

STRING függvényei

Készítette: Vastag Attila

2020

Függvény	Leírás
capitalize()	Visszatér string másolatával úgy, hogy az első karaktert nagybetűssé alakítja
isalnum()	Igazzal tér vissza, ha a stringben minden karakter alfanumerikus, és legalább egy karakter hosszú.
isalpha()	Igazzal tér vissza, ha a stringben minden karakter alfabetikus, és legalább egy hosszú.
isspace()	Igazzal tér vissza, ha a strig csak whitespace karaktereket tartalmaz és legalább egyet.
istitle()	Igazzal tér vissza, ha a string címszerű, azaz minden szava nagybetűvel kezdődik, és legalább egy karaktert tartalmaz.
isupper()	Igazzal tér vissza, ha a string minden karaktere nagybetűs és legalább egy hosszú.
lower()	Visszatér azzal a stringgel, amelyben str minden karaktere kisbetűs (a string minden elemét kisbetűre alakítja).
swapcase()	Visszaadja azt a stringet, amelyben str kisbetűi nagybetűssé vannak konvertálva és fordítva.
title()	Visszaadja azt a stringet, amelyben str minden szava nagybetűvel kezdődik és a szavak további karaktere kisbetűs.
upper()	Visszadja str azon másolatát, amelyben a karakterek csupa nagybetűsek.

join(iterable)

Visszatér egy olyan stringgel, amely az iterable objektumból való stringek konkatenációja. A stringek közötti elválasztó az str string.

```
from typing import *
```

```
iterable: List[str] = ["alma", "korte", "dio"]  
result: str = ", ".join(iterable)
```

```
print(result)
```

```
print(result)
```

```
#output
```

```
"alma, korte, dio"
```

split

Lehetővé teszi egy karakterlánc szavakból álló listára való felbontását (alapértelmezetten a betűközöknél vagy a második argumentumként megadott stringnél).

```
from typing import *
```

```
text: str = "bonjour le monde"  
textAsList: List[str] = text.split()
```

```
print(textAsList)
```

```
#output  
['bonjour', 'le', 'monde']
```

```
from typing import *
```

```
text: str = "Katarzyna Skoworonska Dolata; Polland; 190; center"  
textAsList: List[str] = text.split(";")
```

```
print(textAsList)
```

```
#output  
['Katarzyna Skoworonska Dolata', ' Polland', ' 190', ' center']
```

split

#main

```
time: str = "16:00"  
minutes, seconds = time.split(":")
```

```
print(f"{minutes} min")  
print(f"{seconds} s")
```

#output

16 min

00 s

find, rfind

A **find** részstring keresését teszi lehetővé egy karakterláncban (a karakterlánc elejétől vagy a végétől kezdve az **rfind** esetén) visszatérési értékként azt az indexet megadva, ahol a részstringet először megtalálta. (Az index függvény hasonló, csak egy kivételt generál abban az esetben, ha nem találta meg a részstringet a karakterláncban.)

```
from typing import *
```

```
text: str = "bonjour le monde"
```

```
index: int = text.find("on")
```

#balról nézve

```
indexRight: int = text.rfind("on")
```

#jobbról nézve

```
print(index) #output 1
```

```
print(indexRight) #output 12
```

count

Egy részstring előfordulásának számlálását teszi lehetővé egy karakterláncban.

```
from typing import *  
  
text: str = "bonjour le monde"  
count: int = text.count("on")  
  
print(count)
```

#output

2

replace

A stringben egy rész-string egy másik stringgel való helyettesítését teszi lehetővé.

```
from typing import *
```

```
text: str = "bonjour le monde"  
text = text.replace("on", "ON")
```

```
print(text)
```

#output

bONjour le mONde