

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Fonksiyonlar **def** anahtar kelimesiyle tanımlanır. `def fonk_adi(parametre):`

# Function Definition

```
def add(a, b):  
    return a + b
```

Parameters

# Function Call

```
add(2, 3)
```

Arguments

Görsel: <https://getkt.com/blog/types-or-function-arguments-in-python/>, Erişim: Haziran 2022.

getKT.com

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

188

188

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Fonksiyonlar **def** anahtar kelimesiyle tanımlanır. `def fonk_adi(parametre):`

```
def my_function(fname):  
    print(fname + " Refsnes")
```

```
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```

Emil Refsnes  
Tobias Refsnes  
Linus Refsnes

```
def my_function(fname, lname):  
    print(fname + " " + lname)  
my_function("Emil", "Refsnes")
```

Emil Refsnes

- Fonksiyonunuza kaç tane argüman iletileceğini bilmiyorsanız, fonksiyon tanımında parametre adından önce bir **\*** ekleyin. Bu şekilde fonksiyon bir **tuple argüman** alacak ve buna göre öğelere erişebilecektir:

```
def my_function(*kids):  
    print("The youngest child is " + kids[-1])  
my_function("Emil", "Tobias", "Linus")
```

The youngest child is Linus

Python kütüphanelerinde bu tür tuple parametre ismi genelde **\*args** olarak karşımıza çıkar.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

189

189

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Ayrıca **anahtar = değer** sözdizimi ile bağımsız değişkenler gönderebilirsiniz. Bu şekilde argümanların sırası önemli değildir.

```
def my_function(child3, child2, child1):
    print("The youngest child is " + child3)

my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```

The youngest child is Linus

- Fonksiyonunuza kaç tane anahtar kelime argümanının iletileceğini bilmiyorsanız, fonksiyon tanımında parametre adından önce iki yıldız işareti ekleyin: \*\*. Bu şekilde fonksiyon bir **argüman sözlüğü** alacak ve buna göre öğelere erişebilecektir:

```
def my_function(**kid):
    print("His last name is " + kid["lname"])

my_function(fname = "Tobias", lname = "Refsnes")
```

His last name is Refsnes

Python kütüphanelerinde bu tür dictionary parametre ismi genelde **\*kwargs** olarak karşımıza çıkar.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

190

190

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

Aşağıda **add** ve **calculate** fonksiyonlarının çağırılması ve çıktısı verilmiştir.

Buna göre **add** ve **calculate** fonksiyonlarını kodlayın.

- İpucu: **add** için **tuple argüman**, **calculate** için **argüman sözlüğü** yöntemlerini kullanın.

```
print("\nadd fonksiyon çıktıları:")
print(add(3, 5, 7, 2, 1, 7, 4, 3))
print(add(3, 5, 7))
print(add(3))
```

```
add fonksiyon çıktıları:
32
15
3
```

```
print("\ncalculate fonksiyon çıktısı:")
print(calculate(2, add=3, multiply=4))
```

```
calculate fonksiyon çıktısı:
20
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

191

191

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

Cevap:

```
def add(*args):  
    # print(args[1])  
  
    sum = 0  
    for n in args:  
        sum += n  
    return sum
```

```
def calculate(n, **kwargs):  
    # print(kwargs)  
    # for key, value in kwargs.items():  
    #     print(key)  
    #     print(value)  
    n += kwargs["add"]  
    n *= kwargs["multiply"]  
    # print(n)  
    return n
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

192

192

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Aşağıdaki örnek, varsayılan bir parametre değerinin nasıl kullanılacağını gösterir. Eğer fonksiyonu argümentsiz çağırırsak, varsayılan değeri kullanır:

```
def my_function(country = "Norway"):  
    print("I am from " + country)  
  
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

```
I am from Sweden  
I am from India  
I am from Norway  
I am from Brazil
```

- Örnek: Değer döndüren fonksiyon:

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

```
15  
25  
45
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

193

193

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Örnek: Rekürsif (recursive) fonksiyon:

```
def tri_recursion(k):  
    if(k > 0):  
        result = k + tri_recursion(k - 1)  
        print(result)  
    else:  
        result = 0  
    return result  
  
print("\n\nRecursion Example Results")  
tri_recursion(6)
```

```
Recursion Example Results  
1  
3  
6  
10  
15  
21
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

194

194

## Ön Hazırlık: Python 3 – Fonksiyonlar

- Örnek: Liste parametre alan fonksiyon

```
def my_function(sayilar):  
    for x in sayilar:  
        print(x)  
  
geri_sayim = ["dokuz", "sekiz", "yedi"]  
my_function(geri_sayim)
```

```
dokuz  
sekiz  
yedi
```

- Fonksiyon tanımları boş olamaz, ancak herhangi bir nedenle içeriği olmayan bir fonksiyon tanımlamanız gerekirse, hata almamak için **pass** ifadesini girin.

```
def myfunction():  
    pass
```

- Örnek: return ifadesi

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

```
15  
25  
45
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

195

195

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Aıştırma Ödevi

Reeborg's World: Maze

- Fonksiyon tanımlayıp kullanarak çözün.

#### Lost in a maze

Reeborg was exploring a dark maze and the battery in its flashlight ran out.

Write a program using an `if/elif/else` statement so Reeborg can find the exit. The secret is to have Reeborg follow along the right edge of the maze, turning right if it can, going straight ahead if it can't turn right, or turning left as a last resort.

#### What you need to know

- The functions `move()` and `turn_left()`.
- Either the test `front_is_clear()` or `wall_in_front()`, `right_is_clear()` or `wall_on_right()`, and `at_goal()`.
- How to use a `while` loop and `if/elif/else` statements.
- It might be useful to know how to use the negation of a test (`not` in Python).

#### Difficulty level



#### HATIRLATMA!!!

1-Çözümünüz harita üzerinde **herhangi bir** spawn noktasından geçerli olmalı!  
2-Sonra ekteki üç json'a göre test edeceğiz. Robot 1000den az adım atarak sonuca gitmeli.

Benim en kısa çözümüm:  
1 yeni fonksiyon  
17 satır kod

Bunu geçebilir misiniz?

#### Additional options

#### Help

- [Documentation](#)
- [Discussion forum \(English/French\)](#)
- Using Python, execute a program get help on the `move()` function,
- Click on Reeborg keyboard to see
- [Bug reports, suggestions, other is](#)
- (English/French only) Email: [Andr](#)

#### World: creation, edition, ...

**Edit world** You can create your own w

world .json **Save**  
Saves the world (as a json object) to a problems.

**Open world from file** Loads a world fr

196

196

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Aıştırma

Cevap:

The screenshot shows the Reeborg's World interface. On the left, there's a maze with a robot at the entrance. The maze is composed of black walls and white paths. The robot is a small green figure with a flashlight. On the right, there's a Python code editor with the following code:

```
Python Code library
1 def turn_right():
2     turn_left()
3     turn_left()
4     turn_left()
5
6 while not at_goal():
7     if right_is_clear():
8         turn_right()
9         move()
10    if at_goal():
11        break
12 elif wall_in_front():
13     turn_left()
14 elif front_is_clear():
15     move()
16     if at_goal():
17         break
```

28.06.2022

197

197

198

## 199

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

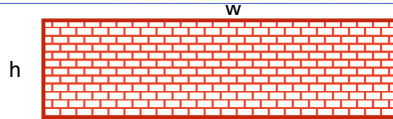
Cevap:

```
import math

def paint_calc(height, width, cover=5):
    return math.ceil((height * width) / cover)

test_h = int(input("Duvar yüksekliği: "))
test_w = int(input("Duvar genişliği: "))

print("%d kutu boya gerekir" %paint_calc(test_h, test_w))
```



### Örnek Çıktı

```
Duvar yüksekliği: 2
Duvar genişliği: 4
2 kutu boya gerekir
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

200

200

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

Sağ altta, iki sözlük içeren bir Liste olan travel\_log koleksiyonunu görebilirsiniz. Çözümünüze bu 13 satırlık kod ile başlamanız gerekiyor.

Rusya için sağ üstteki girdiyi travel\_log'a eklemek üzere bir fonksiyon yazın. Eklemeden önceki ve sonraki listeyi yazdırın. **travel\_log'u doğrudan DEĞİŞTİRMEYİN. Bir fonksiyon yoluyla işlem yapmanız gerekir.**

### İpucu

- Fonksiyonun adının ne olması gerektiğini görmek için sağ üstteki fonksiyon çağrısına bakın.
- Fonksiyonun girdileri **konumsal argümanlardır**. Sıra çok önemlidir.
- Parametre adlarını kendiniz belirleyebilirsiniz.

### Fonksiyon çağrısı:

```
add_new_country("Russia", 2, ["Moscow", "Saint Petersburg"])
```

```
You've visited Russia 2 times.
```

```
You've been to Moscow and Saint Petersburg.
```

### Ödev sözlük:

```
1 travel_log = [
2   {
3     "country": "France",
4     "visits": 12,
5     "cities": ["Paris", "Lille", "Dijon"]
6   },
7   {
8     "country": "Germany",
9     "visits": 5,
10    "cities": ["Berlin", "Hamburg",
11              "Stuttgart"]
12  },
13 ]
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

201



## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

Cevap:

```
15 def add_new_country(name, visit_count, cities_visited):
16     new_country = {"country": name, "visits": visit_count, "cities": cities_visited}
17     travel_log.append(new_country)
18
19
20 print("Before:")
21 print(travel_log)
22 print()
23 add_new_country("Russia", 2, ["Moscow", "Saint Petersburg"])
24
25 print("After:")
26 print(travel_log)
27
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

202

202

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

Bir değerin asal sayı olup olmadığını kontrol eden `prime_checker` fonksiyonu yazın. edip kullanıcıya bilgi döndüren. Değer kullanıcıdan alınacak, sonuç bilgi olarak ekrana yazdırılacaktır. Bkz. [https://tr.wikipedia.org/wiki/Asal\\_say%C4%B1](https://tr.wikipedia.org/wiki/Asal_say%C4%B1), Erişim Haziran 2022.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

#### Örnek çıktıları:

Asallığı kontrol edilecek sayıyı girin: 12  
12 bir asal sayı DEĞİLDİR.

Asallığı kontrol edilecek sayıyı girin: 97  
97 bir asal sayıdır.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

203

203



## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

`prime_checker` fonksiyonunuzu aşağıdaki girdiler için denetleyin. Gerekliyse, beklenen çıktıya göre düzenleme yapın.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Not: Cevap daha sonraki bir konu başlığı altındadır.

#### YANLIŞ CEVAP çıktıları:

Asallığı kontrol edilecek sayıyı girin: -8  
-8 bir asal sayıdır.

```
Asallığı kontrol edilecek sayıyı girin: 2.3
Traceback (most recent call last):
  File "pyclassroom2022/Day1/asal_kontrol.py", line 37, in <module>
    prime_checker1(number=n)
  File "pyclassroom2022/Day1/asal_kontrol.py", line 10, in prime_checker1
    number = int(number)
ValueError: Invalid literal for int() with base 10: '2.3'
```

#### BEKLENEN çıktıları:

Asallığı kontrol edilecek sayıyı girin: -8  
-8 bir asal sayı DEĞİLDİR.

Asallığı kontrol edilecek sayıyı girin: 2.3  
Hata: Tamsayı tipinde bir girdi olmalıdır.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

204

204

## Ön Hazırlık: Python 3 – Fonksiyonlar

### Alıştırma Ödevi

`prime_checker` fonksiyonunuzu aşağıdaki girdiler için denetleyin. Gerekliyse, beklenen çıktıya göre düzenleme yapın.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Not: Cevap daha sonraki bir konu başlığı altındadır.

#### YANLIŞ CEVAP çıktıları:

Asallığı kontrol edilecek sayıyı girin: -8  
-8 bir asal sayıdır.

```
Asallığı kontrol edilecek sayıyı girin: 2.3
Traceback (most recent call last):
  File "pyclassroom2022/Day1/asal_kontrol.py", line 37, in <module>
    prime_checker1(number=n)
  File "pyclassroom2022/Day1/asal_kontrol.py", line 10, in prime_checker1
    number = int(number)
ValueError: Invalid literal for int() with base 10: '2.3'
```

#### BEKLENEN çıktıları:

Asallığı kontrol edilecek sayıyı girin: -8  
-8 bir asal sayı DEĞİLDİR.

Asallığı kontrol edilecek sayıyı girin: 2.3  
Hata: Tamsayı tipinde bir girdi olmalıdır.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

205

205

## Ön Hazırlık: Python 3 – Fonksiyonlar: `lambda`

- Bir `lambda` fonksiyonu, küçük bir anonim işlevdir.
- Herhangi bir sayıda argüman (argument) alabilir, ancak yalnızca bir ifadeye (expression) sahip olabilir.
- En sık kullanım: başka bir fonksiyonun içinde anonim bir fonksiyon olarak.
- Syntax: `lambda arguments : expression`

Örnekler:

```
x = lambda a: a + 10  
print(x(5))
```

15

```
x = lambda a, b: a * b  
print(x(5, 6))
```

30

```
x = lambda a, b, c: a + b + c  
print(x(5, 6, 2))
```

13

```
def myfunc(n):  
    return lambda a : a * n
```

22  
33

```
mydoubler = myfunc(2)  
mytripler = myfunc(3)
```

```
print(mydoubler(11))  
print(mytripler(11))
```

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

206

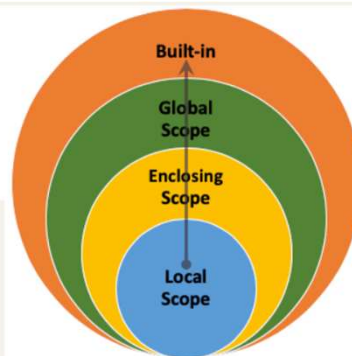
206

## Ön Hazırlık: Python 3 – Fonksiyonlar: Scope of Variables (Değişkenlerin Kapsamı)

LEGB Kuralı

- LEGB (Local (Yerel) -> Enclosing (Çevreleyen) -> Global (Genel) -> Built-in (Yerleşik)), programınızı yürütürken bir Python yorumlayıcısının izlediği mantıktır.

```
# Global scope  
x = 0  
  
def outer():  
    # Enclosed scope  
    x = 1  
    def inner():  
        # Local scope  
        x = 2
```



Kaynak: <https://www.datacamp.com/tutorial/scope-of-variables-python>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

207

207

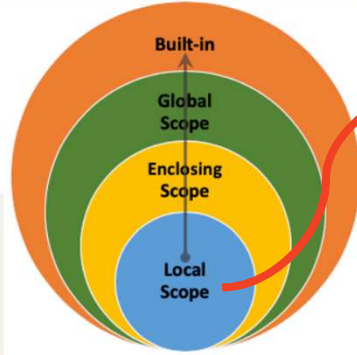
## Ön Hazırlık: Python 3 – Fonksiyonlar: Scope of Variables (Değişkenlerin Kapsamı)

### LEGB Kuralı

- **LEGB** (**L**ocal (Yerel) -> **E**nclosing (Çevreleyen) -> **G**lobal (Genel) -> **B**uilt-in (Yerleşik)), programınızı yürütürken bir Python yorumlayıcısının izlediği mantıktır.

```
# Global scope
x = 0

def outer():
    # Enclosed scope
    x = 1
    def inner():
        # Local scope
        x = 2
```



Bir fonksiyon içinde bir değişken tanımladığınızda, kapsamı YALNIZCA o fonksiyonun içindedir. Tanımlandığı noktadan fonksiyonun sonuna kadar erişilebilir ve fonksiyon çalıştığı sürece (Kaynak) var olur.

Kaynak: <https://www.datacamp.com/tutorial/scope-of-variables-python>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

208

208

```
def outer():
    first_num = 1
    def inner():
        second_num = 2
        # Print statement 1 - Scope: Inner
        print("first_num from outer: ", first_num)
        # Print statement 2 - Scope: Inner
        print("second_num from inner: ", second_num)
    inner()
    # Print statement 3 - Scope: Outer
    print("second_num from inner: ", second_num)

outer()
```

```
# Global scope
x = 0

def outer():
    # Enclosed scope
    x = 1
    def inner():
        # Local scope
        x = 2
```



```
first_num from outer: 1
second_num from inner: 2

NameError                                Traceback (most recent call last)
<ipython-input-4-13943a1eb01e> in <module>
     11 print("second_num from inner: ", second_num)
     12
--> 13 outer()

<ipython-input-4-13943a1eb01e> in outer()
      9 inner()
     10 # Print statement 3 - Scope: Outer
--> 11 print("second_num from inner: ", second_num)
     12
     13 outer()
```

### İç içe fonksiyonlarda:

Yukarıdaki hatanın nedeni, second\_num'a external()'den erişememenizdir. Bu fonksiyon içinde tanımlanmamıştır. Ancak, first\_num'a inner()'den erişebilirsiniz, çünkü first\_num'un kapsamı daha büyük olduğundan outer() içindedir.

[/scope-of-](https://www.datacamp.com/tutorial/scope-of-variables-python)

28.06.2022

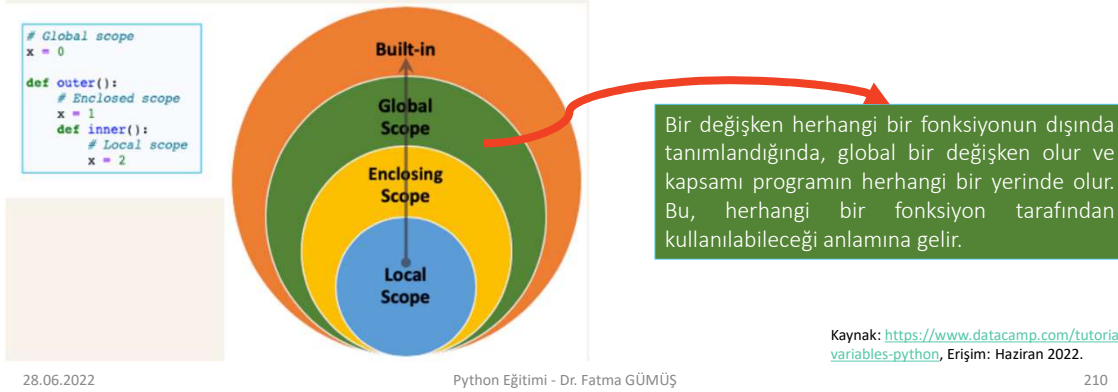
Python Eğitimi - Dr. Fatma GÜMÜŞ

209

## Ön Hazırlık: Python 3 – Fonksiyonlar: Scope of Variables (Değişkenlerin Kapsamı)

### LEGB Kuralı

- **LEGB** (**L**ocal (Yerel) -> **E**nclosing (Çevreleyen) -> **G**lobal (Genel) -> **B**uilt-in (Yerleşik)), programınızı yürütürken bir Python yorumlayıcısının izlediği mantıktır.



28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

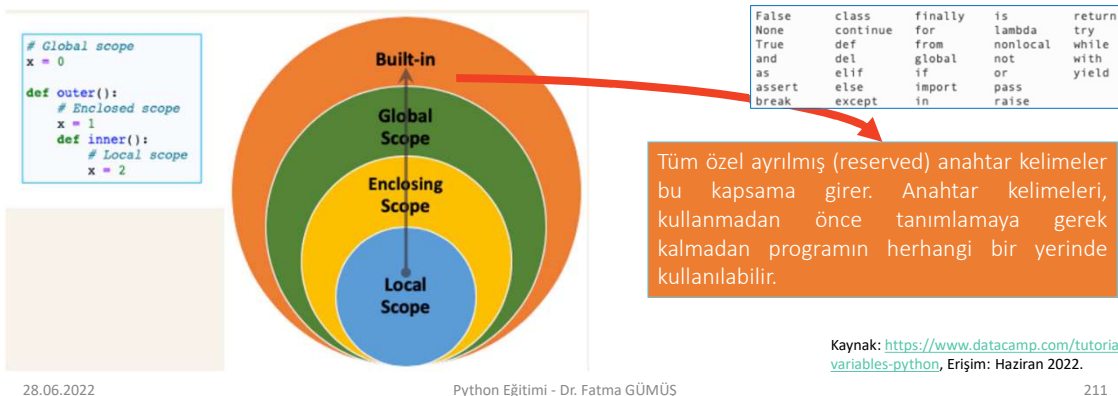
210

210

## Ön Hazırlık: Python 3 – Fonksiyonlar: Scope of Variables (Değişkenlerin Kapsamı)

### LEGB Kuralı

- **LEGB** (**L**ocal (Yerel) -> **E**nclosing (Çevreleyen) -> **G**lobal (Genel) -> **B**uilt-in (Yerleşik)), programınızı yürütürken bir Python yorumlayıcısının izlediği mantıktır.



28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

211

211

## Ön Hazırlık: Python 3 – Fonksiyonlar: Scope of Variables (Değişkenlerin Kapsamı)

NOT:

- Python'da block scope yoktur!

Kaynak: <https://www.datacamp.com/tutorial/scope-of-variables-python>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

212

212

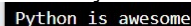
## Ön Hazırlık: Python 3 – Fonksiyonlar: Global ve Yerel Değişkenler

- Bir fonksiyonun dışında oluşturulan değişkenler **global değişkenler** olarak bilinir.
- Global değişkenler, hem fonksiyonların içinde hem de dışında herkes tarafından kullanılabilir.

```
x = "awesome"

def myfunc():
    print("Python is " + x)

myfunc()
```



Python is awesome

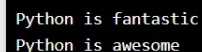
- Bir fonksiyon içinde aynı isimde bir değişken oluşturursanız, bu değişken yerel olur ve sadece fonksiyon içinde kullanılabilir. Aynı ada sahip global değişken, global ve orijinal değerde kalacaktır.

```
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```



Python is fantastic  
Python is awesome

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

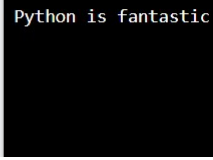
213

213

## Ön Hazırlık: Python 3 – Fonksiyonlar: Global ve Yerel Değişkenler

- Normalde, bir fonksiyon içinde bir değişken oluşturduğunuzda, bu değişken yereldir ve yalnızca o fonksiyonun içinde kullanılabilir.
- Bir fonksiyon içinde global bir değişken oluşturmak için **global** anahtar sözcüğünü kullanabilirsiniz.

```
def myfunc():  
    global x  
    x = "fantastic"  
  
myfunc()  
  
print("Python is " + x)
```

A terminal window with a black background and white text displaying the output of the Python code: "Python is fantastic".

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

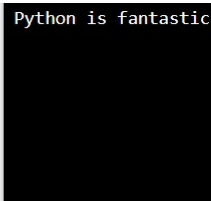
214

214

## Ön Hazırlık: Python 3 – Fonksiyonlar: Global ve Yerel Değişkenler

- Bir fonksiyon içindeki global değişkenin değerini değiştirmek için **global** anahtar sözcüğünü kullanarak bu global değişkenin değerinde güncelleme yaptığınızı belirtin:

```
x = "awesome"  
  
def myfunc():  
    global x  
    x = "fantastic"  
  
myfunc()  
  
print("Python is " + x)
```

A terminal window with a black background and white text displaying the output of the Python code: "Python is fantastic".

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

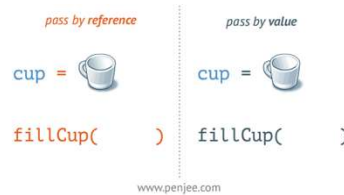
215

215

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

**Pass by reference** – Bazı programlama dillerinde kullanılır, burada fonksiyonun argümanına verilen değerler referansla iletilir, bu da değişkenin adresinin iletildiği ve ardından işlemin bu adreslerde depolanan değer üzerinde yapıldığı anlamına gelir.

**Pass by value** – Değerin doğrudan fonksiyonun bağımsız değişkenine değer olarak iletildiği anlamına gelir. Burada değer üzerinde işlem yapılır ve daha sonra değer adrese kaydedilir. Pass by value, değişkenin bir kopyası için kullanılır.



Kaynak: <https://pythonguides.com/python-pass-by-reference-or-value/>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

216

216

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

Call by reference	Call by value
Bir programlama dilinde bir fonksiyon çağırılırken değişkenlerin değerlerini kopyalamak yerine değişkenlerin adresi kullanılır, buna "Referansla Çağır" denir.	Bir fonksiyonu çağırırken, değişkenleri kopyalayarak değerleri ilettiğimizde, "Değerlere Göre Çağır" olarak bilinir.
Bu yöntemde, bir değişkenin kendisi iletilir.	Değişkenin bir kopyası iletilir.
Değişkendeki değişiklik, fonksiyon dışındaki değişkenin değerini de etkiler.	Bir değişkenin kopyasında yapılan değişiklikler, fonksiyonun dışındaki değişkenin değerini asla değiştirmez.
Fonksiyon çağrılarını kullanarak değişkenlerin değerlerinde değişiklik yapmanızı sağlar.	Gerçek değişkenlerde herhangi bir değişiklik yapmanıza izin vermez.
Orijinal değer değiştirilir.	Orijinal değer değiştirilmedi.

Kaynak: <https://pythonguides.com/python-pass-by-reference-or-value/>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

217

217



## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

Python, «call-by-object» olarak bilinen, bazen «call-by-object-reference» veya «call-by-sharing» olarak da adlandırılan bir mekanizma kullanır.

Bir fonksiyona tamsayılar, string'ler veya tuple gibi değişmez (immutable) argümanlar iletirseniz, call by value gibi davranır. Değişen argümanları (mutable) iletirsek call by reference gibi davranır.

Python dilindeki tüm parametreler (argümanlar) «call-by-object-reference» olarak iletilir.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

218

218

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

Python'da fonksiyon argümanları her zaman «call-by-object-reference» olarak iletilir. Bu, olgusal, biçimsel argümanların ve döndürülen nesnelerin id()'si kontrol edilerek doğrulanabilir.

```
def val(x):
    x = 15
    print(x, id(x))

x = 10
val(x)
print(x, id(x))
```

x = 10

10

  
23425

x = 15

15

  
76525

A new object is created in the memory because integer objects are immutable (not modifiable).

---

```
def val(lst):
    lst.append(4)
    print(lst, id(lst))

lst = [1, 2, 3]
val(lst)
```

lst = [1, 2, 3]

1, 2, 3, 4

  
76345

lst = [1, 2, 3, 4]

lst

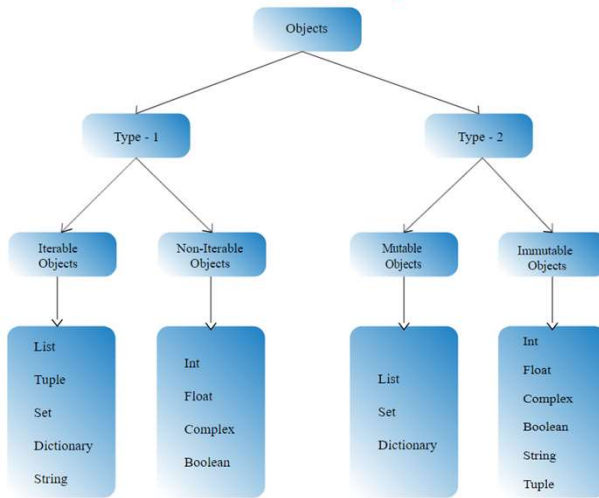
A new object is not created in the memory because list objects are mutable (modifiable). It simply add new element to the same object.

Görsel: <https://medium.com/@lokeshsharma596/is-python-call-by-value-or-call-by-reference-2dd7db74dbd0>, Erişim: Haziran 2022.

219

219

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»



28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

220

**Immutable object reference:** Nesne referansı, fonksiyon parametrelerine iletilir. İşlev içinde değiştirilemezler, çünkü hiç değiştirilemezler, yani değişmezdirler.

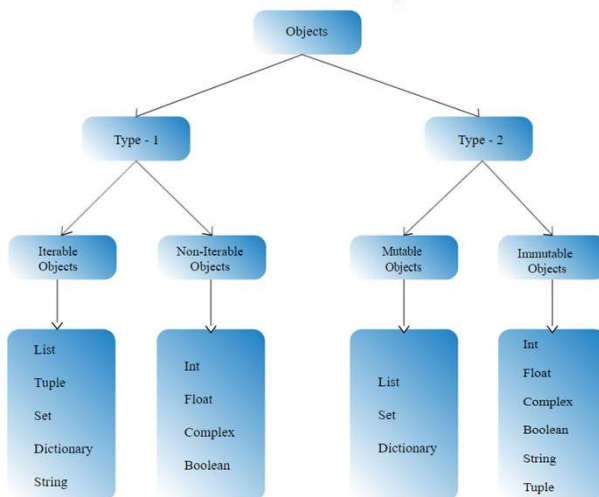
**Mutable object reference:** Yine nesne referansı iletilirler, ancak fonksiyonda yerinde değiştirilebilirler.

Bir listeyi bir fonksiyona iletirsek, iki durumu dikkate almalıyız:

1. Bir listenin öğeleri yerinde değiştirilebilir, yani liste, onu çağırın fonksiyonun kapsamında bile değişecektir.
2. Fonksiyon içinde aynı isimde yeni bir liste atanırsa, eski liste etkilenmez, yani çağırın fonksiyonun kapsamındaki liste ayrıdır.

220

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»



28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

<b>Iterable vs Non-Iterable</b>	<pre> x=[1,2,3,4,5] print(type(x)) for i in x:     print(i) </pre>	<pre> &lt;class 'list'&gt; 1 2 3 4 5 </pre>
<b>Mutable vs Immutable</b>	<pre> x=[1, 2, 3] print('x value: {}'.format(x)) print('id of x: {}'.format(id(x))) print('type of x: {}'.format(type(x))) x[2] = 300 print('modified x: {}'.format(x)) print('id of modified x: {}'.format(id(x))) </pre>	<pre> x = 65 print(type(x)) for i in x:     print(i) </pre> <p>File "H:\python\test\test.py", line 7, in &lt;module&gt; for i in x: TypeError: 'int' object is not iterable</p>

221

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

Örnek:

```
student = {'Jim': 12, 'Anna': 14, 'Preet': 10}
def test(student):
    new = {'Sam': 20, 'Steve': 21}
    student.update(new)
    print("Inside the function", student)
    return
test(student)
print("Outside the function:", student)
```

Inside the function {'Jim': 12, 'Anna': 14, 'Preet': 10, 'Sam': 20, 'Steve': 21}  
Outside the function: {'Jim': 12, 'Anna': 14, 'Preet': 10, 'Sam': 20, 'Steve': 21}

```
student = {'Jim': 12, 'Anna': 14, 'Preet': 10}
def test(student):
    student = {'Sam': 20, 'Steve': 21}
    print("Inside the function", student)
    return
test(student)
print("Outside the function:", student)
```

Inside the function {'Sam': 20, 'Steve': 21}  
Outside the function: {'Jim': 12, 'Anna': 14, 'Preet': 10}

Kaynak: <https://pythonguides.com/python-pass-by-reference-or-value/>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

222

222

## Ön Hazırlık: Python 3 – Fonksiyonlar: «pass/call by reference» vs «pass/call by value»

Örnek: Çıktı ne olur?

```
def set_list(list):
    list = ["A", "B", "C"]
    return list

def add(list):
    list.append("D")
    return list

my_list = ["E"]

print(set_list(my_list))

print(add(my_list))
```

CEVAP

Kaynak: <https://pythonguides.com/python-pass-by-reference-or-value/>, Erişim: Haziran 2022.

28.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

223

223