



**Namatek**

True Education

# Python Functions

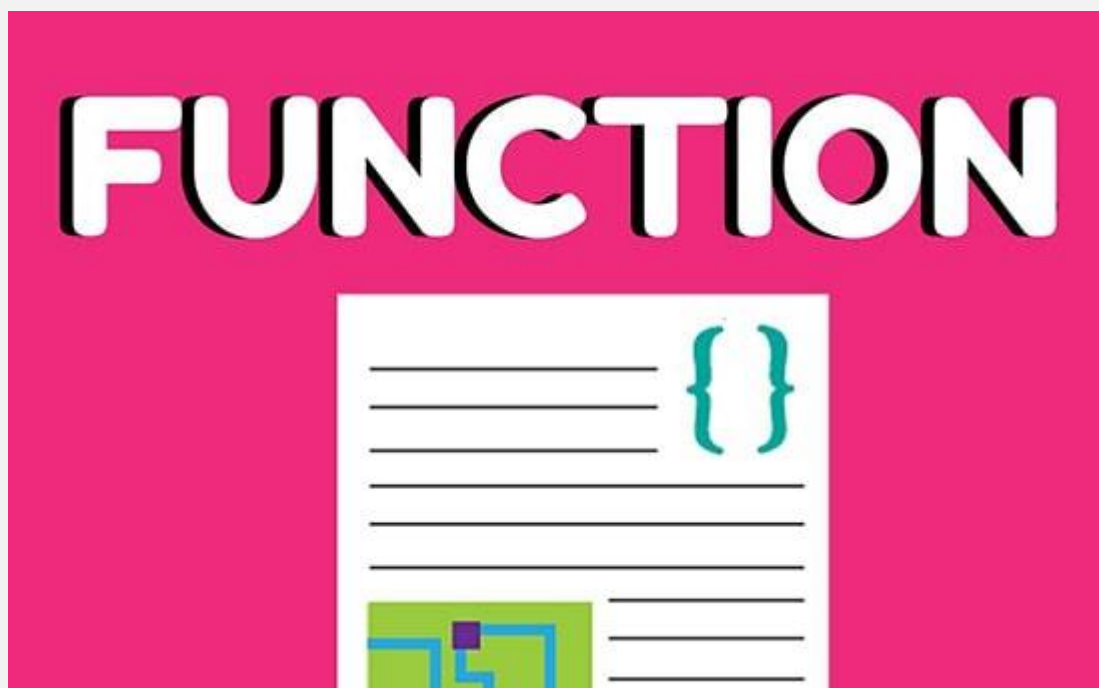
شناخت توابع پایتون  
در ۴ گام

## فهرست مطالب

۱. تابع پایتون چیست؟
۲. انواع توابع پایتون (Python Functions)
۳. نحوه تعریف توابع در پایتون
۴. انواع فراخوانی توابع پایتون

توابع پایتون یکی از بخش های مهم و کلیدی در این زبان برنامه نویسی هستند. چرا که توابع در کپسوله سازی و ساختاردهی کدها تاثیر به سزایی دارند. به همین دلیل در این مقاله به معرفی تابع در زبان برنامه نویسی پایتون، انواع آن ها و نحوه تعریف و فراخوانی توابع خواهیم پرداخت. در ادامه با ما همراه باشید.

## #۱ تابع پایتون چیست؟



توابع (Functions) یک ویژگی مشترک بین همه زبان های برنامه نویسی هستند. تابع یک ساختار یا بلوکی از کدها به صورت سازمان یافته و قابل استفاده مجدد است که برای انجام یک دستورالعمل استفاده می

شود. یک تابع می تواند هر چند بار که یک توسعه دهنده بخواهد در سراسر کد خود اجرا شود.

توابع می توانند یک بلوک کد را چندین بار در یک برنامه اجرا کنند و به توسعه دهندگان این امکان را بدهند که تکرار را در کد خود کاهش دهند. بنابراین توابع ساختاردهی بهتری برای برنامه شما ارائه می دهند. علاوه بر این قابلیت استفاده مجدد از کد را برای برنامه نویسان فراهم می کنند. به عنوان مثال تابعی را در نظر بگیرید که جذر اعداد را محاسبه می کند. این تابع می تواند بسته به نیاز برنامه نویس ده ها ورودی و خروجی مختلف داشته باشد.

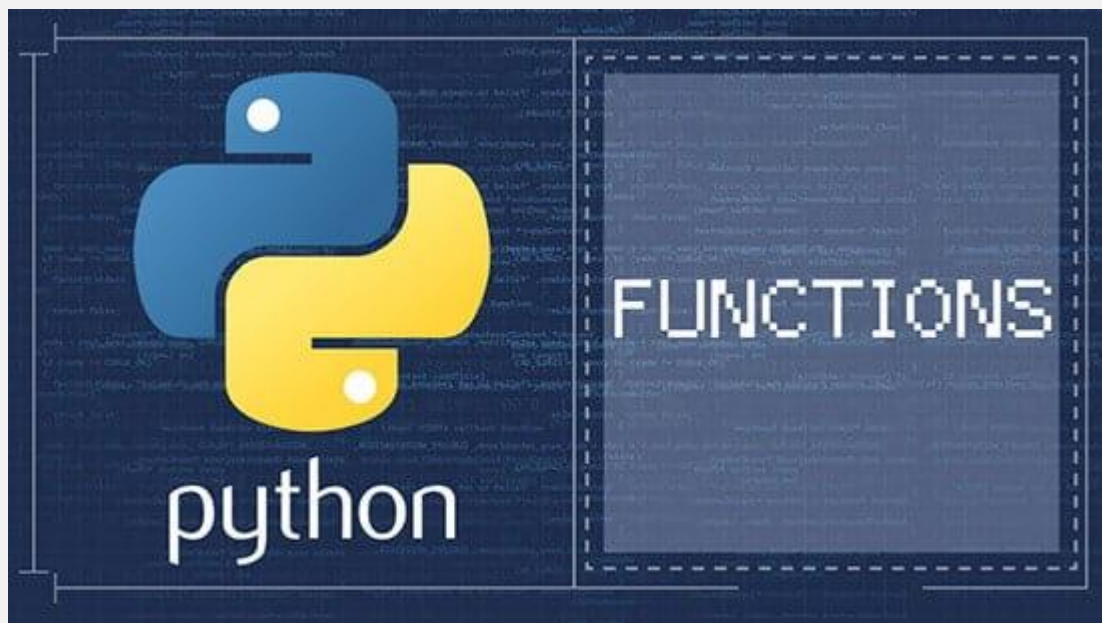
به این عمل استفاده مجدد از کد گفته می شود؛ یعنی قطعه کدی که برای محاسبات مختلف چندین بار مورد استفاده قرار گیرد.

همان طور که می دانید، پایتون نیز مانند سایر زبان های برنامه نویسی شامل تعداد زیادی از توابع داخلی، مانند `print()` و غیره، است؛ اما در کنار این توابع از پیش تعریف شده، شما می توانید توابع خود را نیز در پایتون ایجاد کنید. این توابع را توابع تعریف شده توسط کاربر (`user-defined functions`) می نامند.

برای اجرای کد در توابع پایتون، باید تابع مورد نظر را فراخوانی کنید. پس از تعریف تابع می توان یک تابع را از هر جایی فراخوانی کرد. از طریق فراخوانی، توابع اجرا می شوند و پس از اجرا می توانند خروجی نهایی را نمایش دهند. البته یک تابع به خودی خود هیچ کاری انجام نمی دهد؛

اما زمانی که نیاز به استفاده از یک تابع دارید، می توانید آن را فراخوانی کنید؛ در این زمان کد داخل تابع اجرا خواهد شد.

## #۲ انواع توابع پایتون (Python Functions)



در پایتون دو نوع تابع وجود دارد:

- تعریف شده توسط کاربر (user-defined functions):

توابع تعریف شده توسط کاربر، بلوک های کد قابل استفاده مجدد هستند که توسط توسعه دهنده نوشته شده اند. این بلوک های کد به توسعه دهندگان این امکان را می دهند که کد خود را با کارایی بیشتری سازماندهی کنند. سازماندهی کدها بسیار مهم است؛ زیرا هرچه کد شما سازماندهی شده تر باشد، نگهداری آن نیز آسان تر خواهد بود.

- توابع داخلی پایتون (built-in functions):

توابع داخلی، توابع پرکاربردی هستند که از قبل در پایتون تعریف شده اند.

برخی از توابع داخلی عبارت اند از:

- `print ()` که یک عبارت را در کنسول چاپ می کند.
- `Python len ()` که طول یک لیست را محاسبه می کند.
- `Python str ()` که یک مقدار را به یک رشته تبدیل می کند.

## #۳ نحوه تعریف توابع در پایتون

تعریف یا ایجاد توابع پایتون شامل نوشتن یک بلوک از کد است که می توانیم با ارجاع به نام تابع، آن را فراخوانی کنیم. قوانین ساده ای برای تعریف یک تابع در پایتون وجود دارند.

- توابع پایتون با کلمه کلیدی `def` و به دنبال آن نام تابع و پرانتز مشخص می شوند.
- هر پارامتر یا آرگومان ورودی باید در داخل این پرانتز قرار گیرد. علاوه بر این در پایتون می توانید پارامترهایی را در داخل این پرانتزها تعریف کنید.
- اولین دستور یک تابع می تواند یک دستور اختیاری، به عنوان مثال توضیحات و مستندات تابع باشد.

• بلوک کد در هر تابع باید با کالن (Colon) یا دونقطه (: ) شروع شود.

به عنوان مثال، در زیر ما یک تابع ساده ایجاد کرده ایم که عبارت It's Monday را چاپ می کند:

```
:def print_Monday ()
```

```
print ("It's Monday!")
```

اما زمانی که این کد اجرا شود، هیچ اتفاقی نمی افتد. به این دلیل که برای اجرا می بایست تابع فراخوانی شود. برای انجام این کار، می توانیم تابع خود را به این صورت فراخوانی کنیم:

```
:def print_Monday ()
```

```
print ("It's Monday!")
```

```
print_Monday ()
```

با اجرای کد بالا، مقدار زیر در خروجی نمایش داده می شود:

```
It's Monday!
```

برای درک بهتر در ادامه اجزای اصلی تابع تجزیه و به هر بخش به صورت مجزا خواهیم پرداخت:

۱. کلمه کلیدی `def` در توابع پایتون نشان دهنده این است که می خواهیم یک تابع ایجاد کنیم.

۲. `print_Monday` نام تابع ما است. این نام باید منحصر به فرد باشد.

۳. پرانتز `()` جایی است که پارامترهای تابع دریافت می شوند. به عنوان مثال تابع محاسبه سن باید سال تولد فرد را دریافت کند که از طریق یک متغیر داخل این پرانتز انجام می شود.

۴. دو نقطه `:` انتهای هدر (Header) یا بخش تعریف تابع ما را مشخص می کند.

## #۴ انواع فراخوانی توابع پایتون

همان طور که گفتیم برای تعریف یک تابع فقط کافیست یک نام به آن اختصاص دهید. سپس پارامترهایی را که باید در تابع گنجانده شوند مشخص کرده و بلوک های کد (بخش های مختلف کد به صورت سلسله مراتبی) را ساختاردهی کنید.

هنگامی که ساختار اصلی یک تابع نهایی شد، می توانید با فراخوانی آن از یک تابع دیگر یا فراخوانی مستقیم، آن را اجرا کنید. در زیر مثالی برای فراخوانی تابع `printme ()` به عنوان یکی از توابع پایتون به صورت مستقیم آورده شده است.

```
#!/usr/bin/python

# Function definition is here

:def printme (str)

    "This prints a passed string into this function"

        print str

        return;

# Now you can call printme function

Printme ("I'm first call to user defined function!")

Printme ("Again second call to the same function")
```

هنگامی که کد بالا اجرا می شود، نتیجه زیر نمایش داده می شود:

```
I'm first call to user defined function!

Again second call to the same function
```



در زیر مثالی برای فراخوانی تابع در تابعی دیگر آورده شده است.

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

```
# Function definition is here
```

```
:def printme (str)
```

```
"This prints a passed string into this function"
```

```
    print str
```

```
    return;
```

```
# second Function call first inside
```

```
:def caller_function (str+)
```

"String+"

Printme ("Calling function!")

print str+

return;

caller\_function (" Second function text..!")

با اجرای کد بالا در توابع پایتون محتوای زیر نمایش داده می شود:

Calling function! Second function text...

در کد بالا عبارت Calling function! مربوط به تابع اول و عبارت Second function text .. مربوط به تابع دوم (تابع فراخوانی) هستند.