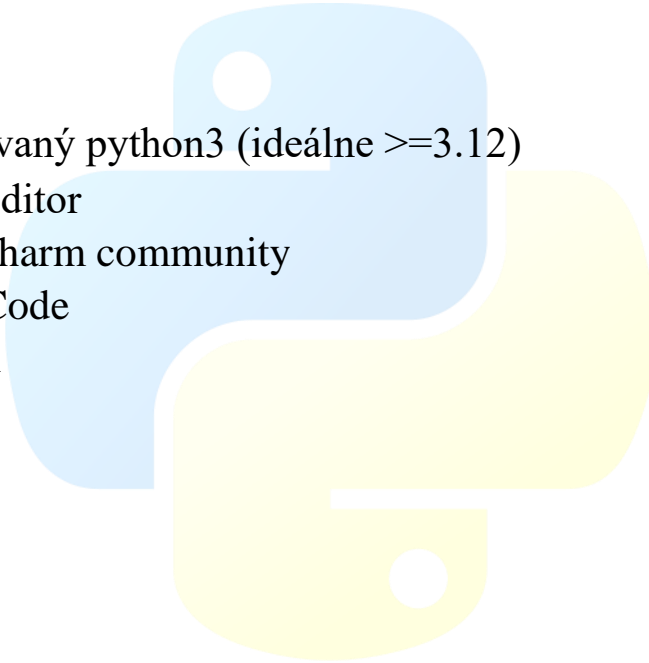




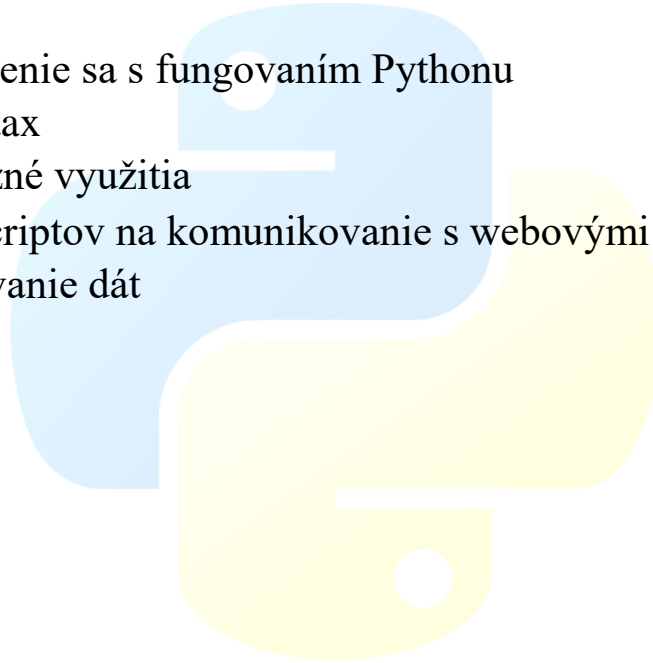
Požiadavky

- Nainštalovaný python3 (ideálne ≥ 3.12)
- Textový editor
 - PyCharm community
 - VSCode
 - Vim



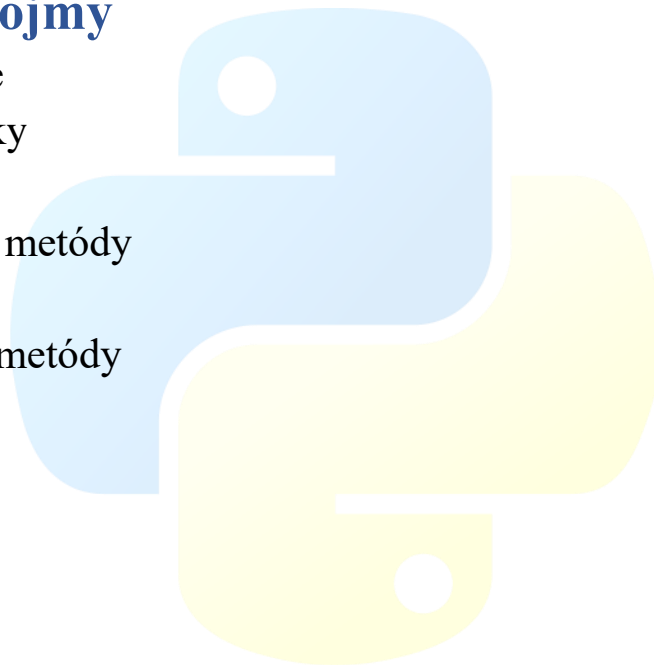
Ciele

- 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
- Dajú sa označovať – nie je to enforcované – len pre prehľadnosť
- Podľa pep8 – názvy sú lower case spojené s _

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 opakování

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

2. Premennivý 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 Animal:  
    def __init__(self, name):  
        self.name = name
```

- Definícia zvierat'a – má svoje meno

Inštancia Triedy

- Chcem spraviť dve zvieratá, čo sa volajú Ferko a Jozko

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

- premenné fero a jozo sú inštancie triedy Animal
- prístupovanie k atribútom jednotlivých inšancií je cez .

Magické metódy

- Široký pojem
- Metódy ktoré python používa pri vstavovaných operáciách
- Majú svoju defaultnú implementáciu – vieme ich prepísať
- Príklady:
 - `__init__` - inšancovanie objektu
 - `__str__`, `__repr__` - stringová reprezentácia objektu
 - `__gt__`, `__lt__`, `__eq__` - porovnanie dvoch objektov

```
class Animal:
    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 = Animal("Fferko", 100)
>>> jazo = Animal("Jozko", 200)
>>> fero > jazo
False
>>> fero < jazo
True
>>> print(fero, jazo)
Fferko Jozko
>>> sorted(zoo)
[Fferko, Fferko, Jozko, Jozko]
>>> sorted(zoo, reverse=True)
[Jozko, Jozko, Fferko, Fferko]
```