

Python 3 – iterator (yineleyici) ve iterable (yinelenen)

Iterator

- Iterator, sayılabilir sayıda değer içeren bir nesnedir.
- Iterator, yinelenen bir nesnedir, yani tüm değerler arasında geçiş yapabilirsiniz.
- Teknik olarak Python'da iterator, `__iter__()` ve `__next__()` fonksiyonlarından oluşan yineleyici protokolünü uygulayan bir nesnedir.

Iterable

- list, tuple, dict ve set veri yapılarının tümü yinelenen nesnelerdir. Bir iterator alabileceğiniz iterable yapılardır. Her birinin iterator elde etmek için kullanılan bir `iter()` fonksiyonu vardır.

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

301

301

Python 3 – iterator (yineleyici) ve iterable (yinelenen)

- list, tuple, dict ve set veri yapılarının tümü yinelenen nesnelerdir.
- String'ler de bir dizi karakter içeren iterable nesnelerdir:

```
mytuple = ("apple", "banana", "cherry")
myit = iter(mytuple)

print(next(myit))
print(next(myit))
print(next(myit))
```

```
apple
banana
cherry
```

```
mystr = "apple"
myit = iter(mystr)
```

```
print(next(myit))
print(next(myit))
print(next(myit))
print(next(myit))
print(next(myit))
```

```
a
p
p
l
e
```

- Iterable bir nesneyi `for` döngüsünde kullanabiliriz:

```
mytuple = ("apple", "banana", "cherry")

for x in mytuple:
    print(x)
```

```
apple
banana
cherry
```

29.06.2022

Dr. Fatma GÜMÜŞ

302

302

Python 3 – iterator (yineleyici) ve iterable (yinelenebilen)

Iterator sınıfı oluşturmak

- Iterator olarak bir nesne/sınıf oluşturmak için `__iter__()` ve `__next__()` fonksiyonları kodlamanız gerekir.
- Python'da tüm sınıfların `__init__()` adında bir fonksiyonu vardır ve bu, nesne oluşturulurken çağırılır.
- `__iter__()` fonksiyonu buna benzer davranır, işlemler yapabilirsiniz (ilk değer atama vb.), ancak her zaman iterator nesnenin kendisini döndürmeniz gerekir.
- `__next__()` fonksiyonunda da işlem yapabilirsiniz ve sıradaki bir sonraki öğeyi döndürmesi gerekir.

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

303

303

Python 3 – iterator (yineleyici) ve iterable (yinelenebilen)

StopIteration

- Yeterli `next()` ifadeniz varsa veya bir `for` döngüsünde kullanılmışsa, soldaki örnek sonsuza kadar devam eder.
- Yinelemenin sonsuza kadar devam etmesini önlemek için `StopIteration` ifadesini kullanabiliriz.
- `__next__()` yönteminde, yineleme belirli sayıda yapılırsa bir hata oluşturmak için bir sonlandırma koşulu ekleyebiliriz (sağda).

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        x = self.a
        self.a += 1
        return x

myclass = MyNumbers()
myiter = iter(myclass)

print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))
```

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        if self.a <= 20:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

myclass = MyNumbers()
myiter = iter(myclass)

for x in myiter:
    print(x)
```

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

304

304

Python 3 – İstisna (Exception) İşleme: try...except

| | |
|----------------|---|
| try | bir kod bloğunu hatalar için test etmenizi sağlar. |
| except | Hata olması durumunda çalışır. |
| else | Hata olmadığında çalışır. |
| finally | try ve except bloklarının sonucundan bağımsız olarak çalışır. |

```
print(x)
```

Traceback (most recent call last):
File "demo_try_except_error.py", line 3, in <module>
 print(x)
NameError: name 'x' is not defined

```
try:  
    print(x)  
except:  
    print("An exception occurred")
```

An exception occurred

```
try:  
    print(x)  
except NameError:  
    print("Variable x is not defined")  
except:  
    print("Something else went wrong")
```

Variable x is not defined

```
try:  
    print("Hello")  
except:  
    print("Something went wrong")  
else:  
    print("Nothing went wrong")
```

Hello
Nothing went wrong

```
try:  
    print(x)  
except:  
    print("Something went wrong")  
finally:  
    print("The 'try except' is finished")
```

Something went wrong
The 'try except' is finished

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

305

305

Python 3 – İstisna (Exception) İşleme: try...except

| | |
|----------------|---|
| try | bir kod bloğunu hatalar için test etmenizi sağlar. |
| except | Hata olması durumunda çalışır. |
| else | Hata olmadığında çalışır. |
| finally | try ve except bloklarının sonucundan bağımsız olarak çalışır. |

```
try:  
    f = open("demofile.txt")  
    try:  
        f.write("Lorum Ipsum")  
    except:  
        print("Something went wrong when writing to the file")  
    finally:  
        f.close()  
except:  
    print("Something went wrong when opening the file")
```

```
C:\Users\My Name>python demo_try_except5.py  
Something went wrong when writing to the file
```

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

306

306

Python 3 – İstisna (Exception) İşleme:Raise an exception

- Bir koşul oluştuğunda, bir istisna (exception) oluşturmayı seçebilirsiniz.
- Bir istisna atmak (veya yükseltmek) için «**raise**» anahtar sözcüğünü kullanılır.

```
x = -1
if x < 0:
    raise Exception("Sorry, no numbers below zero")
```

```
Traceback (most recent call last):
  File "demo_ref_keyword_raise.py", line 4, in <module>
    raise Exception("Sorry, no numbers below zero")
Exception: Sorry, no numbers below zero
```

- Ne tür bir hata oluşturulacağını ve kullanıcıya yazdırılacak metni tanımlayabilirsiniz (Hata türleri için bkz. <https://docs.python.org/3/library/exceptions.html>)

```
x = "hello"
if not type(x) is int:
    raise TypeError("Only integers are allowed")
```

```
Traceback (most recent call last):
  File "demo_ref_keyword_raise2.py", line 4, in <module>
    raise TypeError("Only integers are allowed")
TypeError: Only integers are allowed
```

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

307

307

Python 3 – İstisna (Exception) İşleme:Raise an exception

Alıştırma Ödevi

Daha önce (Fonksiyonlar) bölümünde yazdığınız **prime_checker** fonksiyonunuzu aşağıdaki girdiler için denetleyin. Gerekliyse, beklenen çıktıya göre düzenleme

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|
| 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 |

YANLIŞ CEVAP çıktılar:
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_checker(number=n)
  File "pyclassroom2022/Day1/asal_kontrol.py", line 10, in prime_checker
    number = int(number)
ValueError: Invalid literal for int() with base 10: '2.3'
```

BEKLENEN çıktılar:

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

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

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

308

308

Ön Hazırlık: Python 3 – Fonksiyonlar

Alıştırma Ödevi

Cevap:

BEKLENEN çıktılar:

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

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

```
21 def prime_checker2(number):
22     """
23     Return whether number is prime or not.
24     Note: CHECK for negativity and invalid input.
25     :param number:
26     :return:
27     """
28     try:
29         number = int(number)
30     except ValueError:
31         print("Hata: Tamsayı tipinde bir girdi olmalıdır.")
32         return
33
34     if number <= 1 or (number % 2 == 0 and number > 2):
35         is_prime = False
36     else:
37         is_prime = all(number % i for i in range(3, int(math.sqrt(number)) + 1, 2))
38     if is_prime:
39         print("%d bir asal sayıdır." % number)
40     else:
41         print("%d bir asal sayı DEĞİLDİR." % number)
42
```

29.06.2022

Python Eğitimi - Dr. Fatma GÜMÜŞ

309

309