Cyklus for, range(), moduly math a random

1 Cyklus for

Cyklus `for` opakuje odsadený blok pre každú hodnotu z iterovateľného objektu. Iterovateľné sú napríklad `range()`, reťazec, n-tica, zoznam, množina, slovník a otvorený súbor.

Zápis má tvar `for premenná in postupnosť:` Dvojbodka ukončuje hlavičku. Telo cyklu sa odsádza štyrmi medzerami. Neodsadený príkaz už do cyklu nepatrí. Premenná cyklu postupne odkazuje na jednotlivé hodnoty. Názov premennej možno zvoliť ľubovoľne; má opisovať spracovanú hodnotu. Ak je postupnosť prázdna, telo sa nevykoná ani raz.

```
for i in range(4):
    print(i, "riadok")
```

Príklad vypíše riadky s hodnotami `0`, `1`, `2` a `3`. Po skončení cyklu zostáva meno `i` naviazané na poslednú hodnotu.

Za `in` môže byť priamo n-tica hodnôt. Hodnoty sa môžu opakovať a nemusia mať rovnaký typ.

```
for i in 1, 2, 3, 4, 5, 6:
    print(i, end=" ")
for i in 1, 1, 2, 4, 4, 7:
    print(i, end=" ")
for hodnota in 3.14, "text", 2:
    print(hodnota, type(hodnota))
```

2 Prechádzanie reťazca a zhromažďovanie výsledku

Pri prechádzaní reťazca dostáva premenná cyklu postupne jednotlivé znaky. Reťazec `Python` poskytne znaky `P`, `y`, `t`, `h`, `o`, `n` v tomto poradí.

```
for znak in "Python":
    print(znak)
```

Výsledok sa často zhromažďuje v akumulátore. Akumulátor sa vytvorí pred cyklom, v cykle sa aktualizuje a po cykle sa použije. Počítadlo začína od `0`, súčet od `0`, súčin od `1` a skladaný reťazec od prázdneho reťazca.

```
vstup = input("Zadaj vstup: ")
pocet = 0
retazec = ""
for znak in vstup:
    pocet += 1
    retazec = znak + retazec
print("Počet znakov =", pocet)
print("Obrátený reťazec =", retazec)
```

Príkaz `retazec = znak + retazec` vkladá nový znak na začiatok, preto vytvorí obrátený text. Počet znakov možno získať aj pomocou `len(vstup)` a obrátený text pomocou `vstup[::-1]`.

```
pocet = 0
for mesiac in 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31:
    pocet_dni += mesiac
print("Počet dní v roku =", pocet)
```

3 Funkcia range()

Funkcia `range()` vytvára aritmetickú postupnosť celých čísel. Hodnoty negeneruje všetky vopred, preto je pamäťovo úsporná aj pri veľkom rozsahu. Funkcia `range(stop)` začína nulou. Funkcia `range(start,stop)` používa krok `1` a `range(start,stop,step)` používa zadaný krok. Hodnota `stop` sa nikdy nezahrnie. Krok nesmie byť `0`.

```
print(list(range(4)))             # [0, 1, 2, 3]
print(list(range(1, 7)))          # [1, 2, 3, 4, 5, 6]
print(list(range(2, 11, 2)))      # [2, 4, 6, 8, 10]
print(list(range(10, 0, -1)))     # [10, 9, ..., 1]
print(list(reversed(range(1, 11))))
```

Kladný krok vyžaduje `start < stop`; záporný krok vyžaduje `start > stop`. Inak je výsledok prázdny. Funkcia `range()` prijíma celé čísla, nie hodnoty typu `float`.

```
for i in range(100, 200):
    print(i, end=" ")
```

Príklad vypíše celé čísla od `100` po `199`. Parameter `end=` nahradí koniec riadka medzerou.

4 Počítanie, súčet a súčin

```
n = int(input("Zadaj n: "))
pocet = 0
for i in range(n):
    pocet += 1
print("Prešiel som cyklus", pocet, "krát")
```

Pre nezáporné `n` sa telo vykoná `n`-krát. Pri zápornom `n` je `range(n)` prázdny. Premenná `i` nie je v tele potrebná; bežné meno pre nepoužitú hodnotu je _ (podkčiarkovník).

```
n = int(input("Zadaj n: "))
sucet = 0
for i in range(n):
    sucet += i
print("Súčet =", sucet)
sucet_stvorcov = 0
for i in range(n):
    sucet_stvorcov += i * i
print("Súčet štvorcov =", sucet_stvorcov)
```

Prvý cyklus sčíta čísla od `0` po `n - 1`. Druhý sčíta ich druhé mocniny. Pre jednoduchý súčet možno použiť aj `sum(range(n))`.

```
sucin = 1
for cislo in range(2, 7):
    sucin *= cislo
print("Faktoriál 6 =", sucin)

n = int(input("Zadaj číslo: "))
```

```
faktorial = 1
for cislo in range(2, n + 1):
    faktorial *= cislo
print(n, "faktoriál =", faktorial)
```

Faktoriál nezáporného čísla `n` je súčin $1 \cdot 2 \cdot \ldots \cdot n$. Platí `0! = 1`, preto počiatočná hodnota súčinu musí byť `1`.

5 Vnorené cykly

Cyklus môže obsahovať ďalší cyklus. Pre každú hodnotu vonkajšieho cyklu sa vnútorný cyklus vykoná celý. Celkový počet opakovaní sa pri pravidelných rozsahoch násobí.

```
for i in range(0, 100, 10):
    for j in range(10):
        print(i + j, end=" ")
    print()
```

Vonkajší cyklus vytvára desiatky `0, 10, ..., 90`; vnútorný pridáva jednotky `0` až `9`. Výsledkom je tabuľka čísel `0` až `99` v desiatich riadkoch.

```
pocet = int(input("Zadaj počet riadkov: "))
for riadok in range(1, pocet + 1):
    for cislo in range(1, riadok + 1):
        print(cislo, end=" ")
    print()
```

V riadku číslo `riadok` vnútorný cyklus vypíše čísla od `1` po `riadok`. Odsadenie určuje, že prázdny `print()` patrí do vonkajšieho cyklu, ale už nie do vnútorného.

6 Moduly a príkaz import

Modul je súbor s príkazmi, funkciami, triedami a konštantami. `import math` načíta modul a vytvorí meno `math`. Člen modulu sa zapisuje bodkovou notáciou, napríklad `math.sqrt(2)` alebo `math.pi`.

`from math import sin, pi` vloží vybrané mená priamo do aktuálneho priestoru mien.

`dir(math)` zobrazí mená dostupné v module. `help(math)` zobrazí dokumentáciu modulu a `help(math.log)` dokumentáciu konkrétnej funkcie. Modul treba pred použitím importovať.

```
import math
print(math.pi, math.e, math.tau)
print(math.sqrt(2), math.hypot(3, 4))
dir(math)
help(math.log)
```

7 Modul math

`math` obsahuje matematické konštanty a funkcie pre reálne čísla. Často sa používajú `sqrt()`, `factorial()`, `gcd()`, `floor()`, `ceil()`, `log()`, `exp()`, `sin()`, `cos()` a podobne.

Goniometrické funkcie očakávajú uhol v radiánoch. `math.radians()` prevádza stupne na radiány a `math.degrees()` radiány na stupne. Hodnota `math.sin(math.radians(45))` sa rovná `math.sin(45 * math.pi / 180)`.

```
import math
for uhol in range(0, 91, 5):
    uhol_v_radianoch = math.radians(uhol)
    sinus = math.sin(uhol_v_radianoch)
    print(f"{uhol:3d} {sinus:9.6f}")
```

Formát `:3d` vyhradí celému číslu tri znaky. Formát `:9.6f` vyhradí deväť znakov a zobrazí šesť číslic za desatinnou bodkou.

```
import math
for uhol in range(0, 361, 10):
    uhol_v_radianoch = math.radians(uhol)
    sinus = math.sin(uhol_v_radianoch)
    stlpec = round(sinus * 39 + 39)
    print(" " * stlpec + "*")
```

Hodnota sínusu z intervalu `[-1, 1]` sa násobením `39` a posunutím o `39` prevedie približne na stĺpce `0` až `78`. Hviezdičky vytvoria textový graf sínusoidy.

8 Modul random

Modul `random` generuje pseudonáhodné hodnoty. `random.random()` vráti `float` z intervalu `[0.0, 1.0)`. Funkcia `random.randrange(start, stop, step)` vyberá z hodnôt zodpovedajúcich `range()`. Funkcia `random.choice(postupnosť)` vyberie jeden prvok. `random.choices()` vyberá s opakovaním, `random.sample()` bez opakovania a `random.shuffle()` premieša zoznam na mieste. `random.uniform(a, b)` vracia reálne číslo z `[a, b)`.

```
import random
for i in range(100):
    rnd = random.randrange(1, 7)
    print(rnd, end=" ")
```

Príklad simuluje sto hodov bežnou kockou.

```
import random
spoluhlasky = "bcčdďfghjklľímnňprŕsštťvzž"
samohlasky = "aáäeéiíoóôuúyý"
slovo = ""
for i in range(4):
    slovo += random.choice(spoluhlasky)
    slovo += random.choice(samohlasky)
print(slovo)
```

Príklad vytvorí osempísmenový reťazec striedaním náhodnej spoluhlásky a samohlásky. Generátor je deterministický. Funkcia `random.seed(123)` nastaví počiatočný stav a umožní zopakovať rovnakú postupnosť, čo je užitočné pri testovaní.