python_core_language_part1

November 26, 2021

[ ]:

```
# get formating done automatically according to style `black`
#%load_ext lab_black
```

1 Python core language part 1

Python is an *Interpreted, high-level, general-purpose programming language*

- Interpreted: source code executes directly
- High-level: strong abstraction from the details of the computer
- General-purpose: widest variety of applications
- Programming language: language which holds a set of instructions that produce various kinds of output

Content

You will: - Write Section **??** - Understand the Section **??** - Learn about Section **??** … - … and Section **??** - Work with Section **??** - Create and use Section **??** - Dive into Section **??**

1.1 Your first Line of Code

Simply type the follwing in the empty cell below and execute the code. You execute the code by pressing the `Control` + `Enter` key or click the `Run` button above.

```
print("Hello world.")
```

[ ]:

```
# Exercise
```

1.2 Syntax

- Python is cASe SensItiVe.

```
Print("Hello world.")
```

will not work.

- You can comment a line of code with `#`.

```
# Commented lines will not be executed. They are important features for the human reader of th
```

- In Python, blocks of code are expressed by their indentation. This keeps the code clean and tidy.

```
print("one indentation level")
print("this indentation level again")

    print("another indentation level")
```

will lead to `IndentationError: unexpected indent`

- If a line of code becomes to large, you can simply contiue in the following line

```
print("Python is an interpreted, high-level and general-purpose programming language. "
      "Python's design philosophy emphasizes code readability with its notable use of significa
      "indentation. Its language constructs and object-oriented approach aim to help programme
      "write clear, logical code for small and large-scale projects.")
```

1.2.1 Exercise

Play around with different indentation levels and divide a line of code over two lines. Comment your code using #.

[ ]:
```
# Exercise
```

1.3 Variables

A **variable** is a symbolic name, which contains information referred to as a value. A variable is created with an "assignment" equal sign =, with the variable's name on the left and the value it should store on the right.

```
x = 5
```

In the exapmple above, the assignment `x = 5` sets `x` point to `5`.

The Python language reserves a small set of keywords that have a special language functionality. No object can have the same name as a reserved word. Let us see what happens in the following example:

[ ]:
```
class = 5
```

You can see the keyword list any time by typing `help("keywords")`

[ ]:
```
help("keywords")
```

Programmers are not mathematicians:

[ ]:
```
x = 5
y = x
print(y)
```

```
x = 0
print(y)
> ?
```