**Cykly**

V Pythone sa používajú 2 typy cyklov: while a for.

V prípade cyklu while sa cyklus opakuje, pokiaľ je platná podmienka

i = 1  
while i < 6:  
  print(i)  
  i += 1

Výsledok:

1  
2  
3  
4  
5

Pri cykle while je potrebná tzv. indexujúca premenná, tu je to i a nastavená na hodnotu 1.

## Príkaz break

Pomocou príkazu break môžeme cyklus zastaviť, aj keď podmienka stále platí:

### Príklad

Chceme cyklus ukončiť, keď i je 3:

i = 1  
while i < 6:  
  print(i)  
  if i == 3:  
    break  
  i += 1

Výsledok

1  
2  
3

Cyklus for sa používa na iterácie cez sekvencie a menej spôsobom ako v iných programovacích jazykoch. Pomocou príkazu for môžeme vykonať sadu príkazov pre každú položku v zozname.

### Príklad

Vytlačiť názov ovocia zo zoznamu:

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)

Výsledok

apple  
banana  
cherry

Pomocou príkazu break je možné zastaviť cyklus predtým, než prejde cez všetky položky:

### Príklad

Ukončenie cyklu, keď x je "banana":

fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
  print(x)   
  if x == "banana":  
    break

Výsledok:

apple  
banana

## Funkcia range()

Funkcia range() umožňuje vykonať cyklus podľa určeného počtu opakovaní, predvolene od 0, s krokom 1 a končí určeným čislom. Znamená to, že range(6) má hodnoty od 0 do 5.

### Príklad

for x in range(6):  
  print(x)

Výsledok:

0  
1  
2  
3  
4  
5

Krok 3 (predvolené je 1):

for x in range(2, 30, 3):  
  print(x)

Výsledok:

2  
5  
8  
11  
14  
17  
20  
23  
26  
29

## Príkaz Else v cykle For

Príkaz else v cykle loop určuje blok príkazov, ktoré sa vykonajú po skončení cyklu:

### Príklad

Vytlačiť všetky čísla od 0 do 5 a vytlačiť oznam,keď sa cyklus skončí:

for x in range(6):  
  print(x)  
else:  
  print("Finally finished!")

Výsledok:

0  
1  
2  
3  
4  
5  
Finally finished!

## Vnorené cykly

Vnorené cykly sú cykly vo vnútri cyklov.

Vnútorný cyklus sa vykoná pri každej iterácii vonkajšieho cyklu:

### Príklad

Vytlačiť prídavné meno pre každé ovocie:

adj = ["red", "big", "tasty"]  
fruits = ["apple", "banana", "cherry"]  
  
for x in adj:  
  for y in fruits:  
    print(x, y)

red apple  
red banana  
red cherry  
big apple  
big banana  
big cherry  
tasty apple  
tasty banana  
tasty cherry

## Polia

Python nemá priamu podporu pre polia, ale používajú sa zoznamy. Je možné používať aj polia, ale je nutné použiť knižnicu NumPy.

Polia sa používajú na uloženie viacerých hodnôt do jednej premennej:

Vytvorenie poľa s názvami áut:

cars = ["Ford", "Volvo", "BMW"]

print(cars)

Výsledok:

['Ford', 'Volvo', 'BMW']

Podobne môžeme mať aj 300 názvov. K jednotlivým hodnotám sa môžeme dostať cez index/indexové číslo.

### Príklad

K hodnote 1. položky sa dostaneme takto:

### cars = ["Ford", "Volvo", "BMW"]

### x = cars[0]

### print(x)

### Výsledok:

### Ford

Hodnotu 1. položky možeme zmeniť takto:

cars[0] = "Toyota"

Pomocou funkcie append() môžeme pridať ďalšiu položku:

cars = ["Ford", "Volvo", "BMW"]

cars.append("Honda")

print(cars)

V:

['Ford', 'Volvo', 'BMW', 'Honda']

Python má viacero zabudovaných funkcií, ktoré sa dajú používať v súvislosti so zoznamami či poliami.

|  |  |
| --- | --- |
| **Funkcia** | **Popis** |
| append() | Pridá položku na koniec zoznamu |
| clear() | Odstráni všetky položky v zozname |
| copy() | Vráti kópiu zoznamu |
| count() | Vráti počet položiek s uvedenou hodnotou |
| extend() | Pridá položky do aktuálneho zoznamu na jeho koniec |
| index() | Vráti index prvej položky s určenou hodnotou |
| insert() | Pridá položku do určenej pozície |
| pop() | Odstráni položku v určenej pozícii |
| remove() | Odstráni prvú položkz s určenou hodnotou |
| reverse() | Prevráti poradie zoznamu |
| sort() | Usporiada zoznam |

**Funkcie:**

Funkcia je blok kódu, ktorý sa vykoná, keď sa funkcia zavolá.

Funkcii môžeme predať dáta, tzv. parametre.

Funkcia môže aj vrátiť dáta ako výsledok jej spustenia.

**Vytvorenie funkcie**

Funkcia sa vytvára pomocou kľúčového slova def:

def my\_function():  
  print("Hello from a function")

## Volanie funkcie

Funkcia sa volá uvedením jej mena aj so zátvorkami:

def my\_function():  
  print("Hello from a function")  
**my\_function()**

## Argumenty funkcie

Informácie, ktoré predáme funkcii sa nazývajú argumenty.

Argumenty sa uvádzajú vo vnútri zátvoriek, viacero arg. sa oddeľuje čiarkou. Pri volaní funkcie musíme použiť presne uvedený počet argumentov.

Nasledujúca funkcia má 1 argument (fname). Pri jej zavolaní jej predáme reťazec (meno):

def my\_function(**fname**):  
  print(fname + " Refsnes")  
  
my\_function(**"Emil"**)  
my\_function(**"Tobias"**)  
my\_function(**"Linus"**)

V:

Emil Refsnes  
Tobias Refsnes  
Linus Refsnes

**Príklad**

Táto funkcia má 2 argumenty a aj ju voláme s 2 argumentami:

def my\_function(fname, lname):  
  print(fname + " " + lname)  
  
my\_function("Emil", "Refsnes")

Chybné volanie funkcie (chýba 1 argument):

def my\_function(fname, lname):  
  print(fname + " " + lname)  
  
my\_function("Emil")

## Práca so súbormi

Funkcia open(). Má 2 argumenty: filename a mode.

Sú 4 spôsoby otvorenia súboru:

"r" - Read - Default hodnota. Otvára súbor na čítanie

"a" - Append – Otvára súbor na pripojenie, ak súbor neexistuje, vytvorí ho

"w" - Write - Otvára súbor na zápis, ak súbor neexistuje, vytvorí ho

"x" - Create – Vytvára konkrétny súbor, ak už existuje vráti hlásenie chyby

Tiež je možné špecifikovať, či sa so súborom pracuje v textovom alebo binárnom režime

"t" - Text - Default hodnota. Textový režim

"b" - Binary – Binárny režim (napr. obrázky)

**Príklady**

Na otvorenie stačí uviesť toto:

f = open("demofile.txt")

Je to ekvivalent tohto príkazu:

f = open("demofile.txt", "rt")

(r,t sú defaultné hodnoty)

demofile.txt

Hello! Welcome to demofile.txt  
This file is for testing purposes.  
Good Luck!

f = open("D:\\myfiles\welcome.txt", "r")  
print(f.read())

f.close()

Otvorenie súboru "demofile2.txt" a pripojenie jeho obsahu k súboru:

f = open("demofile2.txt", "a")  
f.write("Now the file has more content!")  
f.close()  
  
#open and read the file after the appending:  
f = open("demofile2.txt", "r")  
print(f.read())

Otvorenie a prepísanie obsahu súboru "demofile3.txt":

f = open("demofile3.txt", "w")  
f.write("Woops! I have deleted the content!")  
f.close()  
  
#open and read the file after the appending:  
f = open("demofile3.txt", "r")  
print(f.read())

**Vytvorenie súboru**

s názvom "myfile.txt":

f = open("myfile.txt", "x")

(vytvorí sa prázdny súbor)

**Vytvorenie súboru, ak ešte neexistuje**

f = open("myfile.txt", "w")