



PROGRAMLAMA TEMELLERİ

Öğr. Gör. Erhan AKAGÜNDÜZ

VERİ YAPILARI

❑ Değişken ve Sabit Kavramları

- Programlama dillerinde ihtiyaç olduğu an ulaşılabilecek veri tutuculara **değişken** adı verilir.
- Değişkenler kutular olarak düşünülebilir.
- Kutular açılarak içinde ne olduğuna bakılabileceği gibi kutuların içine yeni bir şeyler de koyulabilir.
- **Değişkenler**, program içinde değeri değişebilen tutuculardır.

VERİ YAPILARI

- ❑ Değişken kavramını anlayabilmek için günlük hayattan örnekler verilebilir.
- ❑ 12 Dev Adam'ın bir maçını düşünün.
- ❑ Maçın o anki skoru 47-45 12 Dev Adam lehine olsun.
- ❑ Burada 47 sayısı değişkenin o anki değeridir.
- ❑ Cedi Osman'ın attığı üç sayılık isabetli atıştan sonra değişkenin yeni değeri artık 50 sayısıdır.

1	2	9
7	0	6
8	1	0
10	0	0
14	2	3
16	1	0
20	2	4
23	1	14
24	0	1
25	2	13
31	0	0
32	0	0

HOME

GUEST

50

POINTS

45

POINTS

TIME-OUT

03:39

TIME-OUT

BONUS
SERVICE

PERIODS

BONUS
SERVICE

10

FOULS

3

1

FOULS

N. PLAYERS	FOULS	POINTS
6	2	8
7	3	0
10		2
13	0	0
15	1	2
19	2	8
20	0	2
21	3	10
24	3	1
25	0	1
34	1	1
37	0	1

VERİ YAPILARI

- ❑ Değişkenler sadece ***tam sayıları*** değil; ***ondalıklı sayıları, metinleri, boolean (true / false)*** ifadelerini de hafızada tutabilir.
- ❑ Her değişkenin ***bir adı*** ve ***değeri*** vardır.
- ❑ Tekrar kutu örneğini hatırlayınız.
- ❑ Kutunun bir değişken olduğu varsayıldığında;
 - “***paket***” ifadesi ***değişken adı***, kutu açıldığında karşınıza çıkan “***kitap***” ise ***değişkenin değeridir***.

VERİ YAPILARI

- ❑ **Sabit**, kavramı ise uygulama çalıştığı sürece **değeri değişmeyen** veriler olarak ifade edilebilir.
- ❑ **Örneğin;**
 - $1 \text{ inç} = 2,54 \text{ cm}$
 - $\pi = 3,14$
- ❑ Bu gibi değeri değişmemesi gereken veriler sabit olarak tanımlanabilir.

DEĞİŞKEN TANIMLAMA

- ❑ Değişken tanımlamak için her programlama dilinde önceden belirlenmiş bazı kurallar bulunmaktadır.
- ❑ Python programlama dilinde değişken tanımlarken önce **değişken adı** yazılır.
- ❑ Değişken adı yazıldıktan sonra **= (eşittir)** işareti konulur ve **değişkenin değeri** yazılır.

değişken adı = değişken değeri

- ❑ **Ancak burada değişken isimlendirme kurallarına dikkat edilmelidir.**

DEĞİŞKEN TANIMLAMA

Değişken isimlendirirken hata mesajı ile karşılaşmamak için uyulması gereken kurallar şunlardır:

- ❑ Değişken isimleri **case sensitive** yani **büyük küçük harf duyarlıdır.**
- ❑ **Örneğin;** değişken isminin **adres** ya da **Adres** olması bu değişkenlerin farklı iki değişken olduğunu gösterir.
- ❑ Değişken isimlerinin anlaşılır olması işinizi kolaylaştırır.
- ❑ **Örneğin;** kullanıcıdan elektronik posta bilgisi alınacağı zaman bunu e-posta gibi anlaşılır bir değişken ismi ile ifade edebilirsiniz.

DEĞİŞKEN TANIMLAMA

- ❑ Değişken isimlendirilirken farklı standartlar kullanılmaktadır.
- ❑ Python'da genel kabul gören standart **Snake Case** standardıdır.
- ❑ Snake Case standardında değişken isimleri iki farklı kelimedenden oluşuyorsa alt tire (_) ile birleştirilir.
- ❑ **Lower Snake Case** ise tüm harflerin küçük harf olacağı anlamına gelir.
- ❑ Örneğin: **ev_adresi**, **kimlik_numarasi** vs.

DEĞİŞKEN TANIMLAMA

- ❑ Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir.
- ❑ Ancak sayılar başa gelmez.
- ❑ Örneğin **sayi1** doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme **değildir**.
- ❑ Değişken isimlendirilirken alt tire (_) kullanılabilir.
- ❑ Ancak boşluk ve diğer özel karakterler (?,%,!,., + vb.) kullanılmaz.
- ❑ Örneğin; **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan **hataya** neden olacaktır.

DEĞİŞKEN TANIMLAMA

- ❑ Değişken isimlendirilirken özel kullanım için ayrılmış olan **if, for, true** vb. ifadeler hata vermemesine rağmen özellikle kodların daha anlaşılır olması amacıyla **kullanılmamalıdır.**
- ❑ Bazı programlama dillerinde Türkçe karakterlerin (ç,ğ,ı,ö,ş,ü) kullanımı kabul edilirken bazılarında kabul edilmez.
- ❑ Python'da Türkçe karakterler kullanılması hataya neden olmaz.
- ❑ Ancak farklı programlama dillerinde problem yaşanmaması için değişken tanımlarken **Türkçe karakter kullanılmaması önerilmektedir.**

DEĞİŞKEN TANIMLAMA

Örnek: Kısa kenarı 3 cm, uzun kenarı 5 cm olan dikdörtgenin alanını hesaplayınız.

```
kisa_kenar = 3
uzun_kenar = 5
alan = kisa_kenar * uzun_kenar
print(alan)
```

15

OPERATÖRLER

- ❑ Veriler üzerinde işlem yaparak yeni değerler üretilmesini sağlayan programlama dili sembollerine **operatör** adı verilir.
- ❑ Python programlama diline yeni başlayanlar için;
 - aritmetiksel,
 - atama,
 - karşılaştırma,
 - mantıksal ve
 - kimlik operatörleri öğrenmek son derece önemlidir.

ARİTMETİKSEL OPERATÖRLER

Operatör	Tanımı	Örnek
+	Toplama	$a+b$
-	Çıkarma	$a-b$
*	Çarpma	$a*b$
/	Bölme	a/b
%	Mod alma (Bir sayının diğer sayıya bölümünden kalan)	$a\%b$
**	Kuvvet alma (ab)	$a**b$
//	Tam sayı bölme (Bölme işleminde sadece tam kısım alınır.)	$a//b$

ATAMA OPERATÖRLERİ

Operatör	Örnek	Açıklama
=	a=2	a değişkenine 2 değeri atanmıştır.
+=	a+=2	a değişkenine 2 değerini ekleyerek yine a değişkenine atanmıştır. Başka bir ifadeyle a=a+2 anlamına gelmektedir.
-=	a-=2	a değişkeninden 2 değeri çıkarılarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a-2 anlamına gelmektedir.
=	a=2	a değişkeni 2 ile çarpılarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a*2 anlamına gelmektedir.
/=	a/=2	a değişkeni 2 değerine bölünerek yine a değişkenine atanmıştır. Başka bir ifadeyle a=a/2 anlamına gelmektedir.
%=	a%=2	a değişkenin 2 değeri ile modu alınarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a%2 anlamına gelmektedir.
=	a=2	a değişkeninin ikinci kuvveti (a ²) alınarak yine a değişkenine atanmıştır. Başka bir ifadeyle a=a**2 anlamına gelmektedir.
//=	a//=2	a değişkeni 2 değerine tam bölünmüş (kalan dikkate alınmadan) ve çıkan değer yine a değişkenine atanmıştır. Başka bir ifadeyle a=a//2 anlamına gelmektedir.

KARŞILAŞTIRMA OPERATÖRLERİ

Operatör	Tanımı	Örnek
==	Eşittir	a==b
!=	Eşit değildir	a!=b
<	Küçüktür	a	Büyüktür	a>b
<=	Küçük eşittir	a<=b
>=	Büyük eşittir	a>=b

MANTIKSAL OPERATÖRLER

Operatör	Örnek	Açıklama
and	$a < 3$ and $b \geq 5$	İki veya daha fazla şartın tamamının doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeni 3'ten küçük ve b değişkeni 5'e eşit ya da 5'ten büyük olursa True değeri döndürülür.
or	$a < 3$ or $b > 4$	İki veya daha fazla şartın en az birinin doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeninin 3'ten küçük olması ya da b değişkeninin 4'ten büyük olması True değeri döndürmek için yeterlidir.
not	not($a < 3$)	Durumu tersine çevirmek (True ise False; False ise True) için kullanılır. Buradaki örnekte parantez içindeki mantıksal sınamanın sonucu tersine çevrilir. İfadenin not komutu olmadan yazıldığında true döndüreceği varsayıldığında bu haliyle false döndürecektir.