

### co budete potřebovat

#### Základ

python - samotný interpret  
docs  
pycharm - IDE  
pypi - veřejně dostupné knihovny

#### Nástroje pro vývojáře

black - formátování kódu  
{popup="https://pypi.org/project/flake8/"}flake8  
- linter

#### Další

jupyter - REPL na webu  
ipython - REPL v konzoli

### looking around

```
help("if")
dir("pa thl ib")
```

### základní datové typy

#### Skalární

- **bool** = True / False  
- **int** = 10, **float** = 10.01  
- **str** = " nějaký text" 'také string' / " " " mul tiline string " " "

#### Kontejnery

- **list** = [ 1, " app le", some\_var ]  
- **dict** = { " key " : " val ", 10: 20 }

### aritmetika a základní operátory

```
a, b = 2, 3
x = (a + b) * ( 10 / 2 )
x == 25 # True
x != 25 # False
x > 25 # False
a_na_d ruhou = a ** 2 # druhá mocnina
b_modulo_2 = b % 2 # zbytek celočí selného dělení
```

### identity check vs ekvivalence

```
x = [ 1, 2 ]
y = [1, 2 ]
x == y # True - stejný obsah
x is y # False - není to stejný objekt
```

Operátor **is** porovnává dva objekty na shodu identity (jde o stejný objekt); operátor **==** porovnává dva objekty na shodu obsahu.

Singletony - jako je `None` - by se měly porovnávat pomocí `is`.

### co je a co není pravda

```
True and False # False
True or False # True
not (True) # False
# short circuit
True or (1 / 0) # True; short circuit nespustí druhý statement
True and (1 / 0) # ZeroDivision Error: division by zero
```

### None (nedefinovaná hodnota)

```
x = None # proměnná x má nedefinovanou hodnotu
x is None # True
x is not None # False
```

Pozor: porovnání `x == None` je "antipattern", nepoužívat.

### list a jeho indexace

**list = []** definuje prázdný list  
**list[0]** = první položka listu již není 10, ale 1

**list[-1]** vrací první položku od konce

**list[i:]** vrací řez listem od pozice `i` do konce

**del list[i]** smaže položku na pozici `i`

### list a jeho metody

**list.append(x)** přidá na konec `x`  
**list.extend([1,2])** zploští `[1,2]` a obě položky přidá na konec  
**list.insert(i,x)** přidá na pozici `i` položku `x`  
**list.pop(i)** odstraní položku na pozici `i` a vrátí ji  
**list.clear()**, **list.copy()** reset nebo copy  
**list.sort()**, seřadí list (**in place!**)  
**list.reverse()**

### dict - indexace

**dict = {}** vytvoř prázdný dict  
**dict[k]** hodnotu v ulož do slovníku na klíč `k`  
**dict[k]** vrať hodnotu z klíče `k`, nebo padni (`KeyError`)  
**del dict[k]** odstraň ze slovníku hodnotu na klíči `k`

### dict - metody

**dict.keys()**; **dict.values()** vrať klíče nebo hodnoty jako iterátor  
**dict.items()** vrať iterátor tuple (`k, v`)  
**dict.get(k)**, **dict.pop(k)** vrať hodnotu z klíče nebo `None`; vrať a odstraň hodnotu z klíče (`rasies`)  
**dict.update(another\_dict)** aktualizuj z jiného slovníku (add or overwrite)  
**dict.clear()** reset, kopie  
**dict.copy()**

