

برنامه نویسی به زبان

# پایتون

مرجع کامل

تالیف

مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا  
مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از  
کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع  
کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرید

hadikiamarsi@gmail.com

لازم به ذکر است کلیه حقوق مادی و معنوی این اثر برای مولف محفوظ می  
باشد و هرگونه کپی برداری و استفاده از محتویات این کتاب به هر نوعی تحت  
پیگرد قانونی قرار می گیرد

# فصل هفتم

**در این فصل مطالب زیر را خواهید آموخت**

**رشته ها در پایتون ( python )**

**دسترسی به عناصر رشته ها**

**تغییر رشته ها**

**عملگرهای قالب بندی رشته ها**

**علامت نقل قول سه گانه ( Triple Quotes )**

**رشته های یونیکد ( Unicode String )**

**متدهای پیش ساخته رشته ها**

کیا مدتی

# رشته ها در پایتون ( python )

رشته ها از معروف ترین نوع داده ها در زبان برنامه نویسی پایتون ( python ) می باشد . در زبان برنامه نویسی پایتون ( python ) براحتی با استفاده از عملگر مساوی و علامت های نقل قول تکی ( single quotes ) و نقل قول جفتی ( double quotes ) می توان رشته ها را تعریف کرد برای آشنایی بیشتر با این مقوله به مثال زیر توجه نمایید

```
var1 = 'Hello World!'  
var2 = "Python Programming"
```

## دسترسی به عناصر رشته ها

برای دسترسی به عناصر رشته ها یعنی کاراکترها در زبان برنامه نویسی پایتون ( python ) از علامت براکت باز و بسته به همراه شماره اندیس کاراکتر استفاده می گردد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
  
var1 = 'Hello World!'  
var2 = "Python Programming"  
  
print "var1[0]: ", var1[0]  
print "var2[1:5]: ", var2[1:5]
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
var1[0]: H  
var2[1:5]: ytho
```

## تغییر رشته ها

برای تغییر رشته ها در زبان برنامه نویسی پایتون ( python ) باید رشته را مساوی به رشته جدید قرار دهید برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
  
var1 = 'Hello World!'  
print "Updated String :- ", var1[:6] + 'Python'
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

## عملگرهای قالب بندی رشته ها

یکی از بهترین ویژگی هایی که در زبان برنامه نویسی پایتون ( python ) برای رشته ها در نظر گرفته شده است قالب بندی رشته ها با استفاده از عملگر % می باشد زبان برنامه نویسی پایتون ( python ) این قابلیت خود را از زبان برنامه نویسی C به قرض گرفته است برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
print "My name is %s and weight is %d kg!" % ('Zara', 21)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
My name is Zara and weight is 21 kg!
```

لیست کاراکترهای قالب بندی رشته ها در زیر آورده شده است

کاراکتر قالب بندی	توضیحات
%c	کاراکتر
%s	رشته
%i	عدد صحیح علامت دار
%d	عدد صحیح علامت دار
%u	عدد صحیح بدون علامت
%o	عدد اکتال
%x	عدد هگزادسیمال با حروف کوچک
%X	عدد هگزادسیمال با حروف بزرگ
%e	عدد علمی با حروف کوچک
%E	عدد علمی با حروف بزرگ
%f	عدد اعشاری
%g	عدد علمی اعشاری
%G	عدد علمی اعشاری

## علامت نقل قول سه گانه ( Triple Quotes )

در زبان برنامه نویسی پایتون ( python ) برای ایجاد رشته های چند خطی از علامت نقل قول سه گانه ( Triple Quotes ) استفاده می گردد برای آشنایی بیشتر با این قسمت به مثال زیر توجه نمایید

```
#!/usr/bin/python

para_str = """this is a long string that is made up of
several lines and non-printable characters such as
TAB ( \t ) and they will show up that way when displayed.
NEWLINES within the string, whether explicitly given like
this within the brackets [ \n ], or just a NEWLINE within
the variable assignment will also show up.
"""
print para_str
```

بیاد داشته باشید علامت نقل قول سه گانه ( Triple Quotes ) که با استفاده از نقل قول تکی ( single quotes ) ایجاد شده باشد از قوانین نقل قول تکی ( single quotes ) پیروی می کند و علامت نقل قول سه گانه ( Triple Quotes ) که با استفاده از نقل قول جفتی ( double quotes ) ایجاد شده باشد از قوانین نقل قول جفتی ( double quotes ) پیروی می کند

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is a long string that is made up of
several lines and non-printable characters such as
TAB ( ) and they will show up that way when displayed.
NEWLINES within the string, whether explicitly given like
this within the brackets [
 ], or just a NEWLINE within
the variable assignment will also show up.
```

یک رشته Raw به رشته ای گفته می شود که هیچ کاراکتر خاصی برای آن وجود ندارد بنابراین کاراکترهای خاصی مانند کاراکتر خط جدید در آن کار نمی کند برای ایجاد همچین رشته ای یک حرف r در سمت راست رشته قرار می گیرد

```
#!/usr/bin/python

print 'C:\\nowhere'
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
C:\nowhere
```

در زیر مثالی از رشته Raw آورده شده است

```
#!/usr/bin/python
print r'C:\\nowhere'
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
C:\\nowhere
```

## رشته های یونیکد ( Unicode String )

رشته های عادی در زبان برنامه نویسی پایتون ( python ) با استفاده از کدهای اسکی 8 بیتی ذخیره می گردند در حالی که رشته های یونیکد در کدهای 16 بیتی ذخیره می گردند . رشته های 16 بیتی یونیکد برای نمایش کاراکترهای خاص و نمایش دیگر حروف زبان های دیگر دنیا مانند زبان فارسی بکار می رود .

```
#!/usr/bin/python
print u'Hello, world!'
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Hello, world!
```

در زبان برنامه نویسی پایتون ( python ) برای استفاده از رشته های یونیکد از یک حرف u قبل از رشته بکار می رود

## متدهای پیش ساخته رشته ها

در زبان برنامه نویسی پایتون ( python ) متدهای پیش ساخته ای وجود دارد که به برنامه نویسان در دستکاری رشته ها کمک شایانی می کند در زیر با برخی از پرکاربردترین این متدها توضیح داده شده اند

### متدهای رشته ها به همراه توضیحات

**capitalize ()**

تبدیل اولین حرف به حرف بزرگ

```
#!/usr/bin/python
```



```
str = "this is string example....wow!!!";
print "str.capitalize() : ", str.capitalize()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
str.capitalize() : This is string example....wow!!!
```

### **center(width, fillchar)**

این متد یک زیر رشته را به ابتدا و انتهای یک رشته اضافه می نماید

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";
print "str.center(40, 'a') : ", str.center(40, 'a')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
str.center(40, 'a') : aaaathis is string example....wow!!!aaaa
```

### **count(str, beg= 0,end=len(string))**

این متد تعداد تکرار یک زیر رشته در یک رشته را بر می گرداند

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";
sub = "i";
print "str.count(sub, 4, 40) : ", str.count(sub, 4, 40)
sub = "wow";
print "str.count(sub) : ", str.count(sub)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
str.count(sub, 4, 40) : 2
str.count(sub) : 1
```

```
decode(encoding='UTF-8',errors='strict')
```

این متد بر اساس الگوریتمی که برای آن تعیین می نماید یک رشته کد شده را از حالت کد خارج می نماید

```
#!/usr/bin/python
```

```
Str = "this is string example....wow!!!";  
Str = Str.encode('base64','strict');  
  
print "Encoded String: " + Str  
print "Decoded String: " + Str.decode('base64','strict')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Encoded String: dGhpcyBpcyBzdHJpbmVmcGZxhbXBsZS4uLi53b3chISE=  
Decoded String: this is string example....wow!!!
```

```
encode(encoding='UTF-8',errors='strict')
```

این متد بر اساس الگوریتمی که برای آن تعیین می نماید یک رشته را به کد تبدیل می نماید

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";  
print "Encoded String: " + str.encode('base64','strict')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Encoded String: dGhpcyBpcyBzdHJpbmVmcGZxhbXBsZS4uLi53b3chISE=
```

```
endswith(suffix, beg=0, end=len(string))
```

این متد بررسی می نماید که یک رشته به چه زیر رشته ای ختم می گردد

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";  
  
suffix = "wow!!!";
```

```
print str.endswith(suffix)
print str.endswith(suffix,20)

suffix = "is";
print str.endswith(suffix, 2, 4)
print str.endswith(suffix, 2, 6)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
True
True
False
```

**expandtabs (tabsize=8)**

این متد اندازه کاراکتر tab را تعیین می نماید

```
#!/usr/bin/python

str = "this is\tstring example....wow!!!";

print "Original string: " + str
print "Default expanded tab: " + str.expandtabs()
print "Double expanded tab: " + str.expandtabs(16)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Original string: this is      string example....wow!!!
Default expanded tab: this is string example....wow!!!
Double expanded tab: this is      string example....wow!!!
```

**find(str, beg=0 end=len(string))**

این متد موقعیت یک زیر رشته را در یک رشته بر می گرداند و اگر زیر رشته در آن رشته نبود -1 را بر می گرداند

```
#!/usr/bin/python

str1 = "this is string example....wow!!!";
str2 = "exam";

print str1.find(str2)
print str1.find(str2, 10)
```

```
print str1.find(str2, 40)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
15
15
-1
```

**index(str, beg=0, end=len(string))**

این متد موقعیت یک زیر رشته را در یک رشته بر می گرداند و اگر زیر رشته در آن رشته نبود یک پیغام خطا صادر می گردد

```
#!/usr/bin/python
str1 = "this is string example....wow!!!";
str2 = "exam";

print str1.index(str2)
print str1.index(str2, 10)
print str1.index(str2, 40)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
15
15
Traceback (most recent call last):
  File "test.py", line 8, in
    print str1.index(str2, 40);
ValueError: substring not found

shell returned 1
```

**isalnum()**

این متد اگر رشته تنها شامل عدد و حرف باشد True را بر می گرداند و در غیر این صورت False را بر می گرداند

```
#!/usr/bin/python
str = "this2009"; # No space in this string
```

```
print str.isalnum()

str = "this is string example....wow!!!";
print str.isalnum()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

### **isalpha()**

این متد اگر رشته فقط شامل حروف انگلیسی باشد True را بر می گرداند و در غیر این صورت False را بر می گرداند

```
#!/usr/bin/python

str = "this"; # No space & digit in this string
print str.isalpha()

str = "this is string example....wow!!!";
print str.isalpha()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

### **isdigit()**

این متد True را بر می گرداند اگر رشته یونیکد شامل عدد باشد و در غیر این صورت False بر می گرداند

```
#!/usr/bin/python

str = "123456"; # Only digit in this string
print str.isdigit()

str = "this is string example....wow!!!";
print str.isdigit()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

### **islower()**

این متد اگر کلیه حروف رشته شامل حروف کوچک باشد True را بر می گرداند و در غیر این صورت False را بر می گرداند

```
#!/usr/bin/python

str = "THIS is string example....wow!!!";
print str.islower()

str = "this is string example....wow!!!";
print str.islower()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
False
True
```

### **isnumeric()**

این متد True را بر می گرداند اگر رشته یونیکد تنها شامل عدد باشد و در غیر این صورت False بر می گرداند

```
#!/usr/bin/python

str = u"this2009";
print str.isnumeric()

str = u"23443434";
print str.isnumeric()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
False
True
```

## **isspace()**

این متد اگر رشته فقط شامل کاراکتر جای خالی ( space ) باشد True را بر می گرداند و در غیر این صورت False را بر می گرداند

```
#!/usr/bin/python

str = "      ";
print str.isspace()

str = "This is string example....wow!!!";
print str.isspace()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

## **istitle()**

این متد اگر تمام حروف اول هر کلمه بزرگ باشد True را بر می گرداند و در غیر این صورت False را بر می گرداند

```
#!/usr/bin/python

str = "This Is String Example...Wow!!!";
print str.istitle()

str = "This is string example....wow!!!";
print str.istitle()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

## **isupper()**

این متد اگر تمام حروف یک رشته حروف بزرگ باشند True را بر می گرداند در غیر این

صورت False را بر می گرداند

```
#!/usr/bin/python
str = "THIS IS STRING EXAMPLE....WOW!!!";
print str.isupper()

str = "THIS is string example....wow!!!";
print str.isupper()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
False
```

### **join(seq)**

این متد چند رشته را با هم ترکیب می نماید

```
#!/usr/bin/python
s = "-";
seq = ("a", "b", "c"); # This is sequence of strings.
print s.join( seq )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
a-b-c
```

### **len(string)**

این متد اندازه یک رشته را بر می گرداند

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print "Length of the string: ", len(str)
```



اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Length of the string: 32
```

### **ljust(width[, fillchar])**

به سمت چپ یک رشته کاراکتری که تعیین شده است را اضافه می نماید تا طول رشته به اندازه تعیین شده برسد

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print str.ljust(50, '0')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is string example....wow!!!00000000000000000000
```

### **lower()**

این متد تمام حروف کوچک یک رشته را به حروف بزرگ تبدیل می نماید

```
#!/usr/bin/python
str = "THIS IS STRING EXAMPLE....WOW!!!";
print str.lower()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is string example....wow!!!
```

### **rstrip()**

این متد فضای خالی سمت چپ رشته را حذف یا جایگزین می کند

```
#!/usr/bin/python
```

```
str = "    this is string example....wow!!!    ";
print str.lstrip()
str = "88888888this is string example....wow!!!88888888";
print str.lstrip('8')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is string example....wow!!!
this is string example....wow!!!88888888
```

## **Makestrans ()**

این متد یک جدول برای استفاده متد translate ایجاد می کند

```
#!/usr/bin/python
from string import maketrans # Required to call maketrans function.

intab = "aeiou"
outtab = "12345"
trantab = maketrans(intab, outtab)

str = "this is string example....wow!!!"
print str.translate(trantab)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
th3s 3s str3ng 2x1mpl2....w4w!!!
```

## **max(str)**

کاراکتری از رشته که بزرگترین عدد اسکی را دارد را بر می گرداند

```
#!/usr/bin/python
str = "this is really a string example....wow!!!";
print "Max character: " + max(str)

str = "this is a string example....wow!!!";
print "Max character: " + max(str)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Max character: y
Max character: x
```

## `min(str)`

کاراکتری از رشته که کوچکترین عدد اسکی را دارد را بر می گرداند

```
#!/usr/bin/python

str = "this-is-real-string-example....wow!!!";
print "Min character: " + min(str)

str = "this-is-a-string-example....wow!!!";
print "Min character: " + min(str)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Min character: !
Min character: !
```

## `replace(old, new [, max])`

از این متد برای جایگزینی یک رشته با یک رشته جدید استفاده می گردد

```
#!/usr/bin/python

str = "this is string example....wow!!! this is really string"
print str.replace("is", "was")
print str.replace("is", "was", 3)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
thwas was string example....wow!!! thwas was really string
thwas was string example....wow!!! thwas is really string
```

## `rfind(str, beg=0, end=len(string))`

یک زیر رشته را در یک رشته جستجو می کند و موقعیت آن را از آخر اعلام می نماید

```
#!/usr/bin/python

str1 = "this is really a string example....wow!!!";
str2 = "is";

print str1.rfind(str2)
print str1.rfind(str2, 0, 10)
print str1.rfind(str2, 10, 0)

print str1.find(str2)
print str1.find(str2, 0, 10)
print str1.find(str2, 10, 0)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
5
5
-1
2
2
-1
```

**`rindex( str, beg=0, end=len(string))`**

یک زیر رشته را در یک رشته جستجو می کند و موقعیت آن را از آخر اعلام می نماید

```
#!/usr/bin/python

str1 = "this is string example....wow!!!";
str2 = "is";

print str1.rindex(str2)
print str1.index(str2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
5
2
```

## **rjust(width, [, fillchar])**

با اضافه کردن کاراکتر تعیین شده طول رشته را به اندازه تعیین شده می رساند

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print str.rjust(50, '0')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
0000000000000000000000000000000000this is string example....wow!!!
```

## **rstrip()**

فضای خالی سمت راست یک رشته را حذف یا جایگزین می کند

```
#!/usr/bin/python
str = "   this is string example....wow!!!   ";
print str.rstrip()
str = "88888888this is string example....wow!!!88888888";
print str.rstrip('8')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is string example....wow!!!
88888888this is string example....wow!!!
```

## **split(str="", num=string.count(str))**

چند قسمت کردن رشته بر اساس کاراکتر یا رشته ای که برای آن تعیین می نماید

```
#!/usr/bin/python
str = "Line1-abcdef \nLine2-abc \nLine4-abcd";
```

```
print str.split( )
print str.split(' ', 1 )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
['Line1-abcdef', 'Line2-abc', 'Line4-abcd']
['Line1-abcdef', '\nLine2-abc \nLine4-abcd']
```

```
splitlines( num=string.count('\n'))
```

چند قسمت کردن رشته بر اساس کاراکتر خط جدید ( \n )

```
#!/usr/bin/python
```

```
str = "Line1-a b c d e f\nLine2- a b c\n\nLine4- a b c d";
print str.splitlines( )
print str.splitlines( 0 )
print str.splitlines( 3 )
print str.splitlines( 4 )
print str.splitlines( 5 )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
['Line1-a b c d e f', 'Line2- a b c', '', 'Line4- a b c d']
['Line1-a b c d e f', 'Line2- a b c', '', 'Line4- a b c d']
['Line1-a b c d e f\n', 'Line2- a b c\n', '\n', 'Line4- a b c d']
['Line1-a b c d e f\n', 'Line2- a b c\n', '\n', 'Line4- a b c d']
['Line1-a b c d e f\n', 'Line2- a b c\n', '\n', 'Line4- a b c d']
```

```
startswith(str, beg=0, end=len(string))
```

بررسی اینکه رشته با چه رشته ای شروع می گردد

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";
print str.startswith( 'this' )
print str.startswith( 'is', 2, 4 )
print str.startswith( 'this', 2, 4 )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
True
True
```

```
False
```

### **strip([chars])**

کاراکتری را که برای آن مشخص می نماید از اول و آخر رشته حذف می نماید

```
#!/usr/bin/python
```

```
str = "0000000this is string example....wow!!!0000000";  
print str.strip( '0' )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
this is string example....wow!!!
```

### **swapcase()**

این متد حروف بزرگ را به کوچک و حروف کوچک را به بزرگ تبدیل می نماید

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";  
print str.swapcase()
```

```
str = "THIS IS STRING EXAMPLE....WOW!!!";  
print str.swapcase()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
THIS IS STRING EXAMPLE....WOW!!!  
this is string example....wow!!!
```

### **title()**

حروف اول هر کلمه از یک رشته را به حروف بزرگ تبدیل می نماید

```
#!/usr/bin/python
```

```
str = "this is string example....wow!!!";  
print str.title()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
This Is String Example....Wow!!!
```

```
translate(table, deletechars="")
```

برای جایگزین کردن حروف یک رشته بکار می رود

```
#!/usr/bin/python  
  
from string import maketrans # Required to call maketrans function.  
  
intab = "aeiou"  
outtab = "12345"  
trantab = maketrans(intab, outtab)  
  
str = "this is string example....wow!!!";  
print str.translate(trantab)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
th3s 3s str3ng 2x1mpl2....w4w!!!
```

مثالی دیگر

```
#!/usr/bin/python  
  
from string import maketrans # Required to call maketrans function.  
  
intab = "aeiou"  
outtab = "12345"  
trantab = maketrans(intab, outtab)  
  
str = "this is string example....wow!!!";  
print str.translate(trantab, 'xm')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
th3s 3s str3ng 21pl2....w4w!!!
```



## Upper ()

تبدیل تمام حروف کوچک یک رشته به حروف بزرگ

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print "str.capitalize() : ", str.upper()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
str.capitalize() : THIS IS STRING EXAMPLE....WOW!!!
```

## zfill (width)

بر اساس طولی که برای آن در نظر می گیرید صفر به رشته اضافه می نماید تا به طول مورد نظر تعیین شده برسد

```
#!/usr/bin/python
str = "this is string example....wow!!!";
print str.zfill(40)
print str.zfill(50)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
00000000this is string example....wow!!!
00000000000000000000this is string example....wow!!!
```

## isdecimal ()

اگر رشته یونیکد تنها شامل عدد باشد True را بر می گرداند در غیر این صورت False بر می گرداند

```
#!/usr/bin/python
```

```
str = u"this2009";  
print str.isdecimal();
```

```
str = u"23443434";  
print str.isdecimal();
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
False  
True
```

دی کیامدتی