# Python Hızlı Başvuru Kılavuzu: Temel Bilgiler

## Python Veri Tipleri

### String

Metin olarak saklanan karakter dizisi.

Örnek:

my\_string = "Merhaba"

#### String İşlemleri

* my\_string.upper() → stringin tüm karakterlerini büyük harfe çevirir
* len(my\_string) → stringin uzunluğunu döndürür
* my\_string.find('l') → 'l' karakterinin ilk indeksini döndürür, yoksa -1
* my\_string.replace('H', 'C') → 'H' karakterini 'C' ile değiştirir

### Integer (Tam Sayı)

Tam sayılar.

Örnek:

my\_integer = 12321

### Float (Ondalıklı Sayı)

Ondalıklı sayılar.

Örnek:

my\_decimal = 3.14

### Boolean (Mantıksal Değer)

True veya False değerleri.

Örnek:

a = True

b = False

### Dictionary (Sözlük)

Anahtar‑değer çiftleri koleksiyonu.

Örnek:

my\_dictionary = {'banana': 1, 12: 'laptop', (0,0): 'center'}

#### Dictionary İşlemleri

* my\_dictionary['banana'] → anahtar ile değere erişir
* my\_dictionary.keys() → tüm anahtarları listeler
* my\_dictionary.values() → tüm değerleri listeler

### Tuple (Demet)

Değiştirilemez nesne koleksiyonu.

Örnek:

tup = (1, 3.12, False, "Merhaba")

### List (Liste)

Değiştirilebilir nesne koleksiyonu.

Örnek:

my\_collection = [1, 1, 3.12, False, "Merhaba"]

#### List İşlemleri

* len(my\_collection) → listenin uzunluğunu döndürür
* my\_collection.extend(["Daha", "Fazla"]) → çoklu öğe ekler
* my\_collection.append("Tek") → tek bir öğe ekler
* del(my\_collection[2]) → belirtilen indeksteki öğeyi siler
* clone = my\_collection[:] → listeyi klonlar
* my\_collection\_3 = my\_collection + my\_collection\_2 → listeleri birleştirir
* sum(number\_collection) → sayısal listedeki öğelerin toplamını hesaplar
* item in my\_collection → öğenin listede bulunup bulunmadığını kontrol eder
* item not in my\_collection → öğenin listede bulunmadığını kontrol eder

### Set (Küme)

Sırasız ve benzersiz öğeler koleksiyonu.

Örnek:

a = {100, 3.12, False, "Hoşça kal"}

b = {100, 3.12, "Hoşgeldin"}

#### Set İşlemleri

* my\_set = set([1,1,2,3]) → listeden küme oluşturur
* a.add(4) → öğe ekler
* a.remove("Hoşça kal") → öğe siler
* a.difference(b) → a \ b farkını verir
* a.intersection(b) → a ∩ b kesişimini verir
* a.union(b) → a ∪ b birleşimini verir
* a.issubset(b) → a ⊆ b ise True
* a.issuperset(b) → a ⊇ b ise True

## İndeksleme ve Dilimleme

* İndeksleme: my\_string[element\_number]
* Dilimleme: my\_string[start:stop]

## Karşılaştırma Operatörleri

* Eşit: a == b
* Küçüktür: a < b
* Büyüktür: a > b
* Büyük veya eşit: a >= b
* Küçük veya eşit: a <= b
* Eşit değil: a != b

## Aritmetik Operatörler

* + : Toplama
* - : Çıkarma
* \* : Çarpma
* / : Bölme
* // : Tamsayı bölme (sonuç tam sayıya yuvarlanır)

## Mantıksal Operatörler

* and → her iki ifade de True ise True
* or → ifadelerden en az biri True ise True
* not → ifadenin tersini döndürür

## Döngüler

### For Döngüsü

for x in range(n): döngüyü n kez çalıştırır

for x in iterable: iterable içindeki her öğe için çalıştırır

### While Döngüsü

while koşul: koşul True olduğu sürece döngüyü çalıştırır

## Koşullu İfadeler

if koşul:

# koşul doğruysa çalışır

elif başka\_koşul:

# önceki koşul yanlış, bu koşul doğruysa çalışır

else:

# tüm koşullar yanlışsa çalışır

## Hata Yakalama (try/except)

try:

# denenmek istenen kod

except Tür:

# belirli hatada çalışacak kod

else:

# hata yoksa çalışacak kod

### Yaygın Hata Türleri

* IndexError: İndeks aralık dışında
* NameError: Değişken tanımlı değil
* SyntaxError: Yazım hatası
* ZeroDivisionError: Sıfıra bölme hatası

## range() Fonksiyonu

range(stop) → 0'dan stop-1'e kadar sayı üretir

range(start, stop, step) → start'dan stop-1'e kadar step adımlarla sayı üretir

## Web Kazıma (Web Scraping)

from bs4 import BeautifulSoup

soup = BeautifulSoup(html, 'html5lib')

soup.prettify()

soup.find(tag)

soup.find\_all(tag)

## HTTP İstekleri (requests)

import requests

response = requests.get(url, params)

response.status\_code

response.text

requests.post(url, params)

## Fonksiyonlar

def fonksiyon\_adi(param1, param2=varsayılan):

# kod

return çıktı

çıktı = fonksiyon\_adi(değer1, değer2)

## Dosyalarla Çalışma

### Okuma

file = open(dosya\_adı, "r")

icerik = file.read()

file.close()

### Yazma

file = open(dosya\_adı, "w")

file.write(icerik)

file.close()

## Nesneler ve Sınıflar

class SınıfAdı:

def \_\_init\_\_(self, param1, param2):

self.özellik1 = param1

self.özellik2 = param2

def metod\_adı(self, param):

return çıktı

nesne = SınıfAdı(değer1, değer2)

nesne.metod\_adı(değer)

© IBM Corporation 2021. Kaynak: "Python Cheat Sheet: The Basics" Coursera kurs materyali.