PHOT 110: Introduction to programming

Practical 2: Input, output, and expressions

Michaël Barbier, Spring semester (2024-2025)

1. Expressions in the Python console

Calculate the values y of following expressions, where $a = 3$, $b = 3/7$, $c = 3.00 \times 10^8$. First assign the parameters a, b, c, and import the functions you need from the `math` package. For the complex exponential function import `exp` from `cmath` instead of `math`. To use the imaginary unit i type `1j`. The results are given in blue so you can compare:

$y = 3a^5 - b^2 \quad \approx 728.8$

$y = \dfrac{iab}{a-b} \quad = 0.5i$

$y = (40/c)^4 \quad \approx 3.16 \times 10^{-28}$

$y = \sqrt{5}/b^2 \quad \approx 12.2$

$y = \sin(b\pi) \times \dfrac{1+b^2}{a^2} \quad \approx 0.13$

$y = \arctan(a) \quad \approx 1.25$

$y = \dfrac{\sinh(1)}{2} \quad \approx 0.59$

$y = \dfrac{e^{i\pi/4}}{\sqrt{2}} \quad = \dfrac{1+i}{2}$

$y = \dfrac{6!}{10!} \quad \approx 1.98 \times 10^{-4}$

2. Input, formatting output, and expressions in Python scripts

2.1 Gloryfier

Prompt a user for his/her name. Print the sentence: “All hail [name of the user]” as a response.

2.2 Repeat words

Prompt the user for a word. Print the word 8 times on one line with spaces in between and put an exclamation mark at the end.

2.3 Braking distance

Prompt a user for the speed of a car. Calculate the braking distance on asfalt and concrete roads and print them. The (kinetic) friction constants for concrete and asphalt are given in following table:

Material	Kinetic	Static
Rubber on concrete (dry)	0.68	0.90
Rubber on concrete (wet)	0.58	-.–
Rubber on asphalt (dry)	0.72	0.68
Rubber on asphalt (wet)	0.53	-.–
Rubber on ice	0.15	0.15

The braking distance D for a car driving at a speed v is determined by the perception-reaction time $t_r \approx 1$ of the driver and the friction coefficient of the road μ according to the following formula ($g = 9.81\, m/s^2$):

$$D = vt_r + \frac{v^2}{2\mu g}$$

Velocity measurement by radar gun

Traffic police officers can use a radar gun to measure the speed of a car. The difference in frequency f_t send out by the radar gun and the received echo f_e gives the Doppler frequency $f_d = f_t - f_e$. This frequency is related to the velocity of the car by:

$$f_d \approx 2v\frac{f_t}{c'}$$

where $c' = c/1.0003$. Suppose the radar device is operated in the K-band: $f_t = 24$ GHz. Calculate the Doppler beating frequency f_d measured by the officer if you pass him with a speed $v = 200$ km/h.

Chevaux-Vapeur car tax calculator

Automatically calculate the tax on cars in France (before 1957). For this, ask a user following properties of his/her car:

- Cylinder capacity of his/her car C (e.g. 1.3 liters)
- Maximum revolutions/minute ω

Then print the amount that the user should pay for his/her car. To calculate the tax use the following formula:

$$CV = CK(60\omega)$$

where K is for a four-cylinder engine: 1.5×10^{-4}.

Formatted table

Prompt the user for two floating point numbers and print a table with on the rows the numbers the number itself, the square, and the square root of the number and as columns the two resulting numbers and their sum. Use formatted strings to set the number of digits after the decimal point.

Hints

Comments: You can comment out a line in a Python script by putting a #-symbol in front. Every code/text behind the #-symbol is ignored by Python. This can help to add inline documentation to your code

```
#| echo: true
# This is a commented line, everything after the #-symbol is ignored

# print(1) this command is ignored
print(2)  # we can also start the comment in the middle of the line
print(3)
```

Importing packages: If you want to use mathematical functions such as the sine: `sin(x)`, square root: `sqrt(x)`, or the number π: `pi`, then you need to import the `math` package. See some examples in following code:

```
# Importing the math library
from math import sin, sqrt, pi, radians

alpha_degrees = 60  # Angle in degrees
alpha = radians(alpha_degrees)  # convert to radians
f = sqrt(2) * pi * sin(alpha)
```

Formatted strings: A regular string can be defined as by surrounding text with single or double quotes, for example `a_string = "Hello"`. A formatted string is created by adding f in front of the first quote. Formatted strings allow to include variables into the string using curly brackets. See following example code:

```
my_string = "Hello, this is some text."

year = 2025
age = 6
my_formatted_string = f"This {year} I will become {age} years old."
```