

# FUNCTIONS

- توابع (Functions) بلوکی از کدهای مرتب شده می باشد که قابلیت استفاده مجدد را دارد.
- برخی از نکاتی که می بایست حین تعریف توابع در نظر گرفته شوند:
- با def شروع شده و سپس اسم تابع، پارنتز و : قرار می گیرد.
- پارانتز می تواند در برگیرنده پارامترها و آرگومان ها باشد. همچنین می توان پارامترها را در داخل پارانتز تعریف کرد
- اولین عبارت تابع اختیاری بوده و توضیحاتی را در رابطه با تابع ارائه می کند.
- در نهایت توسط دستور return، عبارتی به کاربر برگردانده می شود.
- برای فراخوانی تابع، ابتدا نام تابع و سپس ( ) قرار می گیرد



```
def functionname( parameters ):
    "function_docstring"
    function_suite
    return [expression]
```

# FUNCTIONS

- می توان اطلاعاتی را در قالب آرگومان ها در اختیار توابع قرار داد. آرگومان ها در داخل پارانتز بعد از اسم تابع تعریف شده و می توان آرگومان های متعددی را با جداسازی توسط کاما , تعریف کرد.
- تفاوت آرگومان و پارامتر: از منظری هر دو به یک معنی می باشند و در واقع اطلاعاتی می باشند که به تابع معرفی می شود اما به طور جزئی تر پارامتر ها هنگام تعریف تابع و آرگومان ها هنگام فراخوانی توابع مورد استفاده قرار می گیرند.
- برخی از انواع آرگومان های توابع:
  - آرگومان های ضروری
  - آرگومان های کلید واژه ای
  - آرگومان های پیش فرض

# FUNCTIONS

## ARGUMENT TYPES

- آرگومان های ضروری می بایست هنگام فراخوانی تابع با ترتیب مشخص شده حین تعریف تابع در اختیار تابع قرار بگیرند.
- آرگومان های کلید واژه ای حین فراخوانی مورد استفاده قرار گرفته و مقدار مربوطه همراه با کلید واژه مشخص به تابع معرفی شود.
- آرگومان های پیشفرض، آرگومان هایی هستند که در صورتی که حین فراخوانی تابع مقداری برای آن ها معرفی نشود، مقدار پیش فرض تعریف شده خود را اتخاذ می کنند.
- عبارت `return [expression]` منجر به خروج از تابع شده و به طور اختیاری می تواند عبارتی را به کاربر بازگرداند. در صورت استفاده از `return` و مشخص نکردن عبارتی که می بایست بازگردانده شود، `None` به کاربر بازگردانده می شود.
- توابع ناشناس (The Anonymous Functions) توابعی هستند که به طرق مرسوم و با استفاده از کلیدواژه `def` تعریف نمی شوند؛ بلکه تعریف آن ها با استفاده از کلید واژه `lambda` صورت می پذیرد. این توابع می تواند هر تعداد آرگومانی را بپذیرند اما تنها یک خروجی می توانند داشته باشند.



```
lambda [arg1 [,arg2,...,argn]]:expression
```