Python in your pocket

Carl Stermann-Lücke, and Zainab Nazari

January 1, 2020

Contents

1 Installation

If you own a Google account you do not need to install anything! This is the easiest that you can reach to a python notebook. In the website:https://colab.research.google.com you can read and write and run the python codes. We generally need to tell apart two different ways of running python:

- scripts
- notebooks

1.1 Mac

1.2 Wondows

1.3 Linux

2 Basics

2.1 Comments

There are two types of comments:

1. line comments using #, using # in the beginning of any line, will make the line as comment and won't be evaluated during the run.
2. multi-line comments using ''' or """, we can comment out a part of our code simply by using it before and after the part of the code. example:

=	assignment	a = 3 → a is a variable with the value 3
+	addition	3 + 2 → 5
-	subtraction	3 - 2 → 1
*	multiplication	3 * 2 → 6
/	division of floating point numbers	3 / 2 → 1.5
//	division of integers	3 // 2 → 1
%	modulo-operator, remainder of an integer division	3 % 2 → 1
**	power	3 ** 2 → 9
>	greater than	3 > 2 → True
<	smaller than	3 < 2 → False
==	check for identity	3 == 3 → True
!=	inverse check of identity	2 != 3 → True
not	boolean inversion	not True → False
or	boolean or	True or False → True
and	boolean and	True and False → False

2.2 Operators

In python we have all the basic mathemetical operators. This includes

- = (assignment of the right-hand-side to the variable on the left-hand-side)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division of floating point numbers)
- // (division of integers)
- % (modulo-operator, remainder of an integer division)
- ** (power)
- not (boolean inversion)
- or (boolean or)
- and (boolean and)

2.3 Indentation

Spacing matters!
Lines of code that are always run after each other are also in the same level of indentation. That means that their first character is the same number of spaces or tabulator-spaces away from the left end of the line.

2.4 conditional statements

This code shows how to write conditional statements:

```
number = 24352
if(number % 2 == 0):
  print(number, "is an even number") #This line is only run if the condition in the if-
    statement is true.
else:
  print(number, "is an odd number") #This line is only run if the condition in the if-
    statement is false.
```

The indented parts are only run under a certain condition. It is also possible to include more branch options using elif:

```
number = 24352
if(number == 0):
  print(number, "is 0") #This line is only run if the condition in the if-statement is true.
elif(number % 2 == 0):
  print(number, "is an even number") #This line is only run if the condition in the elif-
    statement is true and the condition in the if statement (and any possible elif statements
    before this) is false.
else:
  print(number, "is an odd number") #This line is only run if the conditions in the if and the
     elif statements are all false.
```

2.5 while-loops

2.6 functions

3 numpy

3.1 numpy arrays

3.2 slicing