

Cykly

October 20, 2023

1 Cykly

1.1 For cyklus

for i in range(0, 5):
 prikazy

Poznámky

- Při každém průchodu cyklem se zvětší index i o 1
- Index i v našem případě nabývá hodnot 0, 1, 2, 3, 4
- Odsazení je nutné pro správný běh kódu

```
[1]: for i in range(0, 5):  
      print(i)
```

0
1
2
3
4

Jaké hodnoty dostaneme s použitím **range**:

- range(x, y, n) - generuje celá čísla v rozmezí $[x, y)$, n je krok (nepovinný parametr)
- range(1, 5) : 1, 2, 3, 4
- range(5) : 0, 1, 2, 3, 4 (pokud je zadán jenom jeden parametr, spodní mez je 0)
- range(4, 10, 2) : 4, 6, 8 (krok je 2)
- range(5, 0, -1) : 5, 4, 3, 2, 1 (krok je -1, čísla se zmenšují)

```
[2]: list(range(1,5))      # Pro vypis hodnot generovanych range je nutna konverze  
    ↪ na typ list
```

```
[2]: [1, 2, 3, 4]
```

Úkol Násobilka: Uživatel zadá, násobky jakého čísla (n) chce vypsát. Program vypíše násobky do $1.n$ do $10.n$ (např. pro číslo 2 bude výstup 2, 4, 6, 8, 10, 12, 14, 16, 18, 20).

```
[ ]:
```

Úkol Vypište všechna sudá čísla až do hodnoty zadané uživatelem.

[]:

1.2 While cyklus

while podminka:

prikazy

Poznámky

- Cyklus probíhá, dokud je splněna *podminka*
- Musíme ohlídat, aby nešlo o nekonečný cyklus

Příkaz while-else else definuje větev, co se provede po skončení cyklu while (bez přerušení)

while podminka:

prikazy

else:

prikazKonec

Příklad Výpis čísel od 0 do 4 (včetně) s využitím while-cyklu

```
[3]: cislo = 0
while cislo < 5:
    print(cislo)
    cislo += 1          # Zkraceny zapis pro cislo = cislo + 1, plati i pro ↵
↵jine operatory
```

0
1
2
3
4

```
[4]: # Ukazka pouziti while-else
cislo = 0
while cislo < 5:
    print(cislo)
    cislo += 1
else:
    print("Cyklus proběhl bez problémů.")
```

0
1
2
3
4
Cyklus proběhl bez problémů.

Příklad

Ze vstupu načtete posloupnost čísel. Čísla budou zadána každé na samostatné řádce, vstup bude ukončen číslem -1 (ta už do posloupnosti nepatří). Dopředu nevíte, kolik čísel bude. Čísla průběžně

vypisujte.

```
[5]: cislo = int(input("Zadejte číslo: "))
while cislo != -1:
    print(cislo)
    cislo = int(input("Zadejte číslo: "))
```

```
Zadejte číslo: 5
5
Zadejte číslo: 3
3
Zadejte číslo: -1
```

1.3 Užitečné příkazy

continue - přeskočí na další průchod cyklem

break - ukončení cyklu (poznámka: s použitím opatrně)

Příklad (continue) Uživatel zadá číslo. Vypište čísla od 1 do 10 bez čísla zadaného uživatelem.

```
[ ]: cislo = int(input("Zadejte číslo "))
for i in range(1,11):
    if i == cislo:
        continue
    print(i)
```

Příklad (break) Uživatel zadává čísla, vstup ukončí -1. Požadujeme, aby zadaná čísla byla menší než 100. Program čísla sečte (poznámka: čísla nebudeme ukládat, ale rovnou přičítat). Pokud uživatel zadá číslo rovno nebo větší než 100, z cyklu vyskočíme pomocí break.

```
[ ]: cislo = int(input())
suma = 0
while cislo != -1:      # != znamená není rovno
    if cislo >= 100:
        break
    suma += cislo
    cislo = int(input())
print("Součet čísel je ", suma)
```

Příklad (break) Načítání řady čísel ukončených -1. Čísla postupně vypisujeme.

```
[1]: while True:          # Nekonečna smyčka
    cislo = int(input("Zadejte číslo: "))
    if cislo == -1:       # Ukončení smyčky
        break
    print(cislo)
```

```
Zadejte číslo: 4
4
```

Zadejte číslo: 5

5

Zadejte číslo: -1

Úkol Uživatel zadává celá čísla. Vypisujte zadaná čísla, dokud jsou čísla menší než 80. Po načtení prvního čísla, které je větší nebo rovno 80, ukončete načítání a upozorněte uživatele, že výpis končí.

[]:

Úkoly

Ze vstupu načtete posloupnost přirozených čísel. Čísla budou zadána každé na samostatné řádce, vstup bude ukončen číslem -1 (ta už do posloupnosti nepatří). Při řešení následujících úloh nepotřebujete ukládat celou číselnou řadu.

- Vynásobte všechna zadaná čísla
- Najděte nejvyšší číslo v řadě čísel (náповěda: uložte si první číslo jako kandidáta na maximum, porovnávejte aktuálně načtené číslo s kandidátem na maximum - pokud je vyšší, pak máte si uložit tuto hodnotu jako nového kandidáta, pokračujte obdobně až na konec)
- Najděte druhé nejvyšší číslo, vypište jeho pozici v řadě

[]: