

## فرمت‌بندی رشته

پایتون قابلیت‌های زیادی برای فرمت‌بندی رشته‌ها (*String Formatting*) در اختیار ما گذاشته است که می‌توان از آن‌ها برای قراردادن مقدار متغیرها درون رشته استفاده کرده تا کد را بهینه‌تر بنویسیم.

### استفاده از f برای فرمت‌بندی رشته

ساده‌ترین روش برای فرمت‌بندی رشته، استفاده از کاراکتر f است. به این صورت که قبل از رشته مورد نظر کاراکتر f را می‌نویسید. حال برای قرار دادن مقدار یک متغیر در هر جای رشته کافی است آن را بین یک {} بنویسید. برای درک این قابلیت به مثال زیر توجه کنید:

```
1 >>> age = 12
2 >>> height = 178.5
3 >>> isMale = True
4 >>> f'Age : {age}, Height : {height}, Gender : {isMale}'
5 'Age : 12, Height : 178.5, Gender : True'
```

همچنین در پایتون ۳.۸ می‌توانید هنگام استفاده از f با قرار دادن علامت = جلوی یک متغیر، اسم آن متغیر را نیز به همراه مقدارش چاپ کنید. این کار هنگام چاپ کردن مقادیر متغیرها برای دیباگ کد می‌تواند مفید باشد.

```
1 >>> variable = 5
2 >>> f"{variable=}"
3 'variable=5'
```

### استفاده از % برای فرمت‌بندی رشته

یک روش دیگر استفاده از عملگر % بعد از String است. شیوه کار این عملگر به این ترتیب است که ابتدا در رشته مورد نظر یکی از عبارت‌های %d یا %f یا %s (به ترتیب برای متغیرهای صحیح، اعشاری و رشته‌ای) را

قرار می‌دهیم. سپس به دنبال رشته یا متغیر رشته‌ای و بعد از عملگر % متغیری که می‌خواهیم جای مقدار نوشته شده در رشته قرار بگیرد را می‌نویسم (همانند C). برای مثال به کد زیر توجه کنید:

```
1 | greeting = "Hello %s!"
2 | name = input("Please enter your name:\n")
3 | print(greeting % name)
```

این کد یک اسم از کاربر گرفته و در خروجی به او سلام می‌کند. توجه کنید که می‌توانیم به تابع input() یک رشته ورودی بدهیم. این رشته قبل از ورودی گرفتن از کاربر در کنسول به عنوان prompt چاپ خواهد شد. (کاربر میتواند متن قرار گرفته در متد input را در کنسول مشاهده کند.)

اگر بخواهیم در رشته مورد نظرمون بیش از یک متغیر قرار دهیم، می‌توانیم بعد از عملگر % همه مقادیری که می‌خواهیم در رشته قرار بگیرند را به ترتیبی که در درون رشته آمده‌اند، درون پرانتز بنویسیم و مفسر همه مقادیر را به ترتیب درون رشته جایگزین می‌کند. برای مثال فرض کنید که در مثال بالا می‌خواهیم علاوه بر اسم کاربر، سن او را نیز بپرسیم و آن را نمایش دهیم. در آن صورت چنین کدی خواهیم داشت :

```
1 | greeting = "Hello %s with %d years old!"
2 | name = input("Please enter your name:\n")
3 | age = int(input("Please enter your age:\n"))
4 | print(greeting % (name, age))
```

توجه کنید که اگر تعداد مقادیری که درون پرانتز می‌نویسیم با تعداد کل مقادیری که رشته نیاز دارد برابر نباشد، مفسر خطا برمیگرداند. همچنین در صورتی که بخواهید در رشته خود کاراکتر % را داشته باشید، می‌توانید از %% استفاده کنید.

**نکته:** برای ذخیره اعداد اعشاری با دقت مشخص می‌توانید از %Xf استفاده کنید که در آن X همان تعداد رقم اعشاری است که می‌خواهید چاپ کنید. (برای مثال برای ذخیره عدد اعشاری با دو رقم اعشار می‌توانید از %.2f استفاده کنید.)

استفاده از format() برای فرمت‌بندی رشته

یکی از روش‌های دیگر برای این‌کار که می‌تواند کار ما را ساده‌تر کند استفاده از دستور `format()` است. برای استفاده از این دستور دو شیوه وجود دارد که در اینجا به توضیح آن‌ها می‌پردازیم (این روش، که تنها در پایتون ۳ وجود دارد، یکی از وجوه تمایز این نسخه از پایتون با نسخه‌های قبلی آن است).

## روش اول

در این روش مکان‌هایی از رشته که قرار است با مقادیر دیگر جایگزین شوند را با استفاده از `{}` مشخص کرده و تابع `format` رشته را با مقادیری که می‌خواهید در آن مکان‌ها قرار بگیرد صدا کنید. در این صورت در خروجی رشته‌ای دریافت خواهید کرد که در آن مکان‌های `{}` به‌ترتیب با ورودی‌های تابع جایگزین شده‌اند. برای مثال کد مثال قبل را اکنون با استفاده از این تابع دوباره می‌زنیم.

```
1 greeting = "Hello {} with {} years old!"
2 name = input("Please enter your name:\n")
3 age = int(input("Please enter your age:\n"))
4 print(greeting.format(name, age))
```

برخلاف عملگر `%`، در این حالت تنها زمانی مفسر دچار استثنا می‌شود (خطا برمیگرداند) که تعداد ورودی‌های تابع کمتر از تعداد ورودی‌های خواسته شده باشد. (اگر تعداد ورودی‌ها بیشتر باشد به ورودی‌های اضافه توجهی نمی‌کند)

همچنین می‌توان درون `{}` یک : نوشت و بعد از آن توضیحاتی درباره این که آن مقدار چگونه باید چاپ شود داد. برخی از این فرمت‌ها و عملکردشان در زیر آمده‌اند :

فرمت	عملکرد
<code>{: +}</code>	اگر عدد نامنفی بود، قبل از آن یک علامت <code>+</code> قرار می‌دهد
<code>{: .Xf}</code>	عدد داده شده را با <code>X</code> رقم اعشار ذخیره می‌کند
<code>{: b}</code>	عدد داده شده را به مبنای ۲ برده و آن را در رشته می‌ریزد
<code>{: o}</code>	عدد داده شده را به مبنای ۸ برده و آن را در رشته می‌ریزد

فرمت	عملکرد
{:x}	عدد داده شده را به مبنای ۱۶ برده و آن را در رشته می‌ریزد
{:x=y}	تا زمانی که طول رشته از y کمتر باشد، به ابتدای آن کاراکتر x را اضافه می‌کند

در کد زیر می‌توانید طرز استفاده از این فرمت‌ها را ببینید :

```

1 >>> "{:+} {:+} {:+}".format(241.1, -124.21, 0)
2 '+241.1 -124.21 +0'
3 >>> "{:.3f} {:.3f}".format(122, 33.22191)
4 '122.000 33.222'
5 >>> "{:b} {:b}".format(-34, 21)
6 '-100010 10101'
7 >>> "{:x} {:x}".format(241, 124)
8 'f1 7c'
9 >>> "{:o} {:o}".format(241, 124)
10 '361 174'
11 >>> "{:0=8} {:1=8} {:2=2}".format(12, 21.12, 312)
12 '00000012 11121.12 312'

```

## روش دوم

اصلی‌ترین وجه تمایزی که استفاده از `format()` نسبت به عملگر `%` دارد و باعث شده تا در پایتون ۳ اضافه شود همین روش است. در این روش می‌توان درون `{}` یک عدد مثل `i` قرار داد که نشان می‌دهد عضو `ام` ورودی باید این جا قرار گیرد. همچنین می‌توان درون `{}` یک متغیر قرار داد، سپس در ورودی‌های تابع آن را مقداردهی کرد و آن مقدار در آن جا قرار می‌گیرد. برای درک بیشتر به مثال زیر توجه کنید.

```

1 >>> "Hello {0} with {1} years old!".format("Ali", "23")
2 'Hello Ali with 23 years old!'
3 >>> "Hello {name} with {age} years old!".format(name = "Ali", age = "23")
4 'Hello Ali with 23 years old!'
5 >>> "Hello {0} with {age} years old!".format("Ali", age = "23")
6 'Hello Ali with 23 years old!'

```

توجه کنید که در صورت به‌کاربردن این روش، دیگر نمی‌توانید از `{}` به صورت خالی استفاده کنید. همچنین از توضیحاتی که در روش قبل، بعد از `:` می‌آمد، نیز می‌توانیم در این‌جا به شکل زیر استفاده کنیم.

```
1 >>> "{0:.3f} {0:+}".format(23.12)
2 '23.120 +23.12'
3 >>> "{num:.3f} {num:+}".format(num=23.12)
4 '23.120 +23.12'
```