

Tutoriat 2

Șiruri de caractere (Stringuri)

Stringuri: - sunt un tip de date iterabil în Python.
- sunt imutabile, după declarare nu mai pot fi modificate.

```
s="test"  
s[1]='t'
```

```
Traceback (most recent call last):  
  File "<string>", line 3, in <module>  
TypeError: 'str' object does not support item assignment
```

Metode de declarare:

Exemple:

```
s1 = "s1 este un sir de caractere "  
s2 = 's2 este un sir de caractere'  
s3 = """ s3 este un sir de caractere  
        pe mai multe linii"""  
s4 = ''' s4 este un sir de caractere  
        pe mai multe linii'''  
s="apa"  
s5= s * n => s5 este format prin concatenarea lui s' de n ori , dacă n este  
        negativ sau 0 atunci se va memora ""
```

Ex:

```
s = 'ana' * 3 => s = 'anaanaana'  
s6=str(45)
```

Observații:

Ca-n alte limbaje de programare stringurile în python sunt vectori de caractere.

Stringurile sunt indentate începând cu 0.

În Python avem și indici negativi. De exemplu s[-1] reprezintă ultimul caracter din stringul s.

Parcurgerea stringurilor:

Metoda 1

```
for <variabila> in <string>:  
    print(<variabila>)
```

Metoda 2

```
for <variabila> in range(len(<string>))
    print( <string>[<variabila>] )
```

Slicing:

Tip accesare	Rezultat
s[i]	Elementul de pe poziția i în șir
s[i:]	returnează un subșir ce conține caracterele începând cu poziția i în s până la finalul șirului
s[i:j]	returnează un subșir ce conține caracterele începând cu poziția i în s până la poziția j-1
s[:]	returnează întreaga secvență
s[:j]	returnează un subșir ce conține caracterele începând cu poziția 0 în s până la poziția j-1
s[::-1]	returnează întreaga secvență în ordine inversă Ex: "sir"[::-1]="ris"

Metodele:

Metoda	Descriere
<șir>.find(<șir>)	Caută în șir o valoare specificată și returnează poziția unde l-a fost găsit (-1 dacă nu l-a găsit)
<șir>.isdigit()	Returnează True dacă toate caracterele din șir sunt cifre, altfel returnează False
<șir>.capitalize()	Convertește primul caracter în majuscule
<șir>.isupper()	Returnează True dacă toate caracterele din șir sunt majuscule, altfel returnează False
<șir>.upper()	Convertește un șir în majuscule
<șir>.split(<separatori>)	Împarte șirul la separatorul specificat și returnează o listă
<șir>.lower()	Convertește toate caracterele din șir în litere mici

<code><șir>.join(<iterabil>)</code>	Returnează un șir concatenat cu elementele lui iterable.
<code><șir>.replace(<old>,<new>)</code>)	Returnează un șir în care o valoare specificată `a` este înlocuită cu o valoare specificată `b`
<code><șir>.count(<șir>)</code>	Returnează numărul de apariții ale unui subșir în șir
<code>len(<șir>)</code>	Returnează lungimea șirului de caractere

! Pentru a verifica dacă un șir 's' conține subșir 'subsir' putem folosi **in**

Ex:

```
if subsir in s:
    print("Exista")
else:
    print("Nu exista")
```

Concatenarea de stringuri:

Se folosește operatorul +

Ex:

```
s1 = "sir1"
sir2 = "sir2"
sir3 = "sir1" + "sir2" # în sir3 va fi salvată valoarea "sir1sir2"
```

Link-uri utile:

- https://www.w3schools.com/python/python_strings_methods.asp pentru mai multe metode si exemple

