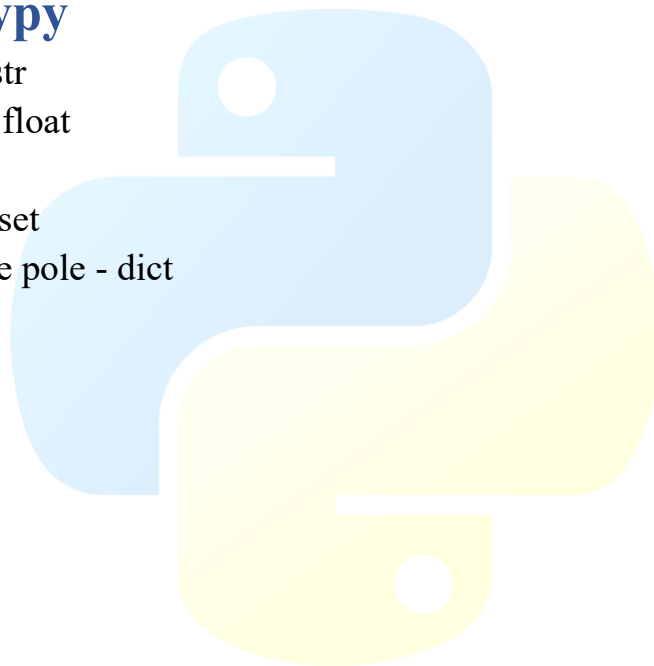


The Python logo, consisting of two interlocking snakes, one light blue and one light yellow, is centered in the background.

# **Python - Základné typy**

## Základné typy

- Reťazec - str
- Číslo - int, float
- Pole – list
- Množina - set
- Asociatívne pole - dict



## Reťazec

- Vieme spájať, indexovať a vyberať podreťazce

```
>>> "CS" + "IRT"
'CSIRT'
>>> retazec: str = "CSIRT"
>>> retazec[0]
'C'
>>> retazec[-1]
'T'
>>> retazec[0:2]
'CS'
>>> retazec[0:2] + retazec[2:]
'CSIRT'
```

- Formátované reťazce

```
>>> retazec: str = "CSIRT"
>>> f"I love {retazec}"
'I love CSIRT'
```

## Užitočné funkcie

- Získanie ascii kódu pre charakter – funkcia ord
- Na základe ascii kódu získať charakter – funkcia chr

```
>>> ord('A')  
65  
>>> chr(65)  
'A'
```

- Nájdenie indexu charakteru – metóda index

```
>>> "CSIRT".index("S")  
1
```

- Nahradenie charakteru iným – metóda replace

```
>>> "CSIRT".replace("S", "s")  
'CsIRT'  
>>> "CSIRT".replace("S", "")  
'CIRT'
```

# Číslo

- Celé čísla - int
- Desatinné čísla - float

```
>>> type(10)
<class 'int'>
>>> type(10.0)
<class 'float'>
>>> type(10.5)
<class 'float'>
```

- Základné matematické operácie - +, -, \*, /, // (Celočíselné delenie)

```
>>> 10 // 3
3
```

- Zaokrúhľovanie – funkcia round

```
>>> round(10.335, 2)
10.34
```

## Pole

- Vieme spájať, indexovať, vyberať podpolia a odstraňovať prvky

```
>>> [1, 2] + [3, 4]
[1, 2, 3, 4]
```

- Indexovanie a podpolia fungujú rovnako ako reťazce

```
>>> pole: list = ["nula", "jeden", "dva", "tri"]
>>> pole[2]
'dva'
>>> pole[2:4]
['dva', 'tri']
```

- Zmena poľa na reťazec – metóda join – prvky poľa musia byť reťazce

```
>>> ", ".join(["nula", "jeden", "dva", "tri"])
'nula, jeden, dva, tri'
```

- Zmena reťazca na pole – metóda split

```
>>> 'nula, jeden, dva, tri'.split(',')  
['nula', 'jeden', 'dva', 'tri']
```

- Pridanie do poľa – metóda append

```
>>> pole: list = [1, 2, 3]  
>>> pole.append(4)  
>>> pole  
[1, 2, 3, 4]
```

- Odstránenie prvku na základe indexu – metóda pop

```
>>> pole: list = [1, 2, 3]  
>>> pole.pop(0)  
1  
>>> pole  
[2, 3]
```

- Odstránenie prvku na základe hodnoty – metóda remove

```
>>> pole: list = [1, 2, 3]
>>> pole.remove(2)
>>> pole
[1, 3]
```

- Usporiadanie poľa – funkcia sorted – parameter reverse

```
>>> pole: list = [1, 2, 3]
>>> sorted(pole)
[1, 2, 3]
>>> sorted(pole, reverse=True)
[3, 2, 1]
```

- Opačné poľa – prvý prvok je posledný atď. – funkcia reversed

```
>>> pole: list = [1, 2, 3]
>>> list(reversed(pole))
[3, 2, 1]
```



# Množina

- Má unikátne prvky

```
>>> mnozina: set = {1, 1, 1, 1, 2, 3}
>>> mnozina
{1, 2, 3}
```

- Vieme spraviť zjednotenie, prienik alebo rozdiel

```
>>> mnozina1: set = {1, 2}
>>> mnozina2: set = {2, 3}
>>> mnozina1.union(mnozina2)
{1, 2, 3}
>>> mnozina1.intersection(mnozina2)
{2}
>>> mnozina1.difference(mnozina2)
{1}
```

- Pridanie prvku do množiny – metóda add

```
>>> mnozina: set = {1, 2, 3}
>>> mnozina.add(4)
>>> mnozina
{1, 2, 3, 4}
```

- Odstránenie je rovnaké ako pri poli – metódy remove a pop
- Vieme zistiť, či je daná množina podmnožinou inej – metóda issubset

```
>>> mnozina1: set = {1, 2}
>>> mnozina2: set = {2}
>>> mnozina2.issubset(mnozina1)
True
```

## Asociatívne pole

- Pole, kde sa môže indexovať aj inými objektami ako celým číslom
- Dvojica – kľúč a hodnota

```
>>> dictionary: dict = {"kluc": "hodnota"}  
>>> dictionary["kluc"]  
'hodnota'
```

- Možné pridávať nové prvky priradením

```
>>> dictionary: dict = {"kluc": "hodnota"}  
>>> dictionary["kluc2"] = "nova hodnota"  
>>> dictionary  
{'kluc': 'hodnota', 'kluc2': 'nova hodnota'}
```

- Získanie hodnoty pre daný kľúč – metóda `get` – vieme dať aj defaultnú hodnotu v prípade, že sa tam daný kľúč nenachádza

```
>>> dictionary: dict = {"kluc": "hodnota"}
>>> dictionary.get("kluc", "neexistuje")
'hodnota'
>>> dictionary.get("kluc2", "neexistuje")
'neexistuje'
```

- Vieme získať všetky kľúče – metóda `keys` – a všetky hodnoty – metóda `values`

```
>>> dictionary: dict = {"kluc1": "hodnota1", "kluc2":
'hodnota2'}
>>> dictionary.keys()
dict_keys(['kluc1', 'kluc2'])
>>> dictionary.values()
dict_values(['hodnota1', 'hodnota2'])
```

## Príklad využitia asociatívneho poľa

- Početnosť jednotlivých písmen v texte

```
>>> veta: str = "Toto je skusobna veta na ktorej budeme ratat pocet  
vyskytov jednotlivych pismen"  
>>> pismena: dict = dict()  
>>> for pismeno in veta:  
...     pismena[pismeno] = pismena.get(pismeno, 0) + 1  
...  
>>> pismena  
{ 'T': 1, 'o': 7, 't': 8, 'j': 3, 'e': 8, 's': 4, 'k': 3, 'u': 2,  
'b': 2, 'n': 4, 'a': 5, 'v': 4, 'r': 2, 'd': 2, 'm': 2, 'p': 2, 'c':  
2, 'y': 3, 'l': 1, 'i': 2, 'h': 1, ' ': 11 }
```

- Usporiadanie podľa početnosti

```
>>> sorted(pismena.keys(), key=lambda x: pismena[x], reverse=True)  
[' ', 'e', 'o', 't', 's', 'n', 'a', 'v', 'j', 'k', 'r', 'y', 'u',  
'b', 'd', 'm', 'p', 'c', 'i', 'T', 'l', 'h']
```

# Python Pomôcka

- V prípade zabudnutia sú tu funkcie dir a help
- dir – vráti všetky atribúty a funkcie daného objektu

```
>>> dir(set())  
['_and_', '__class__', '__class_getitem__', '__contains__',  
 '__delattr__', '__dir__', '__doc__', '__eq__', '__format__',  
 '__ge__', '__getattr__', '__getstate__', '__gt__', '__hash__',  
 '__iand__', '__init__', '__init_subclass__', '__ior__', '__isub__',  
 '__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__',  
 '__new__', '__or__', '__rand__', '__reduce__', '__reduce_ex__',  
 '__repr__', '__ror__', '__rsub__', '__rxor__', '__setattr__',  
 '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__xor__',  
 'add', 'clear', 'copy', 'difference', 'difference_update',  
 'discard', 'intersection', 'intersection_update', 'isdisjoint',  
 'issubset', 'issuperset', 'pop', 'remove', 'symmetric_difference',  
 'symmetric_difference_update', 'union', 'update']
```

- `help` – vrátí docstring, v případě že autori díky napísali

```
>>> help(set().union)
```

```
Help on built-in function union:
```

```
union(*others) method of builtins.set instance
```

```
Return a new set with elements from the set and all others.
```

## Zhrnutie

- Základné typy
  - reťazec, číslo, pole, množina, asociatívne pole
- Pomocné funkcie
  - dir, help

