

# برنامه سازی پیشرفته (توابع: معرفی و کاربرد)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

[eskandari@guilan.ac.ir](mailto:eskandari@guilan.ac.ir)

# یادآوری و مرور .... انواع داده (data type)

یک نوع داده عبارت است از مجموعه ای از مقادیر به همراه مجموعه ای از عملگرها  
بر روی آن مقادیر

Data type = set of values (domain) + set of operators

Integer =  $Z + \{+, *, /, \dots\}$

انواع داده درون ساخت (Built-in data type)  
Numbers, Strings, Lists, Dictionaries, Tuples, Files, Sets,

انواع داده کلاسی

Student, Teacher, Car, TV, ....

انواع داده در پایتون

# یادآوری و مرور .... ساختارهای کنترلی

یک ساختار کنترلی، دستوری است که روند اجرای سایر دستورالعمل های برنامه را مشخص می کند.

if

← تصمیم (Decision)

while, for

← تکرار (Repetition)

ساختارهای کنترلی

break, continue, return

← پرش (Jump)

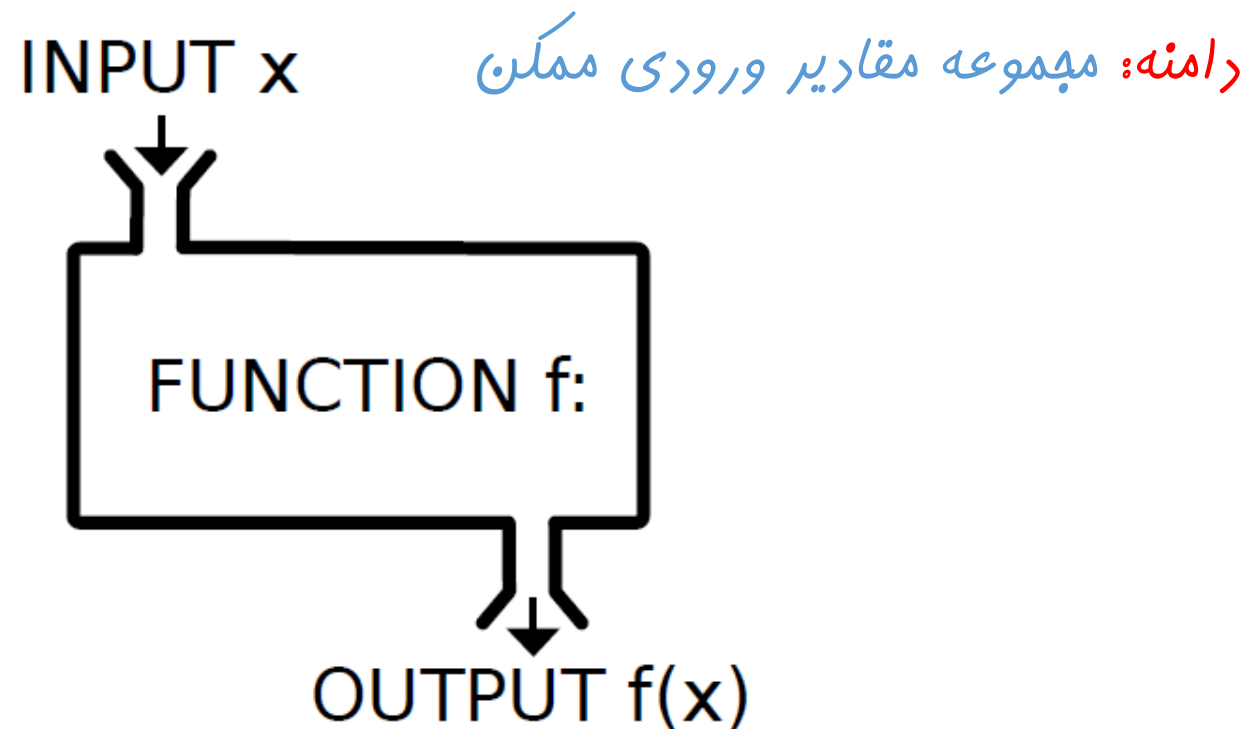
# یادآوری و مرور ....

## Collections

نام کُلکسیون	نوع داده	تغییرپذیر؟	دسترسی به عناصر
string	متنی	✗	اندیس (راست و چپ)
list	هر نوع داده ای	✓	اندیس (راست و چپ)
Dictionary	هر نوع داده ای	✓	کلید

## تابع: تعاریف

هر دستگاهی که یک ورودی را دریافت کرده و بر روی آن عملیاتی انجام داده و یک خروجی تولید کند.



شرط اساسی: برای ورودی های یکسان، خروجی های یکسان تولید کند.

$$\forall x_1, x_2: \text{if } (x_1 == x_2) \text{ then } f(x_1) == f(x_2)$$

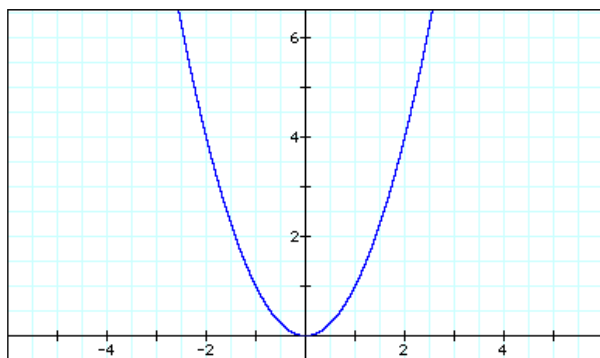
برده: مجموعه مقادیر خروجی ممکن

# تابع: تعاریف

روش های مرسوم نمایش یک تابع

زوج مرتب  $\{(1,1), (2,4), (3,9), (4,16), \dots\}$

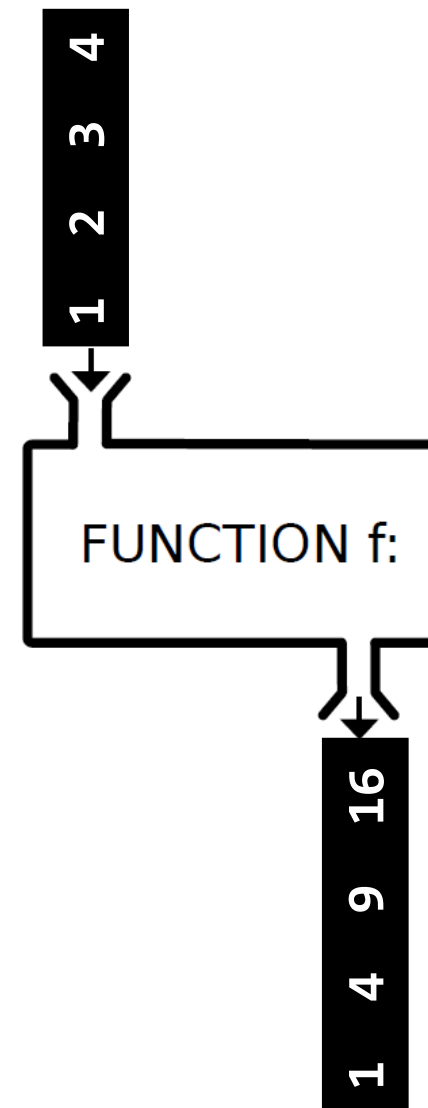
نمایش جبری  $f(x) = x^2$



نمودار

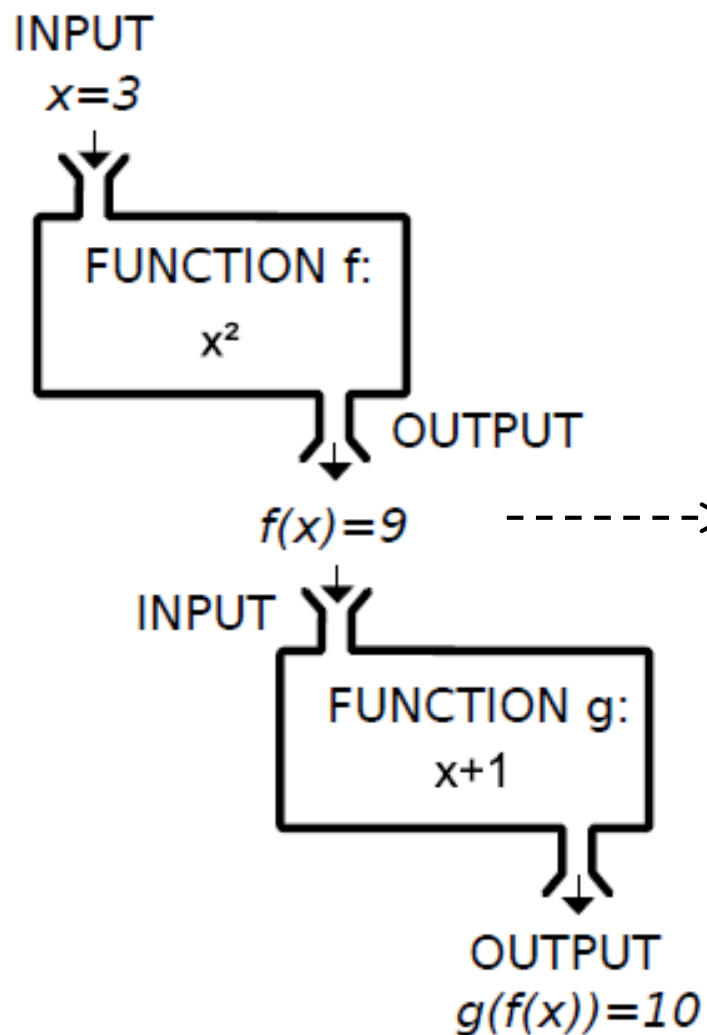
```
def f(x):  
    return x * x
```

ماژول



# تابع: تعاریف

ترکیب توابع: ورودی یک تابع، خروجی تابعی دیگر باشد.



باید خروجی تابع  $f$ ، عضوی از دامنه تابع  $g$  باشد.

$g(f(x))$

## تابع در پایتون

در زبانهای برنامه نویسی از توابع به دو منظور استفاده می شود:

۱- ماژولار سازی برنامه و افزایش سافت یافتگی

به جای حل یک مسئله پیچیده، آن را به چند مسئله ساده تبدیل کرده و آنها را به شکل مجزا حل می کنیم (برای هر مسئله، یک تابع ایجاد می کنیم). در نهایت، مسئله اصلی از طریق ترکیب توابع ساده حل می شود.

۲- کاهش افزونگی کد در برنامه

برای یک مسئله، یک تابع نوشته می شود و در هر جا و به هر تعداد که نیاز باشد، فراخوانی می شود.



# تابع در پایتون

ساختار کلی توابع در زبان پایتون

```
def <name>(arg1, arg2, ... argN):  
    ...  
    return <value>
```

← امضای تابع

→ برنه تابع

# تابع در پایتون

ساختار کلی توابع در زبان پایتون

نام تابع: باید از قوانین نامگذاری شناسه ها تبعیت کند. بهتر است متناسب با کارکرد تابع انتخاب شود.

```
def <name>(arg1, arg2, ... argN):  
    ...  
    return <value>
```

مقدار بازگشتی تابع: `return` یک ساختار پرش می باشد. هر زمان درون تابع، دستور `return` اجرا شود، اجرای تابع فاتمه یافته و نتایج آن برگردانده می شود. یک تابع می تواند فاقد مقدار بازگشتی باشد. علاوه بر این، در پایتون یک تابع می تواند بیش از یک مقدار را برگرداند.

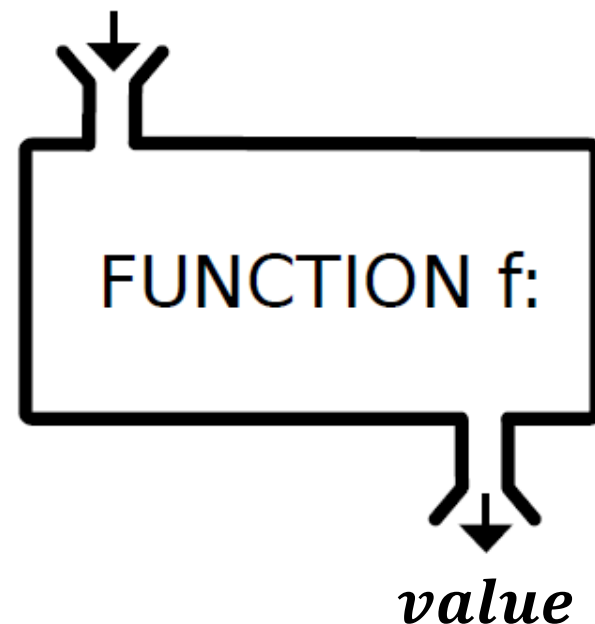
آرگومانها یا پارامترهای تابع: ورودی های ممکن تابع را مشخص می کنند. تابع می تواند فاقد آرگومان باشد. هر کدی، در زمان استفاده از تابع باید آرگومانهای تابع را مقداردهی کند.

# تابع در پایتون

ساختار کلی توابع در زبان پایتون

```
def <name>(arg1, arg2, ... argN):  
    ...  
    return <value>
```

*arg1, arg2, ..., argN*



راهکار بدون استفاده از تابع

## تابع در پایتون

مثال: برنامه ای

بنویسید که سه مقدار را  
از ورودی دریافت کرده  
و تعداد ارقام هر یک را  
چاپ کند.

```
1 a = int(input('enter first number: '))
2 b = int(input('enter second number: '))
3 c = int(input('enter third number: '))
4
5 nod = 1
6 while a >= 10:
7     a = a // 10
8     nod = nod + 1
9 print(nod)
10 nod = 1
11 while b >= 10:
12     b = b // 10
13     nod = nod + 1
14 print(nod)
15 nod = 1
16 while c >= 10:
17     c = c // 10
18     nod = nod + 1
19 print(nod)
```

→ مناسبه تعداد ارقام a

→ مناسبه تعداد ارقام b

→ مناسبه تعداد ارقام c

راهکار با استفاده از تابع

## تابع در پایتون

```
1 def num_of_digits(n):
2     nod = 1
3     while n >= 10:
4         n = n // 10
5         nod = nod + 1
6     return nod
7
8 a = int(input('enter first number: '))
9 b = int(input('enter second number: '))
10 c = int(input('enter third number: '))
11
12 nod1 = num_of_digits(a)
13 nod2 = num_of_digits(b)
14 nod3 = num_of_digits(c)
15
16 print(nod1)
17 print(nod2)
18 print(nod3)
```

$n = a$

$n = b$

$n = c$

مثال: برنامه ای  
بنویسید که سه مقدار را  
از ورودی دریافت کرده  
و تعداد ارقام هر یک را  
چاپ کند.

## تابع در پایتون

مثال: برنامه ای بنویسید که مجموع تعداد ارقام کلیه اعداد مابین ۲۳۵۰ و ۱۲۸۴۰ را محاسبه و چاپ کند.

```
1 def num_of_digits(n):
2     nod = 1
3     while n >= 10:
4         n = n // 10
5         nod = nod + 1
6     return nod
7
8 sum = 0
9 for i in range(2350, 12841):
10     sum += num_of_digits(i)
11
12 print(sum)
```

44805

## تابع در پایتون

مثال: برنامه ای بنویسید که  
مجموع تعداد ارقام کلیه اعداد  
اول مابین ۲۳۵۰ و ۱۲۸۴۰ را  
محاسبه و چاپ کند.

```
1 import math
2 def is_prime(n):
3     for i in range(2, int(math.sqrt(n))+1):
4         if n%i == 0:
5             return False
6     return True
7
8 def num_of_digits(n):
9     nod = 1
10    while n >= 10:
11        n = n // 10
12        nod = nod + 1
13    return nod
14
15 sum = 0
16 for i in range(2350, 12841):
17     if(is_prime(i)):
18         sum += num_of_digits(i)
19 print(sum)
20
```

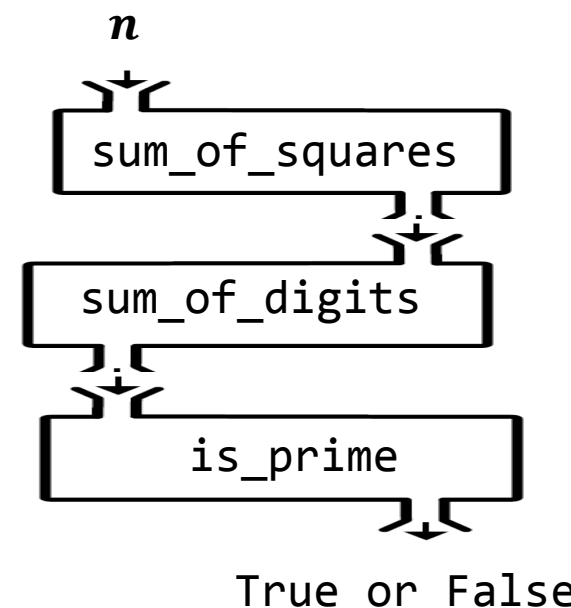
## تابع در پایتون

مثال: برنامه ای بنویسید که عدد صحیح  $n$  را از ورودی دریافت کرده و اول بودن مجموع ارقام عبارت زیر را محاسبه کند:

$$1 + 2 + 4 + 9 + \dots + n^2$$

```
1 import math
2 # a function to calculate 1+4+9+ ... + n**2
3 def sum_of_squares(n):
4     sum = 0
5     for i in range(1,n+1):
6         sum += i**2
7     return sum
8
9 # a function to calculate sum of digits
10 def sum_of_digits(n):
11     sum = 0
12     while(n != 0):
13         sum += n%10
14         n //= 10
15     return sum
16
17 def is_prime(n):
18     for i in range(2, int(math.sqrt(n))+1):
19         if n%i == 0:
20             return False
21     return True
22
23 n = int(input('enter an integer: '))
24 print(sum_of_squares(n))
25 print(sum_of_digits(sum_of_squares(n)))
26 print(is_prime(sum_of_digits(sum_of_squares(n))))
```

```
enter an integer: 123
627874
34
False
```





## تابع در پایتون

تمرین: برنامه ای بنویسید که عدد صحیح  $n$  را از ورودی دریافت کرده و اول بودن مجموع ارقام کلیه اعداد اول کوچکتر از آن را بررسی کند.

تمرین: برنامه ای بنویسید که عدد صحیح  $n$  را از ورودی دریافت کرده و حاصل مجموع زیر را محاسبه و چاپ کند.

$$\frac{1}{2} + \frac{1+3}{2} + \frac{1+3}{2+4} + \frac{1+3+5}{2+4} + \dots + \frac{1+3+\dots+(2n+1)}{2+4+\dots+(2n)}$$

تمرین: برنامه ای بنویسید که عدد صحیح  $n$  را از ورودی دریافت کرده و مجموع ارقام کلیه اعداد اول  $n$  جمله ابتدائی دنباله فیبوناچی را محاسبه و چاپ کند.

## تابع در پایتون

مثال: برنامه ای بنویسید که عدد صحیح  $n$  را از ورودی دریافت کرده و لیستی از اعداد اول کوچکتر از  $n$  و نیز لیستی از مجموع ارقام اعداد اول کوچکتر از  $n$  را ایجاد کرده و چاپ کند.

```
1 import math
2 def is_prime(n):
3     for i in range(2, int(math.sqrt(n))+1):
4         if n%i == 0:
5             return False
6     return True
7
8 def sum_of_digits(n):
9     sum = 0
10    while(n != 0):
11        sum += n%10
12        n //= 10
13    return sum
14
15 n = int(input('enter a number:'))
16 lst1 = [i for i in range(2,n) if is_prime(i)]
17 lst2 = [sum_of_digits(i) for i in range(2,n) if is_prime(i)]
18 print(lst1)
19 print(lst2)
```

enter a number:30

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

[2, 3, 5, 7, 2, 4, 8, 10, 5, 11]