

## Vypisovanie textu na canvas

Ak chceme na canvas vypísať nejaký text, môžeme použiť príkaz `canvas.create_text(x, y, text='retazec')`. Súradnice  $x$  a  $y$  určujú stred vykresleného textu (čo je občas nešikovné, ale je to tak). 'retazec' zase určuje text, ktorý sa vypíše.

Ak chceme meniť font a veľkosť tohto textu, môžeme tak spraviť cez parameter `font` (pozri príklad nižšie). A vieme dokonca meniť aj farbu pomocou už známeho parametra `fill`.

```
canvas.create_text(100, 100, text='Ahoj')
# kedze nasledujuci text ma stred na suradniciach (0, 0), ista cast z neho bude mimo obrazovku
canvas.create_text(0, 0, text='Tento_text_vytrca_von', font='arial_40')
canvas.create_text(50, 70, text='Modry_text', font='arial_40', fill='blue')
```

## List

Jednou z najzákladnejších súčastí množstva programovacích jazykov je pole, v Pythone nazývané **list**. List je v podstate zoznam hodnôt, ku ktorým vieme pristupovať na základe toho istého mena.

Ak chceme naraz vytvoriť a pracovať s veľa podobnými premennými, môžeme použiť list. Tieto premenné si predstavte naukladané v rade jednu za druhou, pričom každá premenná je označená poradovým číslom. **Pozor:** prvky v liste sú očíslované od 0. To znamená, že premenné v liste so 100 hodnotami, majú poradia 0 až 99.

Listy sa v Pythone označujú hranatými zátvorkami. Tie takisto používame, keď chceme pristúpiť k nejakému číslu v liste. Hodnota `moj_list[4]` nám dá hodnotu na 4 pozícii (teda 5 v rade) listu s názvom `moj_list`.

Naozaj sa jednotlivé hodnoty v liste nelíšia od premenných, iba ich označenie je iné, kompaktnéjšie.

Ak chceme do listu pridať nový prvok, môžeme použiť príkaz `.append(nova_hodnota)`, ktorý vloží zadanú hodnotu na koniec listu.

Najlepšie si to celé ukážeme na príkladoch:

```
>>> pole = [] # vytvorili sme prazdny list
>>> pole.append(4) # pridali sme donho hodnotu 4
>>> pole.append('meno') # tiez tam vieme retazec
>>> print(pole)
[4, 'meno']
>>> nove_pole = [0]*5 # vytvori pole s 5 premennými s hodnotou 0
>>> print(nove_pole)
[0, 0, 0, 0, 0]
>>> nove_pole[3] = 2 # na pozíciu 3 priradíme hodnotu 2
print(nove_pole)
[0, 0, 0, 2, 0]
>>> nove_pole[2] = 4 + nove_pole[3]
>>> nove_pole[0] = nove_pole[0] + 1
>>> print(nove_pole)
[1, 0, 6, 2, 0]
```