

متغیر های لوکال و گلوبال

قبل از اینکه درباره این موضوع بحث کنیم باید درباره اسکوپ ها یاد آوری کنیم :

اسکوپ : زاویه دید نسبی کد در حال اجرا به محیط پیرامون

لوکال : موقعیت محلی

گلوبال : موقعیت جهانی

```
test.py
1  a = 2
2
3  print(a)
4
5
```

الان متغیر a یک متغیر گلوبال هست چون در کل پروژه قابل دسترسی ولی :

```
test.py > ...
1  a = 5
2
3  def show():
4      b = 2
5      print
6
```

در اینجا : متغیر a اسکوپش نسبت به کل پروژه گلوبال ولی نسبت به تابع لوکال است (چون خارج از تابع است و در داخل تابع در دسترس نیست)

متغیر b نسبت به تابع گلوبال و نسبت به کل پروژه لوکال هست (چون خارج از تابع در دسترس نیست)

مواقعی که داریم با توابع چه خالی و چه توابع داخل کلاس ها صحبت میکنیم مهمه که بدونیم اینها چرا مهم هستن

مثلا :

```
test.py > show
1 a = 2
2
3 def show():
4     a+= 1
5
6 show()

Traceback (most recent call last):
  File "c:\Users\hamid\Desktop\New folder (2)\test\test.py", line 6, in <module>
    show()
    ^^^^^
  File "c:\Users\hamid\Desktop\New folder (2)\test\test.py", line 4, in show
    a+= 1
    ^
UnboundLocalError: cannot access local variable 'a' where it is not associated with a value
PS C:\Users\hamid\Desktop\New folder (2)\test>
```

اینجا متغیر a یک متغیر لوکال نسبت به کل پروژه هست (یعنی داخل تابع در دسترس نیست) و وقتی روش تغییری ایجاد کنیم ارور میگیریم که cannot access to local variable یعنی نمیتونه به متغیر لوکال دسترسی داشته باشه

ولی اینجا :

```
test.py > ...
1 a = []
2
3 def show():
4     a.append("a")
5
6 show()
7 show()
8
9 print(a)
```

```
['a', 'a']
```

همون a هستا ! ولی ایندفعه لیسته ولی ارور نگرفتیم !!!! چرا ؟

اینجاست که می‌گیم mutable / immutable چیه :

من سر کلاس با یه مثال توضیح دادم گفتم قابل اسلایس شدن باشن که درسته ولی یه جور میان بره پس باید بدونید دقیق چیه این mutable بودن :

* متغیر هایی که قابل اسلایس بندی شدن (ایندکس گذاری) باشن صد در صد قابل تغییر (mutable) هستن مثل لیست ها رشته ها توپل ها دیکشنری ها ست ها و...

ولی متغیر هایی مثل اعداد صحیح اعداد اعشاری بولین ها و قابل تغییر نیستن

حالا ربطش چیه ؟ متغیر های قابل تغییر همیشه خدا global هستن در هر شرایطی (حتی اگر مثلن خارج از تابع باشه) :

```
test.py > ...
1  a = []
2
3  def show():
4      a.append("a")
5
6  show()
7  show()
8
9  print(a)
```

['a', 'a']

با اینکه خارج از تابع هست ولی بازم قابل دسترسه . ولی در اعداد صحیح :

```
test.py > show
1 a = 2
2
3 def show():
4     a+= 1
5
6 show()

Traceback (most recent call last):
  File "c:\Users\hamid\Desktop\New folder (2)\test\test.py", line 6, in <module>
    show()
    ^^^^^
  File "c:\Users\hamid\Desktop\New folder (2)\test\test.py", line 4, in show
    a+= 1
    ^
UnboundLocalError: cannot access local variable 'a' where it is not associated with a value
PS C:\Users\hamid\Desktop\New folder (2)\test>
```

دیدیم که a رو نمیتونیم تغییر بدیم !

**** بحث ارگومان شدن جداسـت ما داریم درباره تغییر دادن متغیر ها صحبت میکنیم :

```
test.py > show
1 a = 2
2
3 def show(input):
4     print(input)
5
6 show(a)
7
```

اینجا ارگومان شده a در تابع show و پرینت شده . ما داریم درباره تغییر دادن اونا صحبت میکنیم

حالا چکار کنیم که `a` که عدد صحیح هست هم قابل تغییر باشه در اسکوپ خارجی خودش؟


```
test.py > ...
1  a = 2
2
3  def show():
4      global a
5
6      a += 1
7
8  show()
9
10 print(a)
```

اول قبل از تغییر روی متغیر میاییم از keyword مشخص شده به اسم `global` استفاده میکنیم و بعد اسم متغیر رو میزنویسیم . حالا با خیال راحت روی متغیر `immutable` که قابل تغییر در اسکوپ خارجی خودش نیست , تغییر ایجاد میکنیم

کاربرد این گلوبال در کلاس ها :

مثلا قبلن ما میخاستیم کاری کنیم که در برنامه فروش بلیط سینما تعداد کل بلیط ها شمرده بشه ولی میومدیم `self.count` رو به اضافه یک میکردیم ولی روی متغیر اصلی تغییری ایجاد نمیشد . این `count` رو به ابجکت هامون اختصاص میداد و هر کدوم از `count` های ابجکت هارو به اضافه یک میکرد .

```
test.py > ...
Codiumate: Options | Test this class
1 class Ticket :
2
3     count = 0
4
5     Codiumate: Options | Test this method
6     def __init__(self,i_name):
7         self.name = i_name
8
9         self.count +=1
10
11 t1 = Ticket('n')
12 t2 = Ticket('n')
13
14 print(t1.count)
15 print(t2.count)
16
17 print(Ticket.count)
```



همونطور که میبینید به `count` ابجکت `t1` `t2` اضافه کرده نه به `count` که مال خود کلاس هست

اگرم بجای `self.count += 1` بنویسیم `count+=1` ارور میگیریم که `count` در اسکوپ تابع `__init__` نیست . چکار کنیم ؟

روش اول : متغیر رو خارج از اسکوپ تابع `__init__` قرار بدیم (داخل کلاس هم نباید باشه چون هرچی داخل کلاس باشه داخل اسکوپ تابع میره به لطف `self` گفتیم که متغیر های گلوبال با `self` در کل پروژه قابل دسترس اند) :

```
test.py > ...
1  count = 0
2
3  Codiumate: Options | Test this class
4  class Ticket :
5      Codiumate: Options | Test this method
6      def __init__(self,i_name):
7          self.name = i_name
8
9          global count
10         count +=1
11
12
13  t1 = Ticket('n')
14  t2 = Ticket('n')
15
16
17  print(count)
```

2

در این حالت `count` خارج از اسکوپ تابع داندر اینیت و کلاس هست پس با گلوبال کردنش توی تابع داندر اینیت میتونیم روش تغییر ایجاد کنیم (اگر گلوبال نکنی باز ارور میده که `count` در اسکوپ داندر اینیت نیست)

روش 2 :

شاید نخاییم متغیر رو خارج از کلاس تعریف کنیم . اون موقع باید چکار کنیم ؟

```
test.py > ...
1      Codiumate: Options | Test this class
2  ✓ class Ticket :
3      |
4      |     count = 0
5      |     Codiumate: Options | Test this method
6      |     def __init__(self,i_name):
7      |         self.name = i_name
8      |
9      |         Ticket.count +=1
10
11
12  t1 = Ticket('n')
13  t2 = Ticket('n')
14
15
16  print(Ticket.count)
```

2

به متغیری که مال خود کلاس هست یعنی Ticket.count اضافه کنیم

*** این روش یه مشکلی شاید داشته باشه براتون :


```
test.py > ...
1      Codiumate: Options | Test this class
2      class Ticket :
3
4          count = 0
5          Codiumate: Options | Test this method
6          def __init__(self,i_name):
7
8              self.name = i_name
9
10             Ticket.count +=1
11
12     t1 = Ticket('n')
13     t2 = Ticket('n')
14
15
16     print(f'ticket count is : {Ticket.count}')
17
18     print(t1.count)
19     print(t2.count)
```

```
✓
ticket count is : 2
2
2
```

همونطور که میبینید و میدونید اون متغیر count مال همه آبجکت ها هم هست (یک ویژگی گلوبال هست برای کلاس که همه آبجکت ها ازش ارث بری میکنن) پس وقتی تغییر میدیمش با Ticket.count برای کل آبجکت هام تغییر میکنه ! مثلاً شاید این یک دیتابیس بزرگ باشه و نخاییم این حجم از تغییرات رو برای تک تک آبجکت هامون داشته باشیم .

