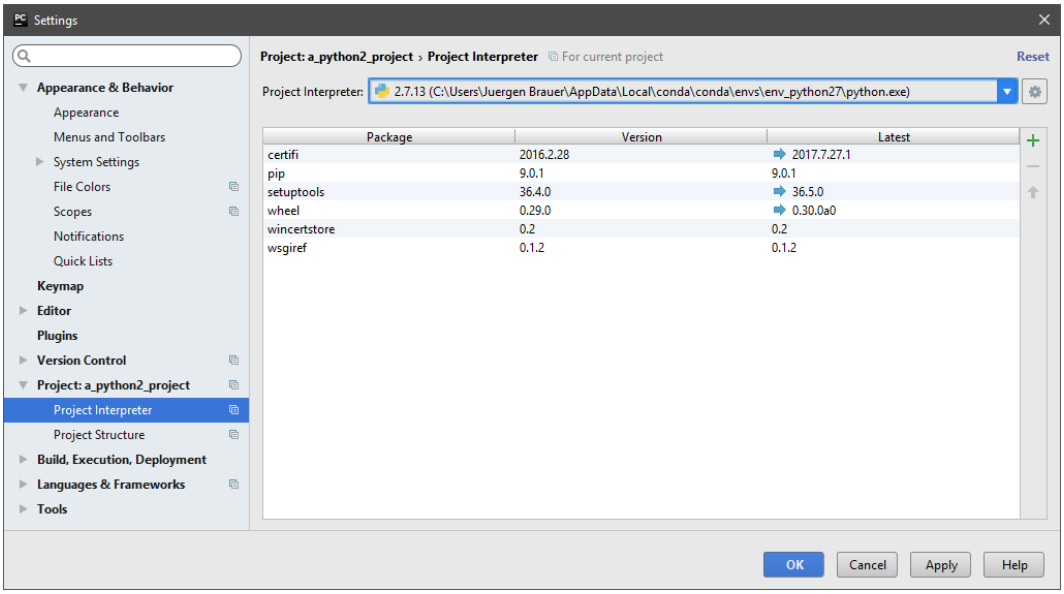
Fortunately, we need not to decide and can use a tool called *conda* to remain flexible. conda allows to create *environments* which can host different versions of Python interpreters and different packages.

Step 1:
Go to `https://www.anaconda.com/download/` and download (Windows) *Anaconda* (64 bit, Python 3.6 version). Anaconda contains the conda package and environment manager that will allow us to create environments with different Python versions.



Step 2:
Start the *Anaconda navigator*. Then create two environments. An environment
`env_python27`
where you choose Python 2.7 as interpreter and another environment
`env_python36`
where you choose Python 3.6 as interpreter.

Step 3:
Install *PyCharm*, a nice Python IDE. Start PyCharm and create a project `a_python2_project`. Open the
`File → Settings → Project Interpreter`
dialog and add the directory where conda stores the `env_python27` environment to the Project Interpreter selection box (Add local) by choosing the Python interpreter in that environment directory. Then choose this newly added interpreter as the project interpreter.



Step 4:
Now add a new file `python2_code.py` to the project, enter the following code and run it:

```
1 import sys
2 print "Your Python version is: " + sys.version
```

It should output something like this:

```
Your Python version is:  2.7.13 |Continuum Analytics, Inc.|
(default, May 11 2017, 13:17:26) [MSC v.1500 64 bit (AMD64)]
```

Step 5:
Now prepare a `a_python3_project`, choose the `env_python36` environment as project interpreter and add a `python3_code.py` source code file to this project with the following code:

```
1 import sys
2 print("Your Python version is: " + sys.version)
```

It should output something like this:

```
Your Python version is:  3.6.2 |Continuum Analytics, Inc.|
(default, Jul 20 2017, 12:30:02) [MSC v.1900 64 bit (AMD64)])]
```

Step 6:
Now search for articles in the internet describing some differences between Python2

and Python3 and augment the code in both project source files `python2_code.py` and `python3_code.py` to show and try out these differences. Try to find as many differences as possible, at least 5.

Exercise 2 - Python syntax

Loops

a) Write a for-loop and a while-loop that print the numbers between 10 and 30 with a step size of 3 in one line:

```
10  13  16  19  22  25  28
10  13  16  19  22  25  28
```

b) Format the following code correctly such that it runs and produces the output specified:

```
counter=0
while(counter<10):
print(counter, end=' ')
counter +=1
else:
print("counter=" + str(counter))

for i in range(1, 10):
if(i%5==0):
break
print(i, end=' ')
else:
print("i=" + str(i))
```

Desired output:

```
0 1 2 3 4 5 6 7 8 9 counter=10
1 2 3 4
```

Dynamic typing

a) How can you get the information which data type Python uses internally for each of the variables a-f? Print the data type for each of the variables.

```
a = 2
b = 3.1
c = 'd'
d = "a string"
e = [1,22,333]
f = (4,55,666)
```

b) Which output will be generated by the following program?

```
b = 21
b = b+b
print(b)
b = "3"
b = b+b
print(b)
print(type(int(b)))
print(type(str(type(int(b)))))
print(str(type(str(type(int(b)))))[3])
```

Selections

a) Let the user enter a number. If the number is between 1-3 output "A", if it is between 4-6 output "B", if it is between 7-9 output "C". If it is not between 1-9 output "Invalid number!". The user shall be able to enter numbers till he enters the word "exit". If he enters a string that is not a number and different from the word "exit", output "Invalid command!"

Example program run:

```
1
A
1.384
A
5.4
B
9
C
9.0
C
9.1
Invalid number!
10
Invalid number!
-1.45
Invalid number!
test
Invalid command!
exit
```

Functions

a) Define a function f1 that accepts two parameters value1 and value2, computes the sum and the product and returns both. What is the type of the "thing" that you return?

b) Now define a function f2 that does the same but provides default values for both arguments (e.g. default value1 = 2, default value2 = 3). So the behavior of your function f2 should look like this without specifying any of the values:

```
print(f2())
(5,6)
```

Is it possible now to call f2 without specifying value1, but only value2? How?

Lists

a) Define a list which stores the strings "cheese", "milk", "water". Output the list. Then append the string "apples". Output the list. Remove the string "milk". Output the list. Iterate over the list and output each element.

b) What does "list comprehension" mean in Python? Use a list comprehension to generate a list of π rounded to the first, second, third, fourth and fifth decimal and output this list:

```
[3.1, 3.14, 3.142, 3.1416, 3.14159]
```

Classes

a) Define a class "car" that accepts the car's name and its maximum speed as parameters in the class constructor. Three attributes shall be stored for a car: its name, its maximum speed and its mileage. Define a method "set_speed" which allows to set the current speed. Also define a method "drive" which accepts as parameter the number of hours to drive and increases the mileage accordingly. Finally, also provide a method "show_status" which outputs the current speed and mileage. Test your class by creating two object instances.

b) Derive a class "convertible" that uses "car" as base class. Pass the time for letting the roof of the convertible down as an argument to the class. Overwrite the "show_status" method of the base class and output also the time to open the roof.