Python Basics

Question 1: Expressions & Types

Every value in Python has a particular type. Values such as 3 and 17 have type int (short for **int**eger), while a value like 3.1415 is of type float (short for **float**ing point number, referring to the decimal point floating between digits). Other types include **str**ings, denoted by a sequence of characters surrounded by either single or double quotes (‘ or “). The **bool**ean type can have the values **True** and **False** (don’t forget the capital and do not use quotes because that would make them of the type string!).

As explained in the theory, Python follows a specific order in which it executes expressions: first expressions surrounded by parentheses are evaluated, then powers, followed by multiplication and division and finally subtraction and addition. Besides those there exist special operators such as the modulo operator % that returns the remainder after division.

a) Look at the following expressions. Try to predict what the result would be after evaluation in Python. First write down what you would expect on this page and afterwards check in your Python editor. Try to predict what the type of the result would be.

Expression	Your prediction	Result in Python
1 + 2 * 3		
(1 + 2) * 3		
2 ** 4		
13 % 5		
7 // 2		
-7 // 2		
7 / 2		
7 / /2.0		
9 % -2		
‘pom’ * 4		
‘blablabla’ / 3		
2+		

In Python, the datatype is predicted automatically based on the values used (= dynamic typing). In case of doubt, the type of a variable can always be checked by using the function **type(**value**)**.

b) There exist ways to change the type of a value in Python. Predict what the result of the following expressions will be if you evaluate them in Python. Think carefully about what type the result will have. First write down what you would expect on this page and afterwards check in your Python editor.

Expression	Your prediction	Result in Python
abs(-2)		
7/2		
float(2)		

float(7//2)		
float(7) // 2		
int(2.6)		
round(2.6)		
round(2.5)		
round(2.4)		

c) Find the result of the following questions by inputting the expressions into Python.

1. The volume of a sphere with radius r is $\frac{4}{3}\pi r^3$. What is the volume in cm³ of a sphere with a radius of 5 cm if π= 3.14159?
2. Tom participated in a running competition of 6 miles in New York. He ran the distance in 53 minutes and 30 seconds. What is the average speed (expressed in km/h) with which Tom finished the race if you know that one mile is equal to 1.61km?

Question 2: Boolean and conditional logic

A **bool**ean is a type that can have the values **True** and **False**. The **boolean operators** are **and**, **or**, and **not**. With the **not** operator True becomes False and vice versa. With **and** all variables have to be True in order to get True as result; with the **or** operator only one value has to be True.

Relational operators (conditional operators): greater than **>** or equal to **>=** , less than **<** or equal to **<=**, equal to **==** (double = otherwise we have an assignment) and not equal to **!=** always produce True or False.

a) Predict what the result is of the following expressions if you input them in Python. First write down what you would expect on this page and afterwards check in your Python editor.

Expression	Your prediction	Result in Python
not False		
True and False		
True or False		
False or not False		
((not True) or False) and (not ((not False) or True))		
not True or False and not not False or True		
3 > 5		
2 <= 2		
“abc” < “bcd”		

Question 3: Variables

From now on we will abandon the Python shell and switch to creating Python files (with extension ".py") in PyCharm.

a) Create two variables *pi* and *r* and assign them the values 3.14159 and 5 respectively. Repeat question 1.c by using those variables.

b) A variable does not have a fixed value (hence its name ;-)). After a first assignment you can overwrite the value of the variable by assigning it another value. Create two variables: *x* with value 4 and *y* with value 5. Look at the three expressions in the table below and predict what would happen when they are inputted in Python. Check your prediction in your editor afterwards.

Expression	Predicted answer	Python answer
x = x + y		
x		
y		

Question 4: Strings

A **str**ing is a type that consists of a sequence of characters. A string is denoted by using single or double quotation marks (' or ").

a) Define two string variables: *name* and *first_name* and assign them your own name and first name. Create a third variable and assign it your full name without having to rewrite it. Warning: there should be a space between your first name and name!

b) Create the string 'bom bom bom bom bom bom bom bom bom bom ' by typing the word 'bom' only once. Make use of the '*' operator.

Question 5: Lists and Tuples

The types **list** and **tuple** can be assigned one or more objects of different type in a certain order. These objects are referred to as elements and can be accessed by their index, ranging from *zero* to *length-1*. (e.g. mylist[1] returns the second element from a list). A list is mutable (meaning that its content can be reordered) and its objects are surrounded by brackets (e.g. [1, 2, 3, 2]). A tuple on the other hand is not mutable and is denoted by parentheses (e.g. ('a', 'b', 'c')).

a) Define a variable 'mylist' and use it to store the first 5 letters of the alphabet.

b) Replace the third letter of 'mylist' by the letter 'z' and print the list to check its content.

c) Create a new variable 'mylist2' and store a list that contains two strings and one integer, i.e. your first name, your name and your age. Print these variables. Now combine the content of both lists and store it in a third variable with the name 'mylist3'. Now print the list 'mylist3' to check its content.

d) Idem to 5a), but now create a tuple instead of a list. What happens when you try to do the same as in question 5b) and replace the value of the third element?

Question 6: Your very first function

Functions in a programming language are very similar to functions in mathematics. Both require one or more parameters used to make calculations and obtain a result. For example in mathematics: $z = f(x, y) = \sqrt{x^2 + y^2}$, where x and y are the arguments and z the result. In python this equation would look as follows:

```
def pythagoras(x, y):
        z = (x**2 + y**2)**0.5
        return z
```

The general structure of a function is always as follows: **def** functionname(parameters): after which the content of the function needs to be indented. A return is used to return the result of the calculations in the function. In general, it is recommended to return the result instead of printing it.

Functions are usually written on top of Python documents but this is not mandatory (but definitely recommended). Just remember that it is impossible to call a function before it is defined! Functions and variables that are defined in other documents or libraries can be called by using the **import** function (see theory).

a) Create a new document and write a function that converts a given temperature from degrees Celsius to Fahrenheit: $°\mathrm{F} = °\mathrm{C} * \frac{9}{5} + 32$. Test your function by calling it in the same document.

b) Write a function that determines if a year is a leap year or not. Remember, a leap year can be divided by 4, but not by 100. However, if the year is divisible by 400 than it is a leap year. Make sure the result of your function is a boolean.

c) Write a function that uses two years. The function determines whether one of the years is a leap year. Hint: use your function of question 6b).

Question 7 (optional): Methods

There are a types in Python that have methods. You can call these methods by using a '.' After a variable of that type.

a) Create a list: 'mylist = [100,200,400,500]'. We want to add the value '300' on the third spot in the list. Try to do so by using the method insert(index, list). After that, add the value 1000 to the list by using the method append(list). Finally, print the sorted list by using the method sort() (without any arguments).

b) Create a string variable that is assigned the following sentence:

"*Zeven Zottegemse zotten zullen zes zomerse zondagen zwemmen zonder zwembroek.*"

a. Count the number of words in that sentence. Hint: strings have the method *split()* that returns a list with each word of that sentence as an element. *len(mylist)* returns the length of your list.
b. Replace for every word the first letter by a 'p'. Hint: the string type has the method *replace(old,new)* , which replaces a substring *old* with another substring substring *new*. Be careful: this method is case sensitive!