



حل تمرین پایتون

جلسه سوم

سونیا ابطی

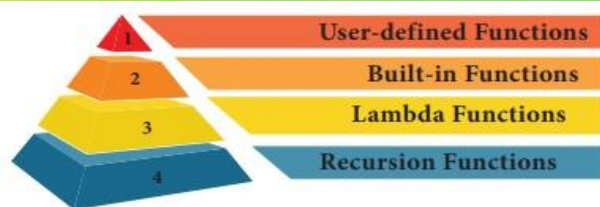


Figure – 7.1 – Types of Python Functions

Functions	Description
User-defined functions	Functions defined by the users themselves.
Built-in functions	Functions that are inbuilt with in Python.
Lambda functions	Functions that are anonymous un-named function.
Recursion functions	Functions that calls itself is known as recursive.

Table – 7.1 – Python Functions and it's Description

User-defined

FUNCTION PARTS

```
def vol_of_sphere(radius):  
    vol = 4/3.0*math.pi*radius**3  
    return vol
```

Function definition

Argument(sent in)
radius is called a
parameter.

←header

} body

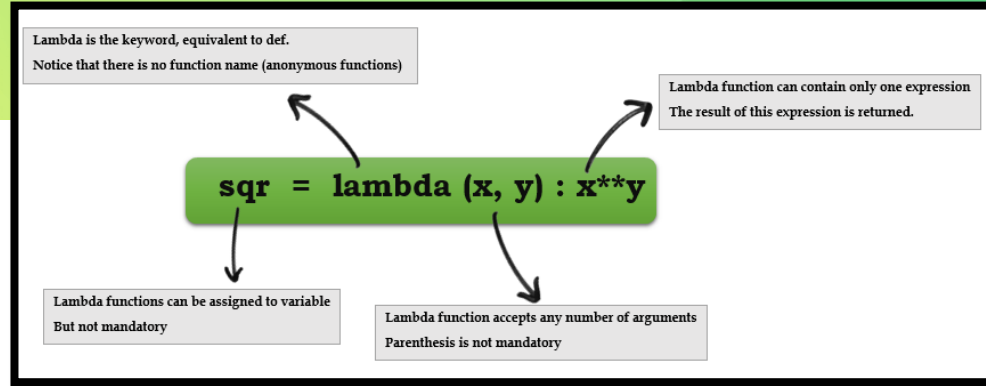
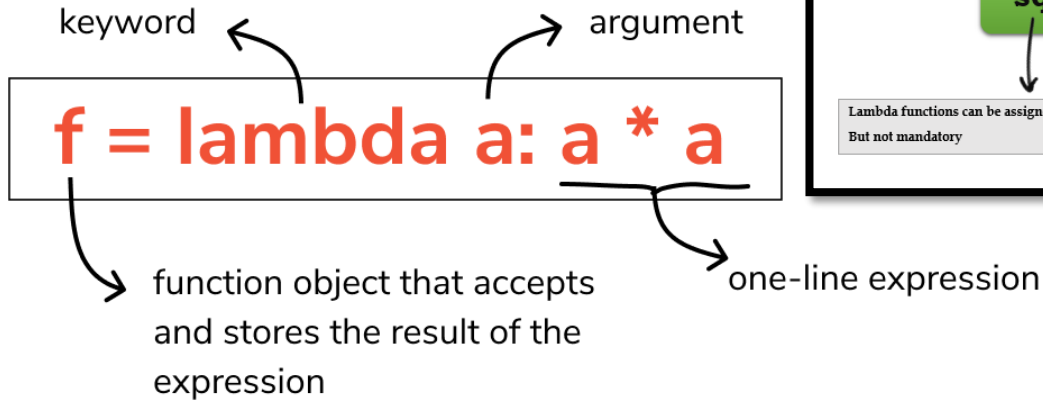
Value to return

Use four spaces for the indentation

Built-in

		Built-in Functions		
<code>abs()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>all()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>any()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>ascii()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bin()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>bool()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	
<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>	

Lambda



Recursion

Python Recursion

```
def fact(n):  
    ...  
    ...  
    return (n*fact(n-1))
```

trytoprogram.com

```
print(factorial(4))  
def factorial(4):  
    if 4 > 1:  
        return 4 * factorial(4 - 1)  
    else:  
        return 1  
def factorial(3):  
    if 3 > 1:  
        return 3 * factorial(3 - 1)  
    else:  
        return 1  
def factorial(2):  
    if 2 > 1:  
        return 2 * factorial(2 - 1)  
    else:  
        return 1  
def factorial(1):  
    if 1 > 1:  
        return 1 * factorial(1 - 1)  
    else:  
        return 1
```

factorial(4)=4*3*2*1

factorial(4-1)=3*2*1

factorial(3-1)=2*1

factorial(2-1)=1

○ ○ ○

```
my_list=[int,float,bool,str,list,tuple,dict]
```

مثال اول

```
def test(x):  
    a=7  
    print(test(5))
```


مثال دوم

تابعی بنویسید که تعداد حروف غیر یکسان دو رشته DNA زیر را بیابد.

```
GAGCCTACTAACGGGAT  
CATCGTAATGACGGCCT
```

مثال سوم

برنامه‌ای بنویسید که عدد دلخواهی را از کاربر دریافت کند و تمام اعداد موجود در بازه عدد ورودی را که با معکوس خود برابر هستند، در یک لیست ذخیره و چاپ نماید.

مثال چهارم

تابعی بنویسید که بگوید آیا عددی کوچکتر از ۱۰۰ وجود دارد که فاکتوریلش برابر معکوسش باشد یا خیر؟

مثال پنجم

تابعی بنویسید که یک ماتریس و یک عدد را از ورودی دریافت کند. اگر عدد مورد نظر در آن ماتریس قرار داشت، شماره سطر و ستون آن ماتریس را در یک لیست برگرداند.

مثال ششم

تابعی بنویسید که از کاربر بپرسد که یک ماتریس چند در چند می خواهد وارد کند.
سپس ماتریس را بخوانید و مقدار ماکزیمم هر سطر را برگردانید.

مثال هفتم

برنامه‌ای بنویسید که دو لیست را از ورودی بگیرد و خانه‌هایی که در دو لیست مقدار یکسان دارد را برگرداند.

مثال هشتم

تابعی بنویسید که یک لیست ورودی را دریافت کند و لیست جدیدی را که شامل تنها عناصر منحصر به فرد باشد را برگرداند. عناصر تکراری فقط یکبار باید در لیست جدید وجود داشته باشند و از چند عنصر تکراری، اولین عنصر در خروجی نمایش داده شود. ترتیب عناصر باید به شکل لیست اصلی محفوظ باشد.

مثال نهم

تابعی بنویسید که مجموع اعضای یک لیست دلخواه را به صورت بازگشتی محاسبه نماید



ددلاین پروژه‌ی تحلیل و طراحی الگوریتم سونیا دو روز دیگر است و او وقت ندارد در حل این سوال آسان به دوستش کمک کند. پس از شما می‌خواهد به او کمک کنید و برنامه‌ای بنویسید که ابتدا دو عدد m و n و دو ماتریس $n \times m$ با نام‌های $M1$ و $M2$ را از ورودی گرفته و سپس با گرفتن دو عدد A و B ، در خروجی ماتریس $AM1+BM2$ را چاپ کند.

ورودی

در خط اول ورودی دو عدد حسابی n و m با فاصله از هم آمده است.

$$0 \leq n, m \leq 20$$

و سپس دوبار $n \times m$ عدد طبیعی به عنوان درایه های دو ماتریس $M1$ و $M2$ داده می شوند و در اخر دو عدد حسابی A و B داده می شوند.

$$0 \leq A, B \leq 20$$

خروجی

خروجی برنامه‌ی شما باید یک ماتریس $n \times m$ باشد.

ورودی نمونه

```
2 3
1 2 1
2 2 1
1 2 3
1 2 3
1 3
```

Copy

خروجی نمونه

```
4 8 10
5 8 10
```

Thank You.

