```python
# %%

a = [10, 20, 30, 40, 50]
e = enumerate(a)
print(type(e))

# %%
d = {'ali': 10, 'hakan': 20, 'gül': 30, 'sevim': 40,
'mehmet':50}
print(list(enumerate(d)))
# [(0, 'ali'), (1, 'hakan'), (2, 'gül'), (3, 'sevim'), (4,
'mehmet')]
# %%
a = [12, 15, 38, 84, 42, 30, 28, 56, 81, 16]
# en büyük elemanın kaçıncı indekste olduğunu bulan program
max_index = 0
for i in range(1, len(a)):
    if a[max_index] < a[i]:
        max_index = i
print("En büyük elemanın indeksi = {}".format(max_index))
# %%
a = [12, 15, 38, 84, 42, 30, 28, 56, 81, 16]
# en büyük elemanın kaçıncı indekste olduğunu bulan program
e = enumerate(a)
max = next(e)
for t in e:
    if t[1] > max[1]:
        max = t
print("En büyük elemanın indeksi = {}".format(max[0]))
# %%
"""
1- Global değişken
2- Yerel değişkenler
"""
x = 10
def foo():
    print(x)
def bar():
    print(x)

foo()
x = 20
bar()
print(x)

# %%
x = 10
```

```python
def foo():
    y = 20
    print(y)
foo()
print(x)
# print(y) # error!!
# %%
x = 100
def foo():
    x = 200
    print(x) # 200
foo()
print(x) # 100
# %%
x = 10
def foo():
    global x
    x = 20
    print(x)
foo() # x = 20
print(x) # 20
# %%
x = 10
def foo():
    print(x)
    global x # error
    x = 20
foo()
print(x)

# %%
def foo():
    global x
    print(x) # exception
# %%
def foo():
    x = [1, 2, 3, 4, 5]
    for i in x:
        print(i)
    print(i)    # geçerli
foo()
# %%
def foo():
    i = 0
    while i < 10:
        x = 10
        i += 1
```

```
    print(x) # geçerli
foo()
# %%
def foo(x):
    if x > 0:
        y = 10
    print(y)
foo(10) # geçerli, sorun yok
foo(-10) # exception oluşur!!!
# %%
# map (function, iterable, ...)
a = [1, 2, 3, 4, 5]
def foo(n):
    return n * n
b = map(foo, a)
print(list(b))
# %%
a = [1, 2, 3, 4, 5]
b = [i * i for i in a]
print(b)
# %%

a = [
    [1, 2, 3, 4, 5],
    [5, 6],
    [7, 8, 9],
    [10]
]
# listenin en uzun olan elemanının uzunluğu
print(max(map(len, a)))
# %%
def foo():
    print('foo begins')
    def bar():
        print('bar begins')
        print('bar ends')
    bar()
    print('foo ends')
foo()
"""
foo begins
bar begins
bar ends
foo ends
"""
# %%
def bar():
```

```
    print('Global bar')
def foo():
    def bar():
        print('Nested bar')
    bar()
foo() # Nested bar
bar() # Global bar
# %%
def bar():
    print('Global bar')
def foo():
    bar()  # error
    def bar():
        print('Nested bar')
# %%
bar = 10
def foo():
    print(bar) # error!!!
    bar = 20
# %%
def bar():
    print('Global bar')
def foo():
    global bar
    bar()
    def bar():
        print('Nested bar')
    bar()
foo()
bar()
"""
Global bar
Nested bar
Nested bar
"""
# %%
def foo():
    val = 10
    def bar():
        print('Bar begins')
        print('val = {0}'.format(val))
        print('Bar ends')
    bar()
    print('foo ends')
foo()
"""
Bar begins
```

```
val = 10
Bar ends
foo ends
"""
# %%
def foo():
    x = 10
    def bar():
        print(x) # error
        x = 20
        print(x)
    bar()
    print(x)
foo()
# %%
def foo():
    x = 10
    def bar():
        nonlocal x
        print(x)     # foo'daki x
        x = 20       # foo'daki x
        print(x)      # foo'daki x
    bar()
    print(x)
foo()
# %%
x = 10
def foo():
    def bar():
        nonlocal x  # error!!!
        print(x)
        x = 20
        print(x)
    bar()
foo()
# %%
def foo():
    x = 10
    def bar():
        def tar():
            nonlocal x  # foo'daki x
            x = 20
        tar()
    bar()
    print(x) # 20
foo()
# %%
```

```
def foo():
    x = 10
    def bar():
        nonlocal x # foo'daki x
        x = 20
        def tar():
            nonlocal x   # bar'daki x
            x = 30
        tar()
    bar()
    print(x)     # 30
foo()
# %%
x = 10
def foo():
    global x
    x = 20
    def bar():
        nonlocal x   # error!!! x üst fonksiyonda global
        x = 30
    bar()
foo()
print(x)
# %%
x = 10
def foo():
    global x
    x = 20
    def bar():
        global x   # geçerli
        x = 30
    bar()
foo()
print(x)
# %%
def writePrimes(val):
    def isPrime(val):
        if val % 2 == 0:
            return val == 2
        for i in range(3, val, 2):
            if val % i == 0:
                return False
            return True
    for i in range(2, val+1):
        if isPrime(i):
            print(i, end=' ')
writePrimes(100)
```