



# Programování 1 (NPRG030)

---

*Cvičení 3: Cykly WHILE a FOR*



# Shrnutí přednášky: Cykly

---

```
1 # Cyklus while
2 while True:           # Podmínka pro ukončení cyklu
3     cislo = int(input("Zadejte kladné číslo nebo 0 pro ukončení: "))
4     if cislo == 0:
5         break         # Předčasné ukončení cyklu
6     if cislo < 0:
7         print("Zadávejte pouze kladná čísla!")
8         continue     # Přeskočení jedné iterace
9     celkovy_soucet += cislo
```

```
10 # Cyklus for
11 for cislo in range(1,101): # Podmínka pro ukončení cyklu
12     if celkovy_soucet > 2000:
13         break         # Předčasné ukončení cyklu
14     if cislo % 3 == 0:
15         continue     # Přeskočení jedné iterace
16     celkovy_soucet += cislo
```

```
19 # Užitečné funkce
20 range(odkud, kam_uz_ne) # Vytvoří sekvenci čísel od `odkud` (včetně) do `kam_uz_ne` (vyjma)
21 range(kam_uz_ne)       # Vytvoří sekvenci čísel od 0 do `kam_uz_ne` (vyjma)
22 range(odkud, kam_uz_ne, o_kolik) # Vytvoří sekvenci čísel s krokem `o_kolik`
```

## Příklad 3.4: Iterace řetězcem a polem

---

- ❖ Vytvořte program v jazyce Python, který:
  - ❖ Vypíše každý prvek ze seznamu [1, 2, 3, 4, 5] zvlášť na samostatný řádek
  - ❖ Vypíše každé písmeno ve slově "Python" zvlášť na samostatný řádek
  - ❖ Sečte všechny prvky v seznamu [1, 2, 3, 4, 5] a výsledek vypíše na výstup
  - ❖ Vypíše všechna sudá čísla, která jsou větší než 1 a menší než 11, a to od největšího po nejmenší
- ❖ Každý dílčí úkol realizujte pomocí některého z cyklů

## Cvičení 3.5: Kreslení trojúhelníku

---

- ❖ Vytvořte *program* v jazyce Python, která *nakreslí textový trojúhelník* ze symbolů `'*'`
  - ❖ Jednotlivé hvězdičky můžete od sebe oddělovat mezerou `' '`
  - ❖ Vstupem programu je výška trojúhelníku
  - ❖ Trojúhelník vypište na standardní výstup
- ❖ Zkuste vytvořit variantu s použitím dvou cyklů i jediného cyklu

## Příklad 3.6: Generátor hesel

---

- ❖ Vytvořte *program* v jazyce Python, který *vygeneruje náhodné heslo* o předem určené délce
  - ❖ Vstupem programu je celočíselný parametr, který určuje délku hesla
  - ❖ Program vykonejte a vygenerované heslo vypište na standardní výstup



### ❖ Cvičení:

- ❖ Sami zkuste vytvořit variantu s použitím for cyklu

# Cvičení 3.7: Odhad $\pi$ metodou Monte Carlo



- ❖ Vytvořte *program* v jazyce Python, který *odhadne hodnotu  $\pi$*  pomocí metody Monte Carlo
  - ❖ Pro výpočet použijte 100 000 vygenerovaných bodů
  - ❖ Výsledek vypište na standardní výstup
- ❖ Náповěda:
  - ❖ Hodnotu  $\pi$  lze odhadnout jako  $4 \cdot \frac{\textit{body\_v\_kruhu}}{\textit{body\_celkem}}$
  - ❖ Souřadnici bodu vygenerujte pomocí funkce `random.uniform(0, 1)` z knihovny `random`
    - ❖ Knihovnu musíte do kódu nejprve importovat - `import random`
  - ❖ Vzdálenost od středového bodu lze určit jako  $x^2 + y^2$ 
    - ❖ Pokud je *vzdálenost* vygenerovaného bodu od středového bodu *menší než 1*, bod se nachází *uvnitř jednotkového kruhu*

# Cvičení 3.8: Caesarova šifra



- ❖ Vytvořte *program* v jazyce Python, který *zašifruje* uživatelem zadaný *text pomocí Caesarovy šifry*
  - ❖ Vstupem programu bude text určený k zašifrování (uvažujte pouze znaky anglické abecedy)
  - ❖ Zašifrovaný text vypište na standardní výstup
- ❖ Náповěda:
  - ❖ Caesarova šifra je typ šifrování, který posune písmena v textu o tři pozice v abecedě dopředu
  - ❖ V anglické abecedě je 26 písmen. Abychom se vyhnuli přetečení, *použijeme operaci modulo* (%), aby výsledek posunu vždy vyšel v intervalu  $\langle 0, 25 \rangle$ 
    - ❖ Např.  $A = 0$  se posune na  $(0 + 3) \% 26 = 3 = D$ , z  $Y = 24$  bude  $(24 + 3) \% 26 = 1 = B$

```
1 zasifrovane_pismeno = chr(((ord(znak) - ord('a') + 3) % 26) + ord('a'))
```

# Jak psát kód: Vestavěné funkce a knihovny

---

- ❖ Využívejte vestavěné funkce a knihovny, než abyste znovu vymýšleli kolo

```
1 # Nevhodný způsob vytvoření listu
2 result = []
3 for item in data:
4     result.append(item)
```

```
1 # Použití vestavěné funkce
2 result = list(data)
```



# Jak psát kód: Méně zanořování

---

- ❖ Omezte hluboké vnořování smyček a podmíněných příkazů pro lepší čitelnost kódu

```
1 # Hluboké zanořování
2 for day in range(TOTAL_DAYS):
3     if condition1:
4         if condition2:
5             # do something
```

```
1 # Méně zanořování
2 for day in range(TOTAL_DAYS):
3     if not condition1:
4         continue
5     if not condition2:
6         continue
7     # do something
```

# Jak psát kód: Používejte enumerate

---

- ❖ Pro iteraci s indexy použijte funkci `enumerate()`

```
1 # Nevhodný způsob iterace
2 animals = ["dog", "cat", "fish", "horse"]
3 for index in range(len(animals)):
4     print(f"Index: {index}, Value: {animals[index]}")
```

```
1 # Lepší způsob
2 animals = ["dog", "cat", "fish", "horse"]
3 for index, value in enumerate(animals):
4     print(f"Index: {index}, Value: {value}")
```



# Reference

---

## While

- ❖ [https://www.w3schools.com/python/python\\_while\\_loops.asp](https://www.w3schools.com/python/python_while_loops.asp)

## For

- ❖ [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp)

## Iterace listem

- ❖ [https://www.w3schools.com/python/python\\_lists\\_loop.asp](https://www.w3schools.com/python/python_lists_loop.asp)