python

Function

در این اسلایدها به مبحث تابع‌نویسی در پایتون می‌پردازیم. شامل:

- Function Definition
- Function Calling

Parameters & Arguments:

- Positional Arguments
- Default Parameters
- Asterisk Parameters
- Keyword Arguments

➢ venitian_masks

https://www.linkedin.com/in/mina-firoozgohar/
www.misgohar.blog.ir
www.misgohar.blogfa.com

تابع چیست؟ و چرایی؟

- تابع یک مفهوم ریاضی است که یک عملی را بر روی ورودی/ورودی‌هایی انجام می دهد و خروجی را برمی گرداند.
- تابع در برنامه نویسی با الهام از همان مفهوم ریاضیاتی، یک نوع ساختار برای گروه‌بندی مجموعه دستوراتی هست که بر روی ورودی/ورودی‌هایی انجام می شوند مزایا:
- جلوگیری از افزونگی کدها و خوانایی و سادگی
- قابلیت استفاده مجدد بدون نوشتن مجدد
- نگهداری، توسعه و پشتیبانی راحت تر و کم‌هزینه‌تر

Redundancy

```
print("Program starts")

print("Hi Peter")
print("Nice to see you again!")
print("Enjoy our video!")

# some lines of codes
the_answer = 42

print("Hi Sarah")
print("Nice to see you again!")
print("Enjoy our video!")

width, length = 3, 4
area = width * length

print("Hi Dominque")
print("Nice to see you again!")
```

```
print("Program starts")

print("Hi Peter")
print("Nice to see you again!")
print("Enjoy our video!")

# some lines of codes
the_answer = 42

print("Hi Sarah")
print("Nice to see you again!")
print("Enjoy our video!")

just_some_code = "whatever"
another_var = "whatever you do"

print("Hi Dominque")
print("Nice to see you again!")
print("Enjoy our video!")
```

Output:

```
Program starts
Hi Peter
Nice to see you again!
Enjoy our video!
Hi Sarah
Nice to see you again!
Enjoy our video!
Hi Dominque
Nice to see you again!
Enjoy our video!
```

در چندین جا از برنامه از یک دستور مشخص برای اسم های متفاوت استفاده شده که می توانیم با نوشتن یک تابع برای این منظور که ورودی (اسم موردنظر) را دریافت می کند و دستور را انجام میدهد از افزونگی دستورات جلوگیری کنیم .

Projects & Teaching

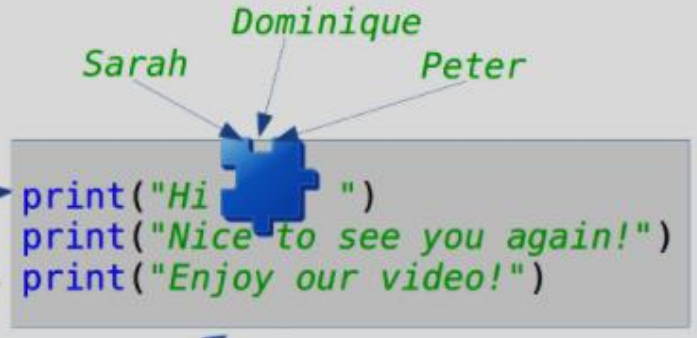
https://t.me/misgohar1

09125783861

www.misgohar.blogfa.com

simple function

```
print("Program starts")

print("Hi Peter")
print("Nice to see you again!")
print("Enjoy our video!")

# some lines of codes
the_answer = 42

print("Hi Sarah")
print("Nice to see you again!")
print("Enjoy our video!")

width, length = 3, 4
area = width * length

print("Hi Dominque")
print("Nice to see you again!")
print("Enjoy our video!")
```




```
Output:
Program starts
Hi Peter
Nice to see you again!
Enjoy our video!
Hi Sarah
Nice to see you again!
Enjoy our video!
Hi Dominque
Nice to see you again!
Enjoy our video!
```

```
def greet(name):
    print("Hi " + name)
    print("Nice to see you again!")
    print("Enjoy our video!")


print("Program starts")

greet("Peter")


# some lines of codes
the_answer = 42

greet("Sarah")

width, length = 3, 4
area = width * length
```

Syntax of Function

- برای استفاده از تابع در برنامه نیاز به دو مرحله داریم:
  1. نوشتن تعریف تابع
  2. فراخوانی تابع
- ساختار نوشتن تعریف تابع:

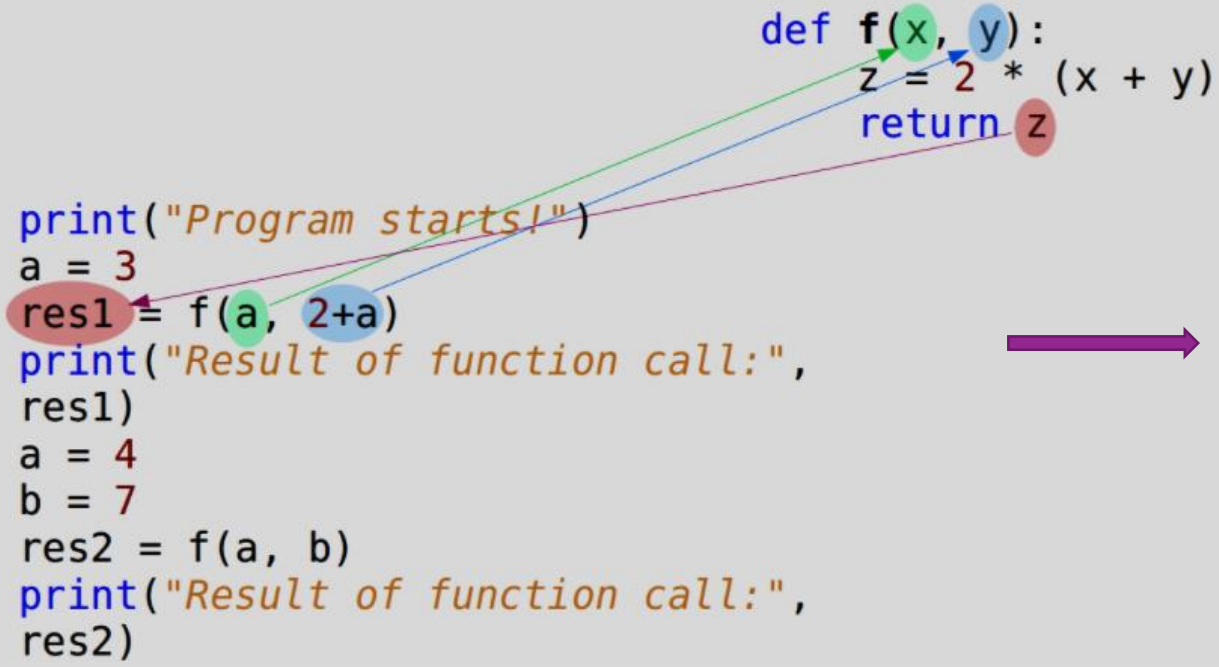
```
def function-name(Parameter list):
    statements, i.e. the function body
```

- لیست پارامترها می تواند خالی یا چند پارامتر داشته باشد.
- فراخوانی تابع:

```
function-name(Argument list)
```

- به ورودی هایی که هنگام فراخوانی داخل پرانتز قرار میگیرد آرگومان می‌گوییم . آرگومانها در فراخوانی همان پارامترها در تعریف هستند .هر زمان که فراخوانی شود دستورات بدنه تابع اجرا می شود .درنتیجه تا قبل از فراخوانی دستورات تابع مانند یک جسم بی‌جان است و درون حافظه اجرا وجود ندارد.

simple function 2




```
def f(x, y):
    z = 2 * (x + y)
    return z


print("Program starts!")
a = 3
res1 = f(a, 2+a)
print("Result of function call:", res1)
a = 4
b = 7
res2 = f(a, b)
print("Result of function call:", res2)
```

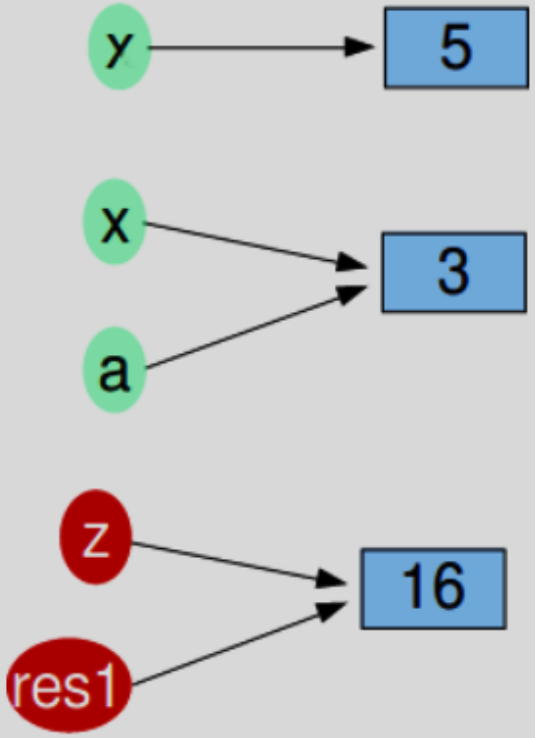
```
Output:
Program starts!
Result of function call: 16
Result of function call: 22
```


Projects & Teaching

https://t.me/misgohar1

09125783861

www.misgohar.blogfa.com



variables

- گفتیم متغیرها در پایتون به خانه های حافظه اشاره می کنند .درنتیجه شکل پارامترها و متغیرهای کد به صورت روبرو:

```
def f(x, y):
    z = 2 * (x + y)
    return z

print("Program starts!")
a = 3
res1 = f(a, 2+a)
print("Result of function call:",
res1)
a = 4
b = 7
res2 = f(a, b)
print("Result of function call:",
res2)
```

Function for convert the temperature

```
def fahrenheit(T_in_celsius):
    """ returns the temperature in degrees Fahrenheit
    """
    return (T_in_celsius * 9 / 5) + 32

for t in (22.6, 25.8, 27.3, 29.8):
    print(t, ": ", fahrenheit(t))
```

Use of docstring

```
help(fahrenheit)
```

Output:

```
Help on function fahrenheit in module __main__:
fahrenheit(T_in_celsius)
    returns the temperature in degrees Fahrenheit
```

Output:

```
22.6 :  72.68
25.8 :  78.44
27.3 :  81.14
29.8 :  85.64
```

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

Practice(BMI)

- شاخص تراکم بدن :

BMI = weight(kg) / (height(m)**2)

WHO CLASSIFICATION OF WEIGHT STATUS	
WEIGHT STATUS	**BODY MASS INDEX (BMI), kg/m^2**
Underweight	<18.5
Normal range	18.5 – 24.9
Overweight	25.0 – 29.9
Obese	≥ 30
Obese class I	30.0 – 34.9
Obese class II	35.0 – 39.9
Obese class III	≥ 40

```
def BMI(weight, height):
    """ calculates the BMI where
        weight is in kg and height in metres"""
    return weight / height**2


height = float(input("What is your height? "))
weight = float(input("What is your weight? "))

bmi = weight / height ** 2
print(bmi)
if bmi < 15:
    print("Very severely underweight")
elif bmi < 16:
    print("Severely underweight")
elif bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal (healthy weight)")
elif bmi < 30:
```

- نکته :چون ساختار if وelif به گونه ای ست که هر کدام از شرط ها برقرار باشد دستور مربوطه اجرا می شود و دیگر بقیه شرط ها بررسی نمی شوند درنتیجه می توانیم تنها شرط کوچک تر بودن را بررسی از کران بالا را در شرط بنویسیم.

Default arguments/ parameters

- می توانیم یک مقدار پیش فرض برای پارامترهای تابع تعیین کنیم .در این صورت چنانچه هنگام فراخوانی تابع هیچ آرگومانی برای آن پارامتر ارسال نشود آنگاه، مقدار پیش فرض برای آن تنظیم می شود .به اصطلاح »پارامتر/آرگومان اختیاری «نیز می گویند و به پارامترهایی که پیش فرض نیستند یا همان positional argument ها «پارامتر/آرگومان اجباری» گفته می شود.
- در پایتون mutable objectها در پارامترها می توانند دردسرساز یا شگفت انگیز باشند .به این صورت:
  - ❖پارامترهای پیش فرض تنها یک بار ایجاد می شوند و آن هم در زمان (compileیعنی تعریف تابع(
  - ❖شیءهای قابل تغییر آنهایی هستند که بعد از ایجاد امکان تغییر در همان فضای حافظه را دارند .
- درنتیجه در هر بار فراخوانی تابع پارامتر جدید ایجاد نمی شود بلکه مقدار آن به روز می شود.

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

```python
class Robot:
    __counter = 0

    def __init__(self):
        type(self).__counter += 1

    @staticmethod
    def RobotInstances():
        return Robot.__counter

if __name__ == "__main__":
    print(Robot.RobotInstances())
    x = Robot()
    print(x.RobotInstances())
    y = Robot()
    print(y.RobotInstances())
    print(x.RobotInstances())
```

Output:

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

Default Parameters

Default Parameter

```
def addUser(name,family,gender,language='Eng'):
    print('adding: {} {}, gender= {}, language= \
        {}.format(name,family,gender,language))

# Function Calling
addUser('MIS',gender='female',family='Gohar')
```

Output: Adding MIS Gohar, gender= female, language= Fr

```
addUser('MIS',language='Fr',family='Gohar',gender='female)
```

Output:
```
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    addUser('MIS',language='Fr',gender='female',family='Gohar')
TypeError: addUser() got an unexpected keyword argument 'language'
```

```
addUser('MIS','Gohar',gender='female')
```

Output: Adding MIS Gohar, gender= female, language= Eng

نکته‌های مهم :

در تعریف تابع، پارامترهای پیش‌فرض حتماً باید در انتهای پارامترها باشند. در این صورت هنگام اجرا برنامه می‌داند که اگر بعد از positional argumentها از هر جا که دیگر آرگومانی وارد نشود از آنجا باقی پارامترها با مقدار پیش‌فرض جایگزین شوند.

می‌توانیم ترتیب نوشتن positional argumentها را به هم بزنیم در صورتی که نام متغیر درنظر گرفته‌شده برای آن هنگام تعریف تابع را بنویسیم. اما نمی‌توانیم پارامتر پیش‌فرض را قبل از positionalها بیاوریم.

Projects & Teaching

https://t.me/misgohar1

09125783861

www.misgohar.blogfa.com

Default Parameters

Default Parameter

```
def addUser(name,family,language='Eng', gender):
    print('adding: {} {}, gender= {}, language= \
        {}.format(name,family,gender,language))
```

```
SyntaxError: non-default argument follows default argument
```

نکته‌های مهم :

در تعریف تابع، پارامترهای پیش‌فرض حتماً باید در انتهای پارامترها باشند.

اگر این قانون را رعایت نکنیم، قبل از فراخوانی تابع خطای دستوری دریافت می‌کنیم.

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

خطای منطقی مهم در پارامترهای پیش‌فرض

پارامترهای پیش‌فرض برخلاف دیگر پارامترها و متغیرهای تابع compile time هستند و هنگام کامپایل برنامه ایجاد می‌شوند و نه هنگام اجرای فراخوانی. این نکته اگر در نظر گرفته نشود ممکن است خطا ایجاد کند.

اگر پارامتر پیش‌فرض یک مقدار قابل تغییر به عنوان پیش‌فرض داشته باشد آنگاه زمان کامپایل یک نمونه از آن ایجاد می‌شود درنتیجه با هربار فراخوانی همان خانه از حافظه تغییرات داخل تابع را می‌پذیرد. که اگر عملگرهای به‌روزرسانی بر روی آن اتفاق بیفتد در هر اجرا آن حافظه با مقدار قبلی حاصل از اجرای قبلی به‌روزرسانی می‌شود و با بیرون آمدن برنامه از تابع آن حافظه مانند دیگر متغیرهای محلی نخواهد بود و از حافظه پاک نمی‌شود.

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

Logical Error by Default Parameters

```
def average(name,family,n_s, scores=[]):
    if scores == []:
        for i in range(n_s):
            s = input('Enter scores: ')
            scores.append(s)
    avg = sum(scores) / n_s
    print("{} {}'s average is {}".format(name,family,avg))


# Function Calling
1) average('Ali','Hasani',5)


2) average('maryam','ghomi',7)
```

Output:

```
Enter score: 15
Enter score: 12
Enter score: 17
Enter score: 16.5
Enter score: 19
Ali Hasani's average is 15.9
```

Output:

```
maryam ghomi's average is 11.357142857142858
```

توضیح خطا :

در دومین اجرای تابع با اینکه نمرات دانش‌آموز داده نشده اما scores آرگومان پیش‌فرض هست که چون از اجرای قبلی به‌روزرسانی شده درنتیجه خالی نیست و شرط if scores == [] برقرار نمی‌شود و مجموع نمرات داخل scores تقسیم بر تعداد 7 می‌شود. (امتحان کنید)

Projects & Teaching

https://t.me/misgohar1

09125783861

www.misgohar.blogfa.com

Keyword Arguments

- هنگام فراخوانی تابع لازم هست که هم بدانیم چه آرگومان‌هایی باید ارسال کنیم و هم ترتیب آرگومان‌ها را بدانیم و رعایت کنیم. این یک مشکل است که نمی‌توان فقط به ادیتور کد برای حل آن اکتفا کرد.
- راه حل پایتون برای حل این مشکل: Keyword arguments
- پارامترهای keyword در تعریف تابع با ** قبل از نام آن مشخص می‌شوند.

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com

Keyword Parameters

Keyword Parameters

```
def addUser(name,**kwargs):
    print('adding: {} {}, gender= {}, language= \
        {}.format(name,family,gender,language))

# Function Calling
1) addUser('MIS',gender='female',family='Gohar',language='Fr')
```

Output: Adding MIS Gohar, gender= female, language= Fr

2) addUser('MIS','Gohar')

Output:

```
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    addUser('Mina','Gohar')
TypeError: addUser() takes 1 positional argument but 2 were given
```

3) addUser(family='Gohar',language='Fr')

Output:

```
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    addUser(family='Gohar',language='Fr')
TypeError: addUser() missing 1 required positional argument: 'name'
```

نکته‌های مهم :

نمی‌توانیم positional argument ها را بعد از keyword argument بیاوریم.

همه پارامترهای اجباری در تعریف تابع باید قبل از keyword parameters باشند.

Keyword argumentها ترتیب خاصی ندارند.

در هنگام فراخوانی نیز حتما باید ابتدا positional argument و سپس keyword argumentها را یا به همراه نام متغیر باید ارسال کنیم و یا متغیرها و مقادیر آنها را در یک دیکشنری قرار داده و دیکشنری را با ** قبل از خود به تابع ارسال کنیم.

Projects & Teaching
https://t.me/misgohar1
09125783861
www.misgohar.blogfa.com