PYTHON

19. Prechádzanie znakmi reťazca

Vypracovala: Ing. Eva Gabonayová
Predmet: Informatika
Vzdelávacia oblasť: Matematika a práca s informáciami
Dátum spracovania: 9. 10. 2018

Použitie for cyklu

- na postupné prechádzanie reťazcom po jednotlivých znakoch

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

```
P
y
t
h
o
n
>>>
```

```
retazec = 'Python'
for znak in retazec:
    print(znak)
```

Čo bude robiť nasledujúci program? Čo vypíše na obrazovku?

```
retazec = 'Python,
poradie = 0
for znak in retazec:
    poradie += 1
    print(znak)
print(poradie)
```

Program postupne vypíše všetky písmená reťazca na samostatný riadok a spočíta počet znakov. Na konci vypíše celkový počet znakov reťazca.

Funkcia len(*reťazec*)

- zistí dĺžku reťazca

```
slovo = input('Napíš slovo:')
dlzka = len(slovo)
print('Počet znakov v slove je:', dlzka)
```

```
Napíš slovo:Lokomotíva
Počet znakov v slove je: 10
>>>
```

```
Napíš slová:Poprad je krásne mesto:-)
Počet znakov v slove je: 25
>>>
```

Index

- Číslo (v hranatých zátvorkách), pomocou ktorého sa vieme dostať ku konkrétnemu znaku reťazca
- indexovanie je od **0** po **dĺžku reťazca - 1**

```
retazec = 'Python'
print(retazec[0]) # vypíše 'P'
print(retazec[1]) # vypíše 'y'
```

P	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

```
a = 'Python'
for i in range(len(a)):
        print(i, a[i])
```

```
0 P
1 y
2 t
3 h
4 o
5 n
>>>
```

Funkcia **range(len(a))** zabezpečí, že cyklus prejde postupne pre všetky i od 0 do len(a)-1.

```
-1 n
-2 o
-3 h
-4 t
-5 y
-6 P
>>>
```

```
a = 'Python'
for i in range(1, len(a)+1):
        print(-i, a[-i])
```

Konštrukcia nového reťazca

- Reťazec (string) je v Pythone nemenný typ (immutable), to znamená, že sa nám nepodarí zmeniť znak, napr. retazec[0] = ‚J' (Python vtedy ohlási chybu).
- Namiesto zmeny reťazca musíme vždy konštruovať nový reťazec (môže mať aj rovnaké meno).

```
>>> meno = 'Hana'
>>> meno = 'J'+meno[1]+meno[2]+meno[3]
>>> meno
'Jana'
>>>
```

Celočíselné operácie

- oba operandy musia byť celočíselného typu

`+`	sčitovanie	`1 + 2` má hodnotu `3`
`-`	odčitovanie	`2 - 5` má hodnotu `-3`
`*`	násobenie	`3 * 37` má hodnotu `111`
`//`	celočíselné delenie,	`22 // 7` má hodnotu `3`
`%`	zvyšok po delení	`22 % 7` má hodnotu `1`
`**`	umocňovanie	`2 ** 8` má hodnotu `256`

Operácie s desatinnými číslami

- aspoň jeden z operandov musí byť desatinného typu (okrem delenia `/`)

`+`	sčitovanie	`1 + 0.2` má hodnotu `1.2`
`-`	odčitovanie	`6 - 2.86` má hodnotu `3.14`
`*`	násobenie	`1.5 * 2.5` má hodnotu `3.75`
`/`	delenie	`23 / 3` má hodnotu `7.666666666666667`
`//`	delenie zaokrúhlené nadol	`23.0 // 3` má hodnotu `7.0`
`%`	zvyšok po delení	`23.0 % 3` má hodnotu `2.0`
`**`	umocňovanie	`3 ** 3.` má hodnotu `27.0`

Operácie so znakovými reťazcami

`+`	zreťazenie (spojenie dvoch reťazcov)	`'a' + 'b'` má hodnotu `'ab'`
`*`	viacnásobné zreťazenie reťazca	`3 * 'x'` má hodnotu `'xxx'`

Python na prípady aktualizácie nejakej premennej ponúka špeciálny zápis priraďovacieho príkazu:

a = a+1	a+=1	a = a+b	a+=b
a = a-1	a-=1		
a = a*2	a*=2		
a = a/2	a/=2		
a = a//2	a//=2	celočíselné delenie	
a = a%2	a%=2	zvyšok po celočíselnom delení	
a = a**2	a**=2	druhá mocnina	

Domáca úloha

- e-učebnica: Peter Kučera: **Programujeme v Pythone**, učebnica informatiky pre SŠ, str. 12 14, úlohy 5, 8,10,12,13 vyriešiť
- úlohy 6, 7, 9, 11 preštudovať – napísať komentáre

Zdroje:

- e-učebnica: Peter Kučera: **Programujeme v Pythone**, učebnica informatiky pre SŠ
- Python http://input.sk/python2017/06.html
- Interaktivny Python

Ďakujem za pozornosť!