Python - Úvod

Požiadavky

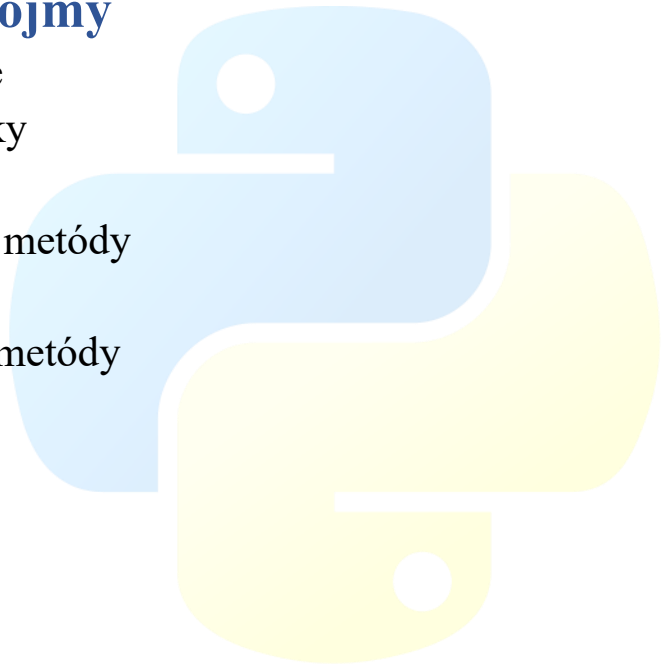
- Nainštalovaný python3 (ideálne >=3.12)
- Textový editor
  - **PyCharm community**
  - VSCode
  - Vim
  - Python IDLE

Ciele

- Oboznámenie sa s fungovaním Pythonu
  - syntax
  - možné využitia
- Písanie scriptov na komunikovanie s webovými aplikáciami a spracovávanie dát

Základné pojmy

- Premenné
- Podmienky
- Cykly
- Funkcie / metódy
- Triedy
- Magické metódy



Premenné

- Netypový jazyk – všetko je jeden veľký typ
- Typy sa dajú označovať – len pre prehľadnosť
- Nálepky, za ktoré sa skrýva dáka hodnota – užitočné ak hodnotu často používame alebo meníme (z matematiky premenná x)

Príklad premenných:

```
example_string: str = "Test String"
example_boolean: bool = True
example_integer: int = 64
example_float: float = 64.32
example_array: list = [1, 2, "element1", "element2"]
```

Podmienky

- Vyhodnocujú sa na základe boolean výrazov

```
if example_integer > 100:
    print(">100")
elif example_integer > 50:
    print(">50")
else:
    print("small")
```

- Logické operátory - and, or, not

	True	False
not	False	True

	True, True	True, False	False, False
and	True	False	False
or	True	True	False

Cykly
1. Pevný počet opakovaní

```
>>> for number in range(10):
...     print(number)
...
0
1
2
3
4
5
6
7
8
9
>>>
```

2. Premenlivý počet opakovaní

2.1. Kontrola na začiatku cyklu

```
number = 0

while number < 10:
    print(number)
    number += 1
```

2.2. Kontrola na konci cyklu

```
number = 0

while True:
    print(number)
    number += 1
    if number >= 10:
        break
```

Metódy

- Často opakovaný kód – zaobalený do metódy – možné volať kedykoľvek
- Parametre – premenné, ktoré vstupujú do metódy
- Po zavolaní môže metóda vracať výsledky svojej operácie – návratová hodnota

```
>>> def divide(number1, number2):
...       if number2 == 0:
...           return 9e99
...       return number1 / number2
...
>>> divide(4, 2)
2.0
>>> divide(4, 0)
9e+99
```

Triedy

- Zaobalenie dátovej štruktúry a jej funkcionality
- Trieda má svoje atribúty a metódy

```
class Person:
    def __init__(self, name):
        self.name = name
```

- Definícia osoby – má svoje meno

Inštancia Triedy

- Chcem spraviť dvoch ľudí, čo sa volajú Ferko a Jozko

```
>>> fero = Person ("Ferko")
>>> jozo = Person ("Jozko")
>>> print(fero.name)
Ferko
>>> print(jozo.name)
Jozko
```

- premenné fero a jozo sú inštancie triedy Person
- pristupovanie k atribútom jednotlivých inštancií je cez .

Magické metódy

- Široký pojem
- Metódy ktoré python používa pri vstavaných operáciách
- Majú svoju defaultnú implementáciu – vieme ich prepísať

- Príklady:
  - __init__ - inštancovanie objektu
  - __str__, __repr__ - stringová reprezentácia objektu
  - __gt__, __lt__, __eq__ - porovnanie dvoch objektov

```python
class Person:
    def __init__(self, name, weight):
        self.name = name
        self.weight = weight

    def __repr__(self):
        return self.name

    def __gt__(self, other):
        return self.weight > other.weight
```

```
>>> fero = Person("Ferko", 80)
>>> jozo = Person ("Jozko", 90)
>>> fero > jozo
False
>>> fero < jozo
True
>>> print(fero, jozo)
Ferko Jozko
>>> classroom = [fero, jozo, fero, jozo]
>>> sorted(classroom)
[Ferko, Ferko, Jozko, Jozko]
>>> sorted(classroom, reverse=True)
[Jozko, Jozko, Ferko, Ferko]
```

Zhrnutie

- Premenné
- Podmienky
- Cykly
- Funkcie / metódy
- Triedy
- Magické metódy