

ورودی و خروجی استاندارد

تابع print

در پایتون ۳، از تابع `print` برای چاپ در خروجی استاندارد استفاده می‌شود. امضای (*signature*) این تابع به شکل زیر است.

```
1 | print(txt_1, txt_2, ... , txt_n, end='\n', sep=' ', file=sys.stdout, flush=True)
2 | #print function signature and it's default options
```

[Copy](#)[Python](#)

- `end` : پایتون در آخر هر عبارت چاپ شده این آرگومان را می‌نویسد. مقدار پیش‌فرض این آرگومان `'\n'` است و یعنی بعد از هر چاپ به خط بعد می‌رود.
- `sep` : این آرگومان بین هر دو آرگومان چاپ شده نوشته می‌شود.
- `file` : شی فایلی که در آن نوشته می‌شود را مشخص می‌کند که در حالت عادی خروجی استاندارد است.
- `flush` : مشخص می‌کند در آخر چاپ، خروجی را `flush` کند یا خیر.

```
1 | >>> print("Hello World")
2 | Hello World
3 | >>> print("Hello World",end="!")
4 | Hello World!>>> print("a", "b", sep=",") #it didn't go to next line!
5 | a,b
```

تابع input

```
1 | input(greeting_text)
```

این تابع آرگومان اول خود را که یک متن است چاپ می‌کند و پس از آن از ورودی یک خط می‌خواند. و آن را به عنوان یک رشته برمی‌گرداند.

```
1 | >>> a = input("Please enter your age : ")
2 | Please enter your age : 15
3 | >>> a
4 | '15'
```

همان‌طور که می‌بینید خروجی تابع `input` یک رشته است؛ و اگر نوع داده‌ی دیگری لازم داشته باشید، باید آن را تبدیل کنید.

```
1 | >>> str_input = input("Age : ")
2 | Age : 15
3 | >>> str_input
4 | '15'
5 | >>> int_input = int(str_input)
6 | >>> int_input
7 | 15
```

```
8 >>> float_input = float(str_input)
9 >>> float_input
10 15.0
```

همان‌طور که می‌بینید ورودی به صورت رشته برگردانده شده و به نوع داده‌های دلخواه تبدیل شده است.

```
1 >>> line = input()
2 12 13 14 17 19 10 9 85
3 >>> line
4 '12 13 14 17 19 10 9 85'
5 >>> line.split()
6 ['12', '13', '14', '17', '19', '10', '9', '85']
7 >>> map(int, line.split())
8 <map object at 0x0000027A236E1358>
9 >>> list(map(int, line.split()))
10 [12, 13, 14, 17, 19, 10, 9, 85]
11 >>> list(map(float, line.split()))
12 [12.0, 13.0, 14.0, 17.0, 19.0, 10.0, 9.0, 85.0]
```

ورودی‌هایی که در یک خط چند عدد دارند معمولاً به شکل بالا ابتدا با تابع `split` از هم جدا می‌شوند سپس روی تمامی اعضای لیست، تابع تبدیل (در مثال بالا `int` و `float`) اعمال می‌شود. `map` تابعی است که یک لیست و یک تابع را گرفته و تابع را روی همه اعضای لیست اعمال می‌کند و خروجی را به صورت یک `map` برمی‌گرداند. این شی `iterable` است و می‌توان روی آن حلقه زد یا همه عناصر آن را به صورت لیست گرفت.

```
1 >>> x, y = map(int, input().split())
2 12 14
3 >>> x
4 12
5 >>> y
6 14
```

اگر تعداد خاصی متغیر را بخواهیم می‌توانیم از قابلیت `unpacking` پایتون استفاده کنیم و در مثال بالا ۲ ورودی اول خط را جای `x` و `y` می‌گذارد.