

جلسه ۲ - رشته‌ها و کار با متن در پایتون

در این جلسه یاد می‌گیریم:

- تعریف رشته‌ها (Strings)
- ایندکس‌گذاری و برش (Slicing)
- توابع و متدهای پرکاربرد رشته (String Formatting)
- ترکیب رشته‌ها (Concatenation)
- تکرار رشته‌ها (Repetition)

تعریف رشته‌ها

رشته یا String مجموعه‌ای از کاراکترهاست که داخل گیومه تکی ' ' یا دوتایی " " قرار می‌گیرد.

```
In [1]: text1 = "Hello"
         text2 = 'Python'
         print(text1)
         print(text2)
```

Hello
Python

ایندکس‌گذاری در رشته‌ها

هر کاراکتر در رشته یک شماره (ایندکس) دارد که از صفر شروع می‌شود.

```
In [2]: word = "Python"
         print(word[0]) # أولين حرف
         print(word[1]) # دومين حرف
         print(word[-1]) # آخرین حرف
```

P
y
n

برش (Slicing) در رشته‌ها

با استفاده از سینتکس `string[start:end]` می‌توانیم بخشی از رشته را جدا کنیم.

```
In [3]: text = "Programming"
         print(text[0:6]) # از اندیس 0 تا 5
         print(text[3:]) # از اندیس 3 تا انتها
```

```
print(text[:5])    # از ابتداء تا اندیس 4 پر شد
print(text[::2])   # پر شد تو تایی
```

```
Progra
gramming
Progr
Pormig
```

توابع و متدهای پرکاربرد رشته

چند متده مفید رشته ها:

- طول رشته را برمیگرداند. `()len`
- تمام حروف را بزرگ میکند. `()upper`
- تمام حروف را کوچک میکند. `()lower`
- جایگزین کردن متن. `replace(old, new)`
- تعداد تکرار زیررشته. `count(substring)`
- یافتن موقعیت زیررشته. `find(substring)`

```
In [4]: text = "Python programming"

print(len(text))
print(text.upper())
print(text.lower())
print(text.replace("Python", "Java"))
print(text.count("m"))
print(text.find("pro"))
```

```
18
PYTHON PROGRAMMING
python programming
Java programming
2
7
```

توابع و متدهای پیشرفته تر رشته

```
In [5]: text = " Python Programming Language 123 "
```

اولین حرف کل رشته را بزرگ و بقیه را کوچک میکند (حتی اگر فاصله در اول باشد، اثر روی اولین حرف بعد از فاصله است) :

```
In [6]: print(text.capitalize())
```

```
python programming language 123
```

اولین حرف هر کلمه را بزرگ میکند :

```
In [7]: print(text.title())
```

```
Python Programming Language 123
```

بزرگ‌ها را کوچک و کوچک‌ها را بزرگ می‌کند :

In [8]: `print(text.swapcase())`

```
pyTHON pROGRAMMING lANGUAGE 123
```

فاصله‌های خالی ابتداء و انتهای رشته را حذف می‌کند :

In [9]: `print(text.strip())`

```
Python Programming Language 123
```

فاصله‌های سمت چپ رشته را حذف می‌کند :

In [10]: `print(text.lstrip())`

```
Python Programming Language 123
```

فاصله‌های سمت راست رشته را حذف می‌کند :

In [11]: `print(text.rstrip())`

```
Python Programming Language 123
```

: بررسی می‌کند آیا رشته با "Py" شروع شده یا نه (False یا True)

In [12]: `print(text.startswith("Py"))`

```
False
```

بررسی می‌کند آیا رشته با "ge" تمام شده یا نه :

In [13]: `print(text.endswith("ge"))`

```
False
```

اولین موقعیت (ایندکس) "Pro" در رشته را برمی‌گرداند، اگر پیدا نشود -1 :

In [14]: `print(text.find("Pro"))`

```
10
```

آخرین موقعیت "g" را در رشته برمی‌گرداند :

In [15]: `print(text.rfind("g"))`

```
28
```

تعداد تکرار "m" در رشته را برمی‌گرداند :

In [16]: `print(text.count("m"))`

```
2
```

رشته را بر اساس فاصله‌ها به لیست کلمات جدا می‌کند :

In [17]: `print(text.split())`

```
[ 'Python', 'Programming', 'Language', '123' ]
```

رشته را از سمت راست، حداکثر به ۳ بخش بر اساس " " تقسیم می‌کند :

In [18]: `print(text.rsplit(" ", 2))`

```
' Python Programming Language 123 ', '', '' ]
```

رشته را بر اساس خطوط جدا می‌کند (کاراکتر \n) :

In [19]: `print("Line1\nLine2".splitlines())`

```
['Line1', 'Line2']
```

اعضای لیست را با "-" به هم وصل می‌کند :

In [20]: `print("-".join(["a", "b", "c"]))`

```
a-b-c
```

رشته "Python" را وسط‌چین کرده و با "-" پر می‌کند تا طولش ۲۰ شود :

In [21]: `print("Python".center(20, "-"))`

```
-----Python-----
```

رشته را چپ‌چین کرده و با "***" پر می‌کند :

In [22]: `print("Python".ljust(20, "*"))`

```
Python*****
```

رشته را راست‌چین کرده و با "***" پر می‌کند :

In [23]: `print("Python".rjust(20, "*"))`

```
*****Python
```

رشته را با صفر از سمت چپ پر می‌کند تا طولش ۵ شود (00042) :

In [24]: `print("42".zfill(5))`

```
00042
```

بررسی می‌کند که تمام کاراکترها حرف هستند (True) برای حروف فارسی هم کار می‌کند :

In [25]: `print("سلام".isalpha())`

```
True
```

بررسی می‌کند که همه کاراکترها عدد هستند :

```
In [26]: print("12345".isdigit())
```

True

بررسی می‌کند که همه کاراکترها یا حرف یا عدد باشند (بدون فاصله یا علامت) :

```
In [27]: print("abc123".isalnum())
```

True

بررسی می‌کند که رشته فقط شامل فاصله باشد :

```
In [28]: print(" ".isspace())
```

True

بررسی می‌کند که رشته به فرمت Title Case باشد (اولین حرف هر کلمه بزرگ) :

```
In [29]: print("Hello World".istitle())
```

True

قالب‌بندی رشته‌ها (String Formatting)

برای ترکیب متن و متغیرها می‌توانیم از:

- f-strings •

- `()format.` متدهای •

- % عملگر •

استفاده کنیم.

```
In [30]: name = "Ali"
```

```
age = 25
```

```
# f-string
```

```
print(f"My name is {name} and I am {age} years old.")
```

```
# format
```

```
print("My name is {} and I am {} years old.".format(name, age))
```

```
# % formatting
```

```
print("My name is %s and I am %d years old." % (name, age))
```

My name is Ali and I am 25 years old.

My name is Ali and I am 25 years old.

My name is Ali and I am 25 years old.

ترکیب رشته‌ها (Concatenation) و تکرار

می‌توان رشته‌ها را با `+` به هم چسباند و با `*` تکرار کرد.

```
In [31]: greet = "Hello"
lang = "Python"

# ترکیب
result = greet + " " + lang
print(result)

# تکرار
print("Hi! " * 3)
```

Hello Python
Hi! Hi! Hi!

تمرین ها

1. یک رشته از کاربر بگیرید و:
 - طول آن را چاپ کنید.
 - آن را به حروف بزرگ و کوچک تبدیل کنید.
2. یک رشته از کاربر بگیرید و تعداد کلمات داخل رشته را بشمارید
3. از کاربر یک جمله بگیرید و تعداد یک کلمه خاص را در آن بشمارید.
4. نام و نام خانوادگی کاربر را بگیرید و با قالب‌بندی (f-string) چاپ کنید.
5. از کاربر یک رشته بگیرید و بر عکس آن را چاپ کنید (با Slicing).
6. دو رشته از کاربر بگیرید و آنها را با یک فاصله بینشان ترکیب کنید.
7. رشته‌ای وارد کنید و خروجی رشته‌ای باشد که تمام فاصله‌های داخل آن حذف شده‌اند
8. رشته‌ای از کلمات وارد کنید، سپس بزرگترین (بیشترین طول) و کوچکترین (کمترین طول) کلمه را چاپ کنید
9. رشته‌ای وارد کنید و تمام حروف بزرگ را کوچک و حروف کوچک را بزرگ کنید (swap case)

1. یک رشته از کاربر بگیرید و:
 - طول آن را چاپ کنید.
 - آن را به حروف بزرگ و کوچک تبدیل کنید.

خروجی های قابل انتظار:

توضیح:

طول رشته ورودی را نمایش می‌دهد و همچنین رشته را به حروف بزرگ و کوچک تبدیل می‌کند.

:Input

Hello World

:Output

Length of string: 11

Uppercase: HELLO WORLD

Lowercase: hello world

2. یک رشته از کاربر بگیرید و تعداد کلمات داخل رشته را بشمارید

خروجی های قابل انتظار:

توضیح:

تعداد کلمات موجود در رشته را می‌شمارد.

:Input

This is a sample sentence

:Output

Number of words: 5

.3. از کاربر یک جمله بگیرید و تعداد یک کلمه خاص را در آن بشمارید.

خروجی های قابل انتظار:

توضیح:

تعداد دفعات تکرار یک کلمه خاص در جمله را پیدا می‌کند.

:Input

Sentence: I love Python because Python is simple

Word to count: Python

:Output

Number of occurrences of 'Python' in the sentence: 2

.4. نام و نام خانوادگی کاربر را بگیرید و با قالب‌بندی (f-string) چاپ کنید.

خروجی های قابل انتظار:

توضیح:

نام و نام خانوادگی را دریافت و به صورت قالب‌بندی شده چاپ می‌کند.

:Input

First name: Ali

Last name: Mohammadi

:Output

Your full name: Ali Mohammadi

.5. از کاربر یک رشته بگیرید و بر عکس آن را چاپ کنید (با Slicing).

خروجی های قابل انتظار:

توضیح:

رشته ورودی را معکوس می‌کند.

:Input

python

:Output

Reversed string: nohtyp

6. دو رشته از کاربر بگیرید و آنها را با یک فاصله بینشان ترکیب کنید.

خروجی های قابل انتظار:

توضیح:

دو رشته را با یک فاصله بین آنها به هم می‌چسباند.

:Input

First string: Hello

Second string: World

:Output

Combined string: Hello World

7. رشته‌ای وارد کنید و خروجی رشته‌ای باشد که تمام فاصله‌های داخل آن حذف شده‌اند

خروجی های قابل انتظار:

توضیح:

تمام فاصله‌های موجود در رشته را حذف می‌کند.

:Input

Hello World How Are You

:Output

String without spaces: HelloWorldHowAreYou

8. رشته‌ای از کلمات وارد کنید، سپس بزرگ‌ترین (بیشترین طول) و کوچک‌ترین (کمترین طول)

کلمه را چاپ کنید

خروجی های قابل انتظار:

توضیح:

کلمات موجود در رشته را بررسی کرده و بزرگ‌ترین و کوچک‌ترین کلمه را نمایش می‌دهد.

:Input

Programming Python is fun

:Output

Longest word: Programming

Shortest word: is

9. رشته‌ای وارد کنید و تمام حروف بزرگ را کوچک و حروف کوچک را بزرگ کنید (swap case)

خروجی های قابل انتظار:

توضیح:

حروف بزرگ را کوچک و حروف کوچک را بزرگ می‌کند.

:Input

PyTHON

:Output

Swapped case string: pYthON