TØ 3. Variables & Functions

Table of contents

In this session we will explore variables and functions - both of which are important elements of Python (and many other programming languages)

1 Variables

To really unlock the utility of Python variables are required, among other things they allow us to express calculations in a simpler way. A variable is defined by using `=`, for example, we can define

```
a = 21 + 21
```

This means that the result of the calculation is stored in the variable `a` and can be used in subsequent calculations.

1.1 Exercise: Calculations with variables

Repeat the calculations

1. $21 + 21$
2. $53 - 11$
3. 6×7
4. $\frac{546}{13}$

But now assign each of them to a variable `a`, `b` `c` etc and then calculate

$$(21 + 21) + (53 - 11) + (6 \times 7) + \frac{546}{13}$$

using these variables. In the cell below replace `_____` with your code

```
a = _____
b = _____
c = _____
d = _____
f = _____ # Do not use any numbers on this line.

print(a, b, c, d, f)
```

> **Note**
>
> The cell above used a new thing `print`. This is a *function*, functions are like recipe - they take a number of inputs (ingredients) and perform some predefined set of operations on those and return an output.
>
> The `print`-function prints what it is given to the screen - in this case the five variables. You will see and use the `print`-function numerous times.

1.2 Exercise: A biological calculation

The number of cells in different parts of the body is given below

Type	Number of cells
Total cells	$37 \cdot 10^{12}$
Brain cells	$86 \cdot 10^{9}$
Liver cells	$240 \cdot 10^{9}$
Skin cells	$16 \cdot 10^{11}$

For each cell type, calculate the percentage each category represents of the total number of cells in the body, i.e., for each category calculate

$$P = \frac{\text{Number}}{\text{Total}} \times 100.$$

```
# Make variables for the different number of cells
total_cells = _______
brain_cells = _______
liver_cells = _______
skin_cells = _______

# Calculate the percentages - use the variables!
brain_pct = _______
liver_pct = _______
skin_pct = _______

# Print the results
print("Brain cell percentage: ", brain_pct)
print("Liver cell percentage: ", liver_pct)
print("Skin cell percentage: ", skin_pct)
```

Tip

The numbers of cells are rather large, so typing the correct amount of zeros is a little painful. The code below shows four different ways of writing the same number in Python code

```
13000000000000     # Basic
13_000_000_000_000 # Underscores to visually distinguish
13 * 10**(12)      # Using exponentation
13E12              # Scientific notation
```

2 Functions

When doing repeated calculations it is generally good practice to define a function. Consider for example that you need to calculate the average of several sets of numbers, without using functions we could write

```
A = 1
B = 2
C = 3
D = 4

average_AB = (A + B) / 2
average_BC = (B + C) / 2
average_CD = (C + D) / 2
```

This is perfectly valid, but it is more prone to error - as we have to write the expression multiple times - and it is harder for someone else to decipher - as they have to check they understand what each line of code does. If we instead used a function

```
A = 1
B = 2
C = 3
D = 4

def average(x, y):
  return (x + y) / 2

average_AB = average(A, B)
average_BC = average(B, C)
average_CD = average(D, C)
```

Now all the logic of calculating the average is contained in the `average` function so we only have one place to check for errors.

i A note on functions

Functions use specific syntax, meaning they have to be written following a set of rules

```
def percentage(number, total):
  intermediate_calculation = number / total
  final_result = intermediate_calculation * 100
  return final_result
```

Line 1

The function is defined with `def` and its name (here `percentage`) and its inputs `(number, total)`.

Line 2

The body of the function is indented by 4 spaces (or 1 tab). The body can consist of as many lines as needed.

Line 4

`return` exits the function and returns the variable that follows - here `final_result`. If a function doesn't have a `return` or just has `return` without any variables it implicitly returns `None`.

Using a function requires "calling" it, this means giving it the inputs it expects - like below

```
result = percentage(1, 2)
```

This code will calculate the function with `number=1` and `total=2` and assign the returned value to `result`. Outside the function the variables `intermediate_calculation` and `final_result` are not available (we would say they are out of scope). There's also nothing special about the names, they are just variables like anywhere else, they could just aswell have been called `nemo_the_fish` or `min_endelige_beregning_af_gennemsnittet_foretaget_mandag_den_13_april` but its a good idea to give variables descriptive but short names.

Only what was returned by the function is available outside the function and if it needs to used later on in the code it needs to be saved to its own variable, so we can think of the code above like

1. `percentage(1, 2)` calculates and returns `final_result`.
2. Whatever is returned is assigned to the variable `result`.

2.1 Exercise: A biological calculation with a function.

Repeat the calculation of cell percentages but now using the `percentage` function defined below

```
# Define a function to calculate the percentage
def percentage(number, total):
  # This function is equivalent to the one shown above, just a little more concise.
  return number / total * 100

# Make variables for the different number of cells
total_cells = 37_000_000_000_000
```

```
brain_cells = 86_000_000_000
liver_cells = 240_000_000_000
skin_cells = 1_600_000_000_000

# Calculate the percentages - use the variables!
brain_pct = _______
liver_pct = _______
skin_pct = _______

# Print the results
print("Brain cell percentage: ", brain_pct)
print("Liver cell percentage: ", liver_pct)
print("Skin cell percentage: ", skin_pct)
```

2.2 Exercise: Fix the (...)

We want to calculate

$$\frac{a+b}{c+d}$$

Someone has implemented the function below to do so, but it is incorrect - your task is to fix it.

```
def fraction_func(a, b, c, d):
    return a + b / c + d

result = fraction_func(152, 16, 2, 2)
print(result)
```

2.3 Implement the (...)

The first term of the quadratic binding equation[1] is

$$\frac{K_D + [P_{tot}] + [L_{tot}]}{2[P_{tot}]} = \frac{a+b+c}{2b}$$

Complete the function below so it calculates this expression

```
def first_term(a, b, c):
    return _______

result = first_term(100, 2, 66)
print(result)
```

[1] An equation that is central to enzyme kinetics, which you will learn more about in later courses.