Colab ve Python'a Giriş

Python programlama diline ve bu eğitimi takip ederken kullanılan ortama (colab) hızlı bir giriş niteliğinde olan colab dosyasına hoş geldiniz.

Colab, Google Cloud kullanarak tarayıcı üzerinde çalışan bir Python geliştirme ortamıdır.

En basit şekliyle aktarmak gerekirse "Merhaba Dünya!" yazdırmak için fareyi [ ] üzerine getirip sol tıka basın böylelikle aşağıdaki kod satırını çalıştırmış olursunuz. Bununla beraber `shift + enter` tuşuna basmanızda aynı şekilde kod satırını/bloğunu çalıştıracaktır.

In [1]:

```
print("Merhaba Dünya!")
```

```
Merhaba Dünya!
```

Fonksiyonlar, Koşullar ve Yinelemeler

Hadi bir Python işlevi (fonksiyon) oluşturalım ve bu fonksiyonu for döngüsü içerisinde çağıralım.

In [2]:

```
def HelloWorldWithXY(x, y):
  if (x < 10):
    print("Merhaba Dünya, x  < 10 (x, 10'dan küçük)")
  elif (x < 20):
    print("Merhaba Dünya, x >= 10 ancak < 20 (x, 10'dan büyük veya eşit ancak 20'den küçük"
  else:
    print("Merhaba Dünya, x >= 20 (x, 20'den büyük veya eşit)")
  return x + y

for i in range(8, 25, 5):  # i=8, 13, 18, 23 (başlangıç, bitiş, adım-artış-miktarı)
  print("--- Şimdi, i: {} için çalışıyor..".format(i))
  r = HelloWorldWithXY(i,i)
  print("HellloWorldWithXY fonksiyonun çıktısı : {}".format(r))
```

```
--- Şimdi, i: 8 için çalışıyor..
Merhaba Dünya, x  < 10 (x, 10'dan küçük)
HellloWorldWithXY fonksiyonun çıktısı : 16
--- Şimdi, i: 13 için çalışıyor..
Merhaba Dünya, x >= 10 ancak < 20 (x, 10'dan büyük veya eşit ancak 20'den kü
çük
HellloWorldWithXY fonksiyonun çıktısı : 26
--- Şimdi, i: 18 için çalışıyor..
Merhaba Dünya, x >= 10 ancak < 20 (x, 10'dan büyük veya eşit ancak 20'den kü
çük
HellloWorldWithXY fonksiyonun çıktısı : 36
--- Şimdi, i: 23 için çalışıyor..
Merhaba Dünya, x >= 20 (x, 20'den büyük veya eşit)
HellloWorldWithXY fonksiyonun çıktısı : 46
```

In [3]:

```
print(HelloWorldWithXY(1,2))
```

```
Merhaba Dünya, x  < 10 (x, 10'dan küçük)
3
```

Oldukça basit, öyle değil mi?

0'dan 3'e (3 dahil değil) bir döngü istiyorsanızi aşağıdaki kodlardan herhangi birini kullanabilirsiniz.

In [4]:

```
print("Öğeler üzerinde yineler. `range(3)` bir liste gibidir ve [0,1,2] olarak düşünülebili
for i in range(3):
  print(i)

print("Gerçek bir liste üzerinde yineleyelim.")
for i in [0,1,2]:
  print(i)

print("Aynı şekilde while döngüsü de çalışacaktır.")
i = 0
while i < 3:
  print(i)
  i += 1
```

```
Öğeler üzerinde yineler. `range(3)` bir liste gibidir ve [0,1,2] olarak düşü
nülebilir.
0
1
2
Gerçek bir liste üzerinde yineleyelim.
0
1
2
Aynı şekilde while döngüsü de çalışacaktır.
0
1
2
```

In [5]:

```
print("Python, devam et anlamına gelen continue ve ara-mola- anlamına gelen break gibi stan
while True:
  print("While döngüsüne girdi.")
  break
```

```
Python, devam et anlamına gelen continue ve ara-mola- anlamına gelen break g
ibi standart anahtar kelimeleri destekler.
While döngüsüne girdi.
```

Numpy ve Listeler

Python programlama dili yerleşik olarak listelere sahiptir. Ancak `Numpy`, makine öğrenimi yaparken yararlı olan birçok destek işlev (fonksiyon) sunar. Bu yüzden eğitim süresi boyunca `numpy` kütüphanesini kullanacağız.

Burada ayrıca bir `import` işlevi göreceksiniz. Bu ifade, tüm numpy paketlerini kullanılabilir hale getirir ve bu kütüphaneye kodda belirtilen kısaltma olan `np` sözdizimi kullanarak erişebiliriz.

In [6]:

```
import numpy as np  # np kullanarak numpy'ı kullanabiliriz.

# Numpy dizisi oluşturalım ve bir öğe ekleyelim.
a = np.array(["Merhaba", "Dünya"])
a = np.append(a, "!")
print("Bulunduğumuz dizi: {}".format(a))
print("Her eleman yazdırılıyor..")
for i in a:
  print(i)

print("\nHer eleman ve indeksleri yazdırılıyor..")
for i,e in enumerate(a):
  print("Indeks: {}, eleman (değer): {}".format(i, e))
```

```
Bulunduğumuz dizi: ['Merhaba' 'Dünya' '!']
Her eleman yazdırılıyor..
Merhaba
Dünya
!

Her eleman ve indeksleri yazdırılıyor..
Indeks: 0, eleman (değer): Merhaba
Indeks: 1, eleman (değer): Dünya
Indeks: 2, eleman (değer): !
```

In [7]:

```
print("\nDizilerle bazı basit matematik işlevlerini gösterelim.")
b = np.array([0,1,4,3,2])
print("Maksimum Değer: {}".format(np.max(b)))
print("Ortalama Değer: {}".format(np.average(b)))
print("Max. Değerin Bulundupu Indeks: {}".format(np.argmax(b)))
```

```
Dizilerle bazı basit matematik işlevlerini gösterelim.
Maksimum Değer: 4
Ortalama Değer: 2.0
Max. Değerin Bulundupu Indeks: 2
```

In [8]:

```
print("\nHer değer ve ifadenin türünü yazdırabilirsiniz.")
print("b'nin veri tiği: {}, b[0]'nin veri tipi: {}".format(type(b), type(b[0])))
```

```
Her değer ve ifadenin türünü yazdırabilirsiniz.
b'nin veri tiği: <class 'numpy.ndarray'>, b[0]'nin veri tipi: <class 'numpy.
int64'>
```

In [9]:

```
print("\nRastgele sayılarla [3,3] boyutlu bir dizi oluşturmak için Numpy kullanalım.")
c = np.random.rand(3, 3)
print(c)
```

```
Rastgele sayılarla [3,3] boyutlu bir dizi oluşturmak için Numpy kullanalım.
[[0.89812848 0.53768853 0.85742181]
 [0.09065575 0.34184737 0.108095  ]
 [0.42980187 0.4986246  0.76696916]]
```

In [10]:

```
print("\nDizilerin boyutlarını yazdırabilirsiniz.")
print("a'nın boyutu: {}".format(a.shape))
print("b'nin boyutu: {}".format(b.shape))
print("c'nin boyutu: {}".format(c.shape))
print("...Gözlem, Python listeleri belirtmek için hem [0,1,2] hem de (0,1,2) kullanır.")
```

```
Dizilerin boyutlarını yazdırabilirsiniz.
a'nın boyutu: (3,)
b'nin boyutu: (5,)
c'nin boyutu: (3, 3)
...Gözlem, Python listeleri belirtmek için hem [0,1,2] hem de (0,1,2) kullan
ır.
```

Colab Özellikleri

Colab, doğrudan erişebileceğiniz bir sanal makinedir. Komutları sanal makinenin terminalinde çalıştırmak için kod satırının önüne bir ünlem işareti (!) ekleyiniz.

In [11]:

```
print("\nDosya sisteminde $ls yapmak")
!ls -l
!pwd
```

```
Dosya sisteminde $ls yapmak
total 4
drwxr-xr-x 1 root root 4096 Jul 16 13:20 sample_data
/content
```

In [12]:

```
print("Numpy'ı yükle.")  # Yalnızca test için, numpy tüm Colab örneklerinde önceden yüklenm
!pip install numpy
```

```
Numpy'ı yükle.
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packag
es (1.19.5)
```

Colab tamamen ücretsizdir. Hatta günde birkaç saatlik kullanım için GPU bile ücretsiz olarak sağlanmaktadır.

GPU Kullanma

- Eğitim sürecinde alıştırmaların çoğu GPU kullanarak daha hızlı yürütülür. Bunuu yapmak için: Çalışma Zamanı | Çalışma zamanı türünü değiştir | Donanım hızlandırıyı GPU olarak seçip Kaydet demeniz yeterlidir.

Colab Hakkında Birkaç Bilgi

- Her hücreyi sırayla çalıştırabilir, isterseniz hücreleri düzenleyebilir ve yeniden çalıştırabilirsiniz.
- Bazen istenmeyen sonuçlarla karşı karşıya kalabilirsiniz. Örneğin, bir dizye bir boyut eklerseniz ve bu hücreyi birden çok kez çalıştırırsanız, sonraki hücreler çalışmayabilir. Sorunla karlışalmanız durumunda ortamınızı sıfırlayabilirsiniz:
- Çalışma zamanı -> Çalışma zamanını yeniden başlat. Bu Python kabuğunu sıfırlar.
- Colab'daki çıktıları şunu yaparak da temizleyebilirsiniz: Düzenle -> Tüm çıkışları temizle

In [12]: