6. Programování - cyklus for

obecná charakteristika cyklu (dva typy)

- používají se k opakování bloku příkazů (v Pythonu oddělený tabulátorem / 2/4 mezerami) několikrát po sobě
- buď na základě pevně daného počtu opakování nebo podmínky → 2 hlavní typy: **for** a **while**

cyklus for - syntaxe a popis, operátor in, řídící proměnná

- jedná se o cyklus s řídící proměnnou (nebo i více)
- dopředu je znám počet opakování
- nemůže nastat nekonečný cyklus
- v Pythonu slouží pro procházení kolekcí (řetězců, seznamů, slovníků…), chová se vlastně jako cyklus **foreach** u jiných jazyků

for proměnná in sekvence:

tělo cyklu

Princip

- řídící proměnná postupně nabývá všech hodnot z pevně dané množiny prvků
- pro každý prvek se vykoná jedna iterace cyklu (příkazy v jeho tělu)

V cyklu for můžeme použít i vícenásobné přiřazení proměnných, pokud kolekce, kterou procházíme, obsahuje další kolekci:

souradnice = [(1,2), (14,25), (128,574), (7895,6548)]

for x,y in souradnice:

print(x,y)

Tohoto se typicky využívá ve spojení s funkcí enumerate tehdy, kdy zároveň s prvkem potřebujeme také jeho index. Fce **enumerate(s)** poskytuje posloupnost n-tic o dvou hodnotách (index, prvek) ze sekvence s:

for index, prvek in enumerate(seznam):

print(index, prvek)

funkce range()

V Pythonu používáme for s fcí **range()** pro stejné chování jako v jiných programovacích jazycích (iterace množinou celých čísel s řídící proměnnou).

- V příkazu range() generuje program čísla od počáteční hodnoty **včetně** až po koncovou hodnotu **vyjma**, po kroku k.
- Povinný parametr je pouze koncová hodnota - **výchozí od=0 a k=1**.
- Fce range() pracuje pouze s celými čísly. Lze použít i záporná.

for proměnná in range(od, do, k):

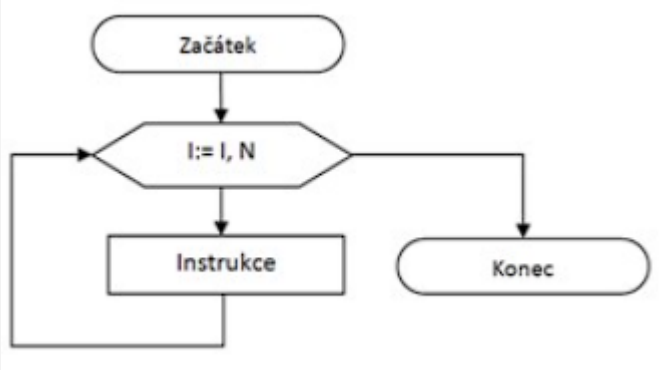
tělo cyklu

využití

Používáme, pokud víme, kolikrát má cyklus proběhnout (jsme řízeni buď seznamem hodnot, nebo počtem čísel).

Cyklus for se často používá pro procházení prvků posloupnosti a jejich zpracování.

vývojový diagram



vnořený cyklus

- = cyklus, který se nachází uvnitř jiného cyklu
- můžeme vnořovat libovolné množství cyklů a používat řídící proměnné vnějších
- řídící proměnné se stejným názvem se přepisují
- využíván k iteraci vícerozměrných dat a polí

Př. vnoření cyklu pro vytvoření obdélníku 4x2:

```
sirka, vyska = 4, 2
for y in range(vyska):
    for x in range(sirka):
        print("* ",end="")
    print()
```

příkazy break a continue

break

- okamžitě ukončí vykonávání cyklu
- příkazy uvedené v tělu za break se už neprovádí
- program pokračuje za cyklem

```
while True:
    heslo=input("Zadej vánoční kód:")
    if heslo=="jedle":
        break
    print("Špatný kód, zadej znovu.")
print("Spávný kód! Veselé Vánoce!")
```

continue

- způsobí ukončení aktuální iterace (opakování)
- zbytek bloku cyklu za continue se neprovede
- program pokračuje následující iterací (while kontroluje podmínku, for přechází na další prvek)

Cyklus for i while mají nepovinnou sekci **else**. Kód v sekci else se provádí po korektním ukončení cyklu. Pokud byl cyklus ukončen pomocí break, nebo return, nebo pokud nastala výjimka, kód v else se neprovede.

náhodná čísla

V programování se používají pro náhodné rozhodování nebo náhodný výběr prvků ze seznamu. Chceme-li pracovat s náhodným výběrem nebo čísly, potřebujeme knihovnu **random** (musíme nejdříve importovat).

- **randint**(od, do) - vygeneruje náhodné číslo z intervalu <od, do> (od a do jsou celá čísla)
- **randrange**(0, 100, 2) - generuje náhodné celé číslo od 0 do 99, sudé
- **choice**(data) - vybere z dat (obvykle řetězec nebo seznam) náhodný prvek
- **sample**(data, k) - vybere z dat (obvykle řetězec nebo seznam) k náhodných prvků
- **random**() - generuje náhodné reálné číslo od 0 do 1
- **uniform**(x, y) - generuje náhodné reálné číslo x<=c<y