

# Python 101

Ders 1:

- + Python Nedir ?
- + Değişkenler
- + Operatörler

# Ben Kimim ?

Lise öğrenimini tamamladıktan sonra 2020 yılında Celal Bayar Üniversitesi Bilgisayar Programcılığı bölümünü bitirdim. 2020 yılı Kasım ayında Ege Üniversitesi Teknopark bünyesinde bulunan RFPI Medikal Görüntüleme şirketinde Yazılım Geliştiricisi olarak 2 yıl çalıştım. Çalışmalarına EGE Solunum Araştırmaları Merkezi Laboratuvarlarında bulunan Doç.Dr.Özkan Doğanay önderliğindeki Fonksiyonel Akciğer Görüntüleme Laboratuvarında devam etmekteyim. Çalışma alanım : görüntü işleme, makine öğrenmesi, Akciğer BT görüntülerinden anatomik yapıların segmentasyon ve analizleri için algoritmalar geliştirmek ve başta covid-19 ve akciğer kanseri olmak üzere çeşitli akciğer hastalıklarının teşhisi ve incelenmesi. Alanımla ilgili çalışmalarından bir tanesi 2022 IEEE Medikal Görüntüleme Konferansına Akciğer Damar ağı segmentasyonu adlı çalışmamızla sunum olarak yer almaya hak kazanmıştır. 2022 yılında Yaşar Üniversitesi Yönetim Bilişim Sistemleri Bölümünü kazandım ve şu an 3.sınıf öğrencisi olarak eğitime devam etmekteyim. Aynı zamanda Fonksiyonel akciğer görüntüleme laboratuvarındaki çalışmalarına devam etmekteyim.



# Python Nedir? Python Neden Kullanılır ?

- ✓ Yüksek seviyeli, yorumlanan bir programlama dilidir.
- ✓ Guido van Rossum tarafından 1991 yılında yayınlanmıştır.
- ✓ Basit, okunabilir ve çok yönlüdür.
- ✓ Geniş bir standart kütüphane ve zengin bir üçüncü taraf kütüphane ekosistemi vardır.
- ✓ Açık kaynaklı bir dildir ve topluluk desteği geniştir.
- ✓ Daha önce programlama deneyimi olmayanlar için bile kolayca öğrenilebilir.
- ✓ Web geliştirme, bilimsel hesaplama, veri analizi, yapay zeka, makine öğrenimi gibi farklı alanlarda yaygın olarak kullanılmaktadır.
- ✓ Basit sözdizimi ve okunabilirliği sayesinde, kodlama sürecini hızlandırır ve hataların azaltılmasına yardımcı olur.
- ✓ Geniş bir standart kütüphane ve zengin üçüncü taraf kütüphane ekosistemi, farklı görevlerin hızlı ve kolay bir şekilde gerçekleştirilmesini sağlar.
- ✓ Bilimsel hesaplama ve veri analizi alanlarında, NumPy, Pandas ve Matplotlib gibi üçüncü taraf kütüphaneler sayesinde güçlü araçlar sunar.
- ✓ Yapay zeka ve makine öğrenimi gibi alanlarda, TensorFlow, Keras ve PyTorch gibi popüler kütüphaneleri ile kullanıcıların ihtiyaçlarına cevap verir.

## March Headline: Programming language Go enters TIOBE index top 10

Go has entered the TIOBE index top 10 this month. Engineers appear to love the Go programming language. What makes it stand out? Go is not revolutionary, but its strength is in combining the right features. It has built-in concurrency and garbage collection, is statically typed and has good performance. It also helps that Go is backed-up by the Google company. This improves the long-term trust in the language. Popular applications such as Docker and Kubernetes have been developed with the aid of Go. Will Go stay in the top 10? That is an interesting question, because position number 10 of the TIOBE index seems to change almost every month. -- *Paul Jansen CEO TIOBE Software*

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

# TIOBE INDEX

Mar 2023	Mar 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.83%	+0.57%
2	2			C	14.73%	+1.67%
3	3			Java	13.56%	+2.37%
4	4			C++	13.29%	+4.64%

# Değişkenler

## -Kurallar

---

- Python tanımlayıcısı, bir değişkeni, işlevi, sınıfı, modülü veya başka bir nesneyi tanımlamak için kullanılan bir addır. Bir tanımlayıcı, A'dan Z'ye bir harf veya a'dan z'ye veya bir alt çizgi (\_) ile başlar ve ardından sıfır veya daha fazla harf, alt çizgi ve rakam (0 ile 9) gelir.
- Python, tanımlayıcılarda @, \$ ve % gibi noktalama işaretlerine izin vermez.
- Python, büyük/küçük harfe duyarlı bir programlama dilidir. Dolayısıyla, Test ve test Python'da iki farklı tanımlayıcıdır.
- Python Sınıfı adları büyük harfle başlar. Diğer tüm tanımlayıcılar küçük harfle başlar.
- Bir tanımlayıcının başında tek bir alt çizgi ile başlatılması, tanımlayıcının özel tanımlayıcı olduğunu gösterir.
- Bir tanımlayıcının başında iki alt çizgi ile başlatılması, son derece özel bir tanımlayıcıyı gösterir.
- Tanımlayıcı ayrıca iki alt çizgi ile bitiyorsa, tanımlayıcı dil tanımlı özel bir addır.

# Değişkenler

## -Reserved Keywords

---

and	as	assert
break	class	continue
def	del	elif
else	except	False
finally	for	from
global	if	import
in	is	lambda
None	nonlocal	not
or	pass	raise
return	True	try
while	with	yield

# Değişkenler

## -Temel Veri Tipleri

---

### 1 . Sayılar (Numbers):

Sayılar, Python'da üç farklı tipte olabilir: tamsayılar (integer), kayan noktalı sayılar (float) ve karmaşık sayılar (complex).

```
x = 2
```

```
x = 2.0
```

```
x = 2.3j
```

# Değişkenler

## -Temel Veri Tipleri

---

### 2.Dizgeler (Strings):

- Dizgeler, bir ya da daha fazla karakterden oluşan veri tipleridir. Tek tırnak ( ' ') veya çift tırnak ( " ") içinde tanımlanırlar.
- String aslında bir dizidir

```
kelime = 'Merhaba'    # tek tırnaklı dizge
```

```
cumle = "Dünya güzeldir." # çift tırnaklı dizge
```



# Değişkenler

## -Temel Veri Tipleri

---

### 3.Listeler (Lists):

Listeler, birden fazla öğeden oluşan veri tipleridir. Öğeler, virgülle ayrılarak köşeli parantezler içinde tanımlanırlar.

```
sayilar = [1, 2, 3, 4, 5] # tam sayılar listesi  
meyveler = ['elma', 'armut', 'çilek'] # dizgeler listesi
```

# Değişkenler

## -Temel Veri Tipleri

---

### 4.Demetler (Tuples):

Demetler de listelere benzer ancak değiştirilemezler. Demetler, parantezler içinde tanımlanırlar.

```
noktalar = (2, 3) # 2 boyutlu demet
```

```
renkler = ('kırmızı', 'yeşil', 'mavi') # renkler demeti
```

# Değişkenler

## -Temel Veri Tipleri

---

### 5.Sözlükler (Dictionaries):

Sözlükler, anahtar-değer çiftlerinden oluşan veri tipleridir. Anahtarlar, değerlere erişmek için kullanılırlar. Sözlükler, süslü parantezler içinde tanımlanırlar.

```
ogrenciler = {'Ahmet': 'A', 'Mehmet': 'B', 'Ayşe': 'C'} # öğrenciler  
sözlüğü
```

```
sebzeler = {1: 'havuç', 2: 'pırasa', 3: 'patates'} # sayısal anahtarlarla  
sözlük
```

# Değişkenler

## -Temel Veri Tipleri

---

### 6.Boolean (Mantıksal):

Boolean, yalnızca iki değerden oluşan bir veri tipidir: True (Doğru) ve False (Yanlış). Boolean veri tipi, koşullu ifadeler ve döngüler gibi programlamada sıklıkla kullanılır.

```
x = True
```

```
y = False
```

# Değişkenler

## - Operatörler

---

### 1. Aritmetik Operatörler

+ Toplama  
- Çıkarma  
\* Çarpma  
/ Bölme  
% Mod alma (kalanı verir)  
\*\* Üs alma  
// Taban bölme

# Değişkenler

## - Operatörler

---

### 2.Atama Operatörleri

= Atama

+ = Toplayarak atama

- = Çıkararak atama

\* = Çarparak atama

/ = Bölerek atama

% = Mod alarak atama

\*\* = Üs alarak atama

// = Taban bölerek atama

# Değişkenler

## - Operatörler

---

### 3. Karşılaştırma Operatörleri:

== Eşit mi?

!= Eşit değil mi?

< Küçük mü?

> Büyük mü?

<= Küçük eşit mi?

>= Büyük eşit mi?

# Değişkenler

## - Operatörler

---

### 4.Mantıksal Operatörler:

and     Mantıksal VE

or     Mantıksal VEYA

not     Mantıksal DEĞİL



# Değişkenler

## - Operatörler

---

### 5. Kimlik Operatörleri:

**is**      **Kimlik kontrolü**

**is not** **Kimlik kontrolü**

# Değişkenler

## - Operatörler

---

### 6.Üyelik Operatörleri

**in** içinde mi?

**not in** içinde değil mi?

# Tip Dönüşümleri

---

Python'da tip dönüşümleri, bir veri tipini başka bir veri tipine dönüştürmek için kullanılır. Bazı durumlarda, bir veri tipinin başka bir veri tipine dönüştürülmesi gerekebilir. Bu işlem, özellikle kullanıcı tarafından verilen girdilerin veya alınan verilerin farklı veri tiplerinde olması durumunda önemlidir.

# Tip Dönüşümleri

---

`int()`: Bir sayı veya sayısal bir ifadeyi tam sayıya dönüştürmek için kullanılır.

```
a = "10"  
b = int(a)  
print(b)  
# çıktı: 10
```

# Tip Dönüşümleri

---

`float()`: Bir sayı veya sayısal ifadeyi ondalıklı sayıya dönüştürmek için kullanılır.

```
a = "3.14"  
b = float(a)  
print(b)  
# çıktı: 3.14
```

# Tip Dönüşümleri

---

`str()`: Bir veriyi, karakter dizisine dönüştürmek için kullanılır.

```
a = 10  
b = str(a)  
print(b)  
# çıktı: "10"
```

# Tip Dönüşümleri

---

`list()`: Bir veriyi, liste veri tipine dönüştürmek için kullanılır.

```
a = "Python"
b = list(a)
print(b)
# çıktı: ['P', 'y', 't', 'h', 'o', 'n']
```

# Tip Dönüşümleri

---

`tuple()`: Bir veriyi, demet veri tipine dönüştürmek için kullanılır.

```
a = ["elma", "armut", "çilek"]  
b = tuple(a)  
print(b)  
# çıktı: ('elma', 'armut', 'çilek')
```



# Tip Dönüşümleri

---

set(): Bir veriyi, kümeler veri tipine dönüştürmek için kullanılır.

```
a = [1, 2, 2, 3, 4, 4, 5]
```

```
b = set(a)
```

```
print(b)
```

```
# çıktı: {1, 2, 3, 4, 5}
```

# Tip Dönüşümleri

---

`tuple()`: Bir veriyi, demet veri tipine dönüştürmek için kullanılır.

```
a = ["elma", "armut", "çilek"]  
b = tuple(a)  
print(b)  
# çıktı: ('elma', 'armut', 'çilek')
```