



# IB111 Základy programovania

---

## Cvičenie číslo -1

### – práca so súbormi, bitmapová grafika

Matej Troják

(Poďakovanie: Valdemar Švábenský)

1. Práca so súbormi

2. Bitmapová grafika

# Čítanie zo súborov

- **čítanie** zo súboru pomocou funkcie **open**
  - vytvorí objekt typu TextIOWrapper
  - ten má vstavané metódy `read` a `readlines` na prechádzanie súboru

```
input_file = open("input.txt", "r")    # otvorenie suboru pre citanie
content = input_file.read()           # nacitanie celeho obsahu suboru
                                       # do retazca
lines = input_file.readlines()        # nacitanie celeho obsahu ako
                                       # zoznam riadkov
```

- po práci so súborom je nutné ho **zavrieť**

```
input_file.close()                   # uzavretie suboru
```

## Čítanie zo súborov – príklad

- súbor input.txt:

```
Dnes je ten den
ked sa konecne naucime
pracovat so subormi.
```

- prechádzanie súboru pomocou read

```
>>> input_file = open("input.txt", "r")
>>> content = input_file.read()
>>> print(content)
'Dnes je ten den\nked sa konecne naucime\npracovat so subormi.'
```

- prechádzanie súboru pomocou readlines

```
>>> input_file = open("input.txt", "r")
>>> lines = input_file.readlines()
>>> print(lines)
['Dnes je ten den\n', 'ked sa konecne naucime\n', 'pracovat so subormi.']
```

# Zápis do súborov

- **zápis** do súboru pomocou metódy `write`
  - stále používame objekt typu `TextIOWrapper`
- pozor na mód práce so súborom!
  - `"r"` (read) - môžeme iba čítať
  - `"w"` (write) - môžeme zapisovať
  - `"a"` (append) - môžeme súbor rozširovať

```
output_file = open("output.txt", "w") # otvorenie suboru pre zapis
output_file.write("Some sentence.") # zapis retazca do suboru
output_file.writelines(lines) # zapis zoznamu retazcov do suboru
```

- po práci so súborom je nutné ho **zavrieť**

```
some_file.close() # uzavretie suboru
```

## Zápis do súborov – príklad (1)

- **zápis** do súboru pomocou metódy `write`
- na zápis potrebujeme mód `"w"` (write) alebo `"a"` (append)

```
>>> output = open("output.txt", "w")
>>> for _ in range(3):
        output.write("Some sentence.\n")
>>> output.close()
```

- súbor `output.txt`:

```
Some sentence.
Some sentence.
Some sentence.
```

## Zápis do súborov – príklad (2)

- **zápis** zopakujem v móde **"w"** (write)

```
>>> output = open("output.txt", "w")
>>> lines = ["Some sentence.\n", "Some sentence.\n", "Some sentence."]
>>> output.writelines(lines)
>>> output.close()
```

- súbor `output.txt`:

```
Some sentence.
Some sentence.
Some sentence.
```

- súbor `output.txt` je stále rovnaký, pretože mód **"w"** buď vytvorí nový súbor alebo existujúcemu vymaže celý obsah

## Zápis do súborov – príklad (3)

- **zápis** zopakujem v móde "a" (append)

```
>>> output = open("output.txt", "a")
>>> for _ in range(3):
    output.write("Some sentence.\n")
>>> output.close()
```

- súbor `output.txt`:

```
Some sentence.
Some sentence.
Some sentence.
Some sentence.
Some sentence.
Some sentence.
```

- mód "a" zapisuje dáta na koniec súboru



## Práca s reťazcami

- keď máme súbor načítaný, už je to iba práca s reťazcami
- pripomenutie:

```
# odstrani biele znaky zo zaciatku a konca retazca
print(' jablko, hruska, pomaranc \n'.strip())
# mozne dat ako parameter iny znak (napr. \n)
# rstrip - iba koniec retazca, lstrip - iba zaciatok retazca
# rozdeli retazec podla zadaneho oddelovaca a vrati casti v zozname
print('jablko,hruska,pomaranc'.split(','))
# nahradenie retazca
print('jablko,hruska,pomaranc'.replace(',', ' ; '))
# zmena velkosti pismen
print('jABlKo'.lower(), 'jABlKo'.upper())
# specialne znaky se vytvareju pomocou lomitka \
print('toto je \n nový riadok')
print('toto je \t tabulator')
print('toto je apostrof \'', "a toto su uvodzovky \""')
print('takto sa pise lomitko \\'')
```

# Úloha 1

- Napíšte funkciu `most_freq_words(filename, length)`, ktorá pre zadaný súbor vypíše najčastejšie sa vyskytujúce slová dĺžky aspoň `length` v texte súboru `filename`.

```
>>> most_freq_words('sherlock-holmes.txt', 5)
which 4185
would 2101
could 1823
there 1772
Holmes 1244
should 1214
about 982
little 979
before 865
There 815
```

- *Hint:* môžu sa hodiť funkcie `reversed` a `sorted`.

## Úloha 2

- Napíšte funkciu `process_names(input_file, output_file)`, ktorá prečíta všetky mená zo súboru `NAMES.txt` a spracuje ich tak, aby mali veľké iba prvé začiatkové písmeno (napr. `JOHN` -> `John`) a usporiada ich abecedne. Potom ich uloží do iného súboru.

```
>>> process_names("NAMES.txt", "Names.txt")
```

- súbor `"NAMES.txt"`

```
JAMES  
JOHN  
ROBERT  
MICHAEL  
MARY  
WILLIAM  
DAVID  
...
```

- súbor `"Names.txt"`

```
Aaron  
Abbey  
Abbie  
Abby  
Abdul  
Abe  
Abel  
...
```

1. Práca so súbormi

2. Bitmapová grafika

# Pillow

- PIL (Python Imaging Library) je modul pre prácu s bitmapovou grafikou
- Pillow je aktuálny *fork* knižnice PIL
  - <https://pypi.python.org/pypi/Pillow/>
  - Windows: automatický inštalátor  
Pillow-<version>.win-amd64-py3.6.exe
  - Unix: pip install pillow
- Volanie:

```
>>> from PIL import Image
```

## Vybrané funkcie modulu Image

- `Image.new()` - vytvorí nový objekt typu `Image`
  - `Image.new(color_model, size)`
  - `Image.new(color_model, size, background_color)`

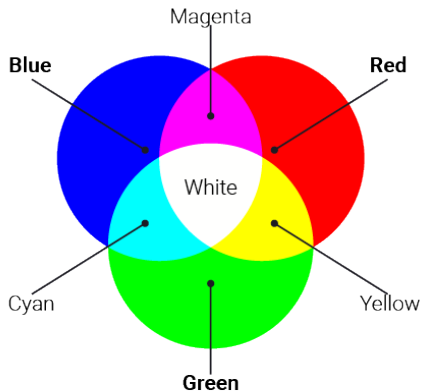
```
>>> img = Image.new("RGB", (100, 100), (255, 255, 255))
```

- `Image.open(filename)` - "otvorí" obrázok
  - Podobne ako pri textových súboroch

```
>>> img = Image.open("photo.jpg")  
>>> img = Image.open("path/pic.jpg")
```

# RGB farebný model

## RGB



# Image – atribúty a metódy

## Atribúty

- Syntax prístupu: `img.attribute`
  - `mode` – farebný mód: "L", "RGB", "RGBA", "CMYK"
  - `size` – dvojica (šírka, výška) v pixeloch

## Metódy

- Syntax volania: `img.function()`
  - `convert()` – konverzia do iného farebného modelu
  - `getpixel()` – získa farbu daného pixela
  - `putpixel()` – zapíše (modifikuje) daný pixel
  - `save()` – uloží obrázok na disk
  - `show()` – zobrazí obrázok v prehliadači



## Metódy objektu Image – príklady (1)

- `convert(color_model)` – konverzia farebného modelu
- `getpixel((x, y))` – získa farbu pixela na súradniciach (x, y) ako trojicu (r, g, b)

```
>>> img = Image.open("photo.jpg")
>>> img = img.convert("RGB")
>>> r, g, b = img.getpixel((50, 100))
>>> print(r, g, b)
255 23 42
```

## Metódy objektu Image – príklady (2)

- `putpixel(position, color)` – zapíše daný pixel
  - `position`: dvojica (`x`, `y`)
  - `color`: trojica (`r`, `g`, `b`)
- `save(filename)` – uloží obrázok na disk pod názvom `filename` do aktuálnej zložky
- `show()` – zobrazí dočasne vygenerovaný obrázok (`.bmp`) v defaultne zvolenom prehliadači obrázkov

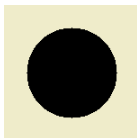
```
>>> img = Image.new("RGB", (20, 20), (255, 255, 255))
>>> img.putpixel((10, 10), (0, 0, 0))
>>> img.show()
>>> img.save("demo.png")
```

## Príklad – kruh

- Kruh so stredom v bode  $(a, b)$  a polomerom  $r$  je množina takých bodov  $(x, y)$ , že platí:  $(x - a)^2 + (y - b)^2 \leq r^2$

```
BLACK = (0, 0, 0)
BG = (239, 236, 202)    # light background color

def disk(size=150, r=50):
    img = Image.new("RGB", (size, size), BG)
    for x in range(size):
        for y in range(size):
            if (x-size//2)**2 + (y-size//2)**2 <= r**2:
                img.putpixel((x, y), BLACK)
    img.save("circle.png")
```

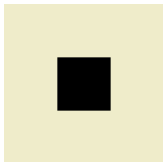


- Pre obrázok o rozmere  $size \times size$  je stred  $(a, b)$  v bode  $(\frac{size}{2}, \frac{size}{2})$

## Úloha 3 – štvorec

- Napíšte funkciu `square(size=150, side=50)`, ktorá vytvorí obrázok o rozmere `size × size` pixelov a nakreslí doňho štvorec so stranou `side` pixelov.

```
>>> square()
```



## Úloha 4 – zmena farieb

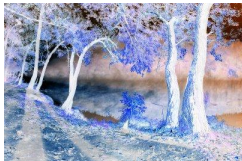
- Napíšte funkciu `modify_colors(filename, modifier, prefix)`, ktorá upraví daný obrázok `filename` podľa danej stratégie `modifier` a výsledný obrázok uloží do nového súboru s prefixom `prefix`.
- Uvažované stratégie:
  - `invert(r, g, b)` – inverzia,
  - `remove_red(r, g, b)` – bez červenej,
  - `remove_green(r, g, b)` – bez zelenej,
  - `remove_blue(r, g, b)` – bez modrej,
  - `grayscale(r, g, b)` – farby šedej.

# Úloha 4 – výsledok

Originál



Inverzia



Bez červenej



Bez zelenej



Bez modrej



Odtiene šedej

