

# Mutable a immutable objekty

- **Mutable** objekty umožňujú modifikáciu dátového typu po jeho definícii, patria sem: lists, dictionaries, sets, custom objekty

- **Immutable** objekty neumožňujú modifikáciu dátového typu po jeho definícii, patria sem: int, float, string, boolean, tuple

Pozícia v pamäti pre mutable objekty sa nemení, ak ľubovoľná hodnota je zmenená, toto neplatí pre immutable objekty



```
# Immutable int
x = 5
print(id(x))

# 140196051032496 - miesto v pamäti

# Vytvorí nové miesto v pamäti, novú premennú
# nepriradí hodnotu pôvodnej premennej
x = x + 1
print(id(x))

# 140196051032528 - nové miesto v pamäti

# Mutable objekt list
my_list = [1, 2, 3, 4]
print(id(my_list))

# 140196080602240 - miesto v pamäti

# Nevytvorí nové miesto v pamäti, a nový list,
# aktualizuje pôvodný list
my_list[0] = 10
print(id(my_list))

# 140196080602240 - pôvodné miesto v pamäti
```

# Polia NumPy (Numerical Python)



- **NumPy** je knižnica Pythonu používaná na prácu s poľami
- NumPy vytvoril v roku 2005 Travis Oliphant
- Je to open source projekt a môžete ho voľne používať
- NumPy je skratka pre Numerical Python
- V Pythone sú zoznamy, ktoré sa využívajú ako polia, ale ich spracovanie je pomalé
- **NumPy poskytuje** objekt poľa, ktorý je až **50x** rýchlejší ako tradičné zoznamy v Pythone
- Objekt poľa v NumPy sa nazýva *ndarray*, poskytuje množstvo podporných funkcií, vďaka ktorým je práca s *ndarray* veľmi jednoduchá
- Polia sa veľmi často používajú v **dátovej vede**, kde sú rýchlosť a zdroje veľmi dôležité
- NumPy polia sú na rozdiel od zoznamov uložené na jednom súvislom mieste v pamäti, takže procesy k nim môžu pristupovať a manipulovať s nimi veľmi **efektívne**
- Sú optimalizované na prácu s najnovšími architektúrami CPU

# Polia NumPy (Numerical Python)



```
# NumPy knižnica nie je build in Python knižnicou.
# Je nutné ju nainštalovať cez pip install NumPy
# Využíva sa alias np
from re import A, U
import numpy as np

# Vytvorenie ndarray objektu použitím array NumPy funkcie
# Ide o jednodimenzionálne (1-D) pole
np_array = np.array([88, 20, 67, 37, 91, 101])
print(np_array)

# Dvojdimenzionálne (2-D) pole je také pole, ktoré obsahuje
# prvky jednodimenzionálneho poľa
np_array_2 = np.array([[88, 20, 67], [37, 91, 101]])
print(np_array_2)

# Trojdimenzionálne (3-D) pole je také pole, ktoré obsahuje
# prvky dvojdimenzionálneho poľa
np_array_3 = np.array([[[88, 20, 67], [37, 91, 101]], [[7, 8, 9], [10, 11, 12]]])
print(np_array_3)
```

```
# K prvkom poľa sa prístupuje cez index.
# Počiatkový index je 0
# Prístup k prvku v 1-D poli
print(np_array[2])

# Prístup k prvku v 2-D poli
print(np_array_2[1,2])

# Prístup k prvku v 3-D poli
print(np_array_3[1,1,2])

# Zobrazenie posledného prvku v 2-D poli
print(np_array_2[1,-1])

# Iterácia cez jednodimenzionálne pole
np_array = np.array(['Jonathan', 'Brand', 'Patrick', 'Erik'])

for value in np_array:
    print(value)

# Iterácia cez jednodimenzionálne pole s výpisom indexu
for index, value in np.ndenumerate(np_array):
    print(index, value)

# Iterácia cez dvojdimenzionálne pole
np_array = np.array(['Jonathan', 'Brand', 'Patrick', 'Erik'], ['Toews', 'Saad', 'Kane'])

for value in np_array:
    for item in value:
        print(item)
```

# Regulárne výrazy

- **Regulárny výraz** je akási sekvencia znakov, ktorá definuje vzor
- Následne sa na základe vzoru hľadá zhoda v porovnávanom reťazci (vstupe)
- V Pythone pre prácu s regulárnymi výrazmi existuje modul „re“
- Python **re** funkcie:
  - **findall**: vráti zoznam obsahujúce všetky zhody
  - **search**: vráti objekt zhody, ak je zhoda kdekoľvek v reťazci, ináč vráti None
  - **split**: vráti zoznam, kde reťazec bol rozdelený podľa zhody
  - **sub**: nahradí jeden alebo viac zhôd reťazcom
- Príklady:
  - `\d{4}-\d{2}-\d{2}` – hľadá dátum vo formáte YYYY-MM-DD
  - `[a-zA-Z0-9]+\@[a-zA-Z0-9]+\.[a-z]+` – hľadá emailovú adresu vo formáte `name@domain2.domain1`
  - `[0-9]{5}` – hľadá 5 znakov číslíc 0 - 9

- Ďalšie informácie:

- <https://docs.python.org/3/library/re.html>



# Regulárne výrazy

## Metaznaky

Znak	Popis	Príklad
[]	Sada znakov	"[a-g]"
\	Signalizuje špeciálnu postupnosť (možno ju použiť aj na escape zo špeciálnych znakov)	"\d"
.	Ľubovoľný znak (okrem znaku nového riadku)	"he.o"
^	Začína so	"^hello"
\$	Končí so	"world\$"
*	Nula alebo viac výskytov	"aix*"
+	Jeden alebo viac výskytov	"aix+"
{ }	Presne špecifikovaný počet výskytov	"al{2}"
	Buď alebo	"stand sit"

# Regulárne výrazy

Špeciálna sekvencia je „\“ nasledovaná jedným zo znakov v zozname a má význam:

Znak	Popis	Príklad
\A	Vráti zhodu, ak špecifikované znaky sú na začiatku reťazca	"\AThe"
\b	Vráti zhodu, ak špecifikované znaky sú na začiatku alebo na konci reťazca	r"\bain"
	("r" na začiatku zabezpečuje, s reťazcom je zaobchádzané ako so "surovým" reťazcom)	r"ain\b"
\B	Vráti zhodu, špecifikované znaky sú v reťazci, ale NIE na začiatku (alebo na konci) slova	r"\Bain"
	("r" na začiatku zabezpečuje, s reťazcom je zaobchádzané ako so "surovým" reťazcom)	r"ain\B"
\d	Vráti zhodu, kde reťazec obsahuje číslice (čísla od 0 do 9)	"\d"
\D	Vráti zhodu, kde reťazec NEOBSAHUJE číslice	"\D"
\s	Vráti zhodu, kde reťazec obsahuje znak medzery	"\s"
\S	Vráti zhodu, kde reťazec NEMÁ obsahovať medzeru	"\S"
\w	Vráti zhodu, kde reťazec obsahuje akékoľvek slovné znaky (znaky od a do Z, čísla 0-9, a znak podtržník)	"\w"
\W	Vráti zhodu, kde reťazec NEOBSAHUJE žiadne slovné znaky	"\W"
\Z	Vráti zhodu, ak sú zadané znaky na konci reťazca	"Slovakia\Z"

# Regulárne výrazy

**Sada znakov** vo vnútri dvojice hranatých zátvoriek [] so zvláštnym významom:

Sada	Popis
[ats]	Vráti zhodu, kde je prítomný jeden zo zadaných znakov (a, t alebo s)
[a-g]	Vráti zhodu pre akýkoľvek malý znak, abecedne od a po g
[^ats]	Vráti zhodu pre ľubovoľný znak OKREM a, t a s
[0123]	Vráti zhodu, v ktorej sa nachádzajú ľubovoľné znaky zo zadaných číslíc (0, 1, 2 alebo 3)
[0-9]	Vráti zhodu pre ľubovoľné číslo od 0 do 9
[0-2][0-9]	Vráti zhodu pre akékoľvek dvojčífné čísla od 00 do 29
[a-zA-Z]	Vráti zhodu pre ľubovoľný znak v abecednom poradí medzi písmenami a a z, malými písmenami alebo veľkými písmenami
[+]	V množinách nemá znak +, *, ,,  , (), \$, {} žiaden zvláštny význam, [+] znamená: vráti zhodu pre ľubovoľný znak + v reťazci