

# Flutter ile Uygulama Geliştirme Kursu | Android & IOS

## Değişken ve Veri Tipleri

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

Android - IOS Developer and Trainer

# Eğitim İçeriği

1. Değişkenler
2. print() metodu
3. Constant – Sabitler
4. Aritmetik Operatörler
5. Tür Dönüşümü
6. Konsol Girdisi

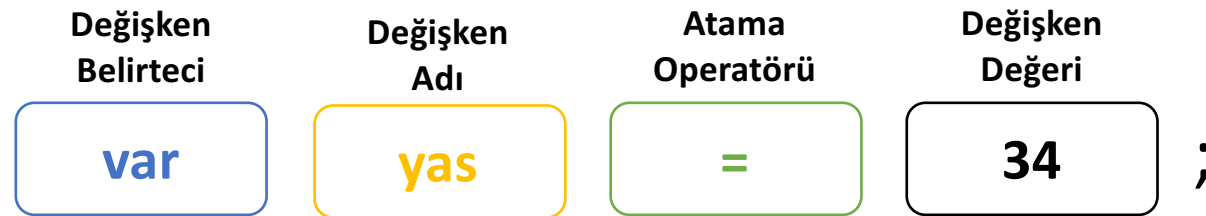
Variables - Değişkenler

# Değişkenler

- Modern diller hafızada saklanan değerleri değişkenler ile ifade etmektedir.
- Değişkenler hafızada geçici olarak saklanan değerleri temsil eder .
- Dart dilinde değişken için **tür belirtmemize** gerek yoktur.

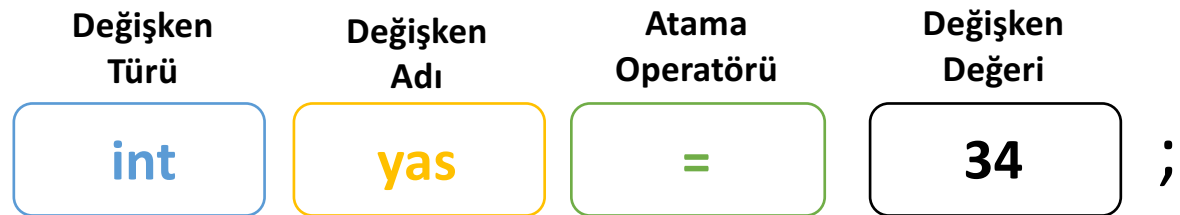
**Not : Değişkenler kalıcı değildir.  
Programdan çıkıldığında değerler kaybolur. Kalıcı değerler için  
değişkenlerin değerleri diske yazılmalıdır.**

## Değişken oluşturma



**var** yas = 34 ;

## Tür Belirterek Değişken oluşturma



```
int yas = 34 ;
```

# Data Tipleri

## Tam Sayılar

Int

## Ondalıklı Sayılar

Double

## Metinsel İfadeler

String : Yazılar ve Harfler

## Mantıksal İfadeler

Bool: True veya False

# Literals – Değerlerin Yazılma Kuralları

- Literals değişkenler için kullanılan değerlerin nasıl yazılması gerektiğini temsil eder.

<u>"Ahmet"</u>	<u>//Metinsel ifade ( String )</u>
<u>"A"</u>	<u>//Harfsel ifade</u>
<u>23</u>	<u>//Tam Sayı ( int )</u>
<u>1.45</u>	<u>//Odalıklı Sayı ( double )</u>



# Değişkenlere isim verme kuralları

- Case sensitive'dir. Büyük küçük harf farkı vardır.
  - Rakamla başlayamaz.
- @ ve % değişken içerisinde kullanılmaz.
- Bazı örnekler ;

Azad	zara	abc	move_name	a_123
myname50	_temp	j	a23b9	retVal

## Örnek 1:

- Bir öğrencinin **adını** , **yaşını** , **boyunu** ve **adının baş harfinin** tutulduğu değişken oluşturunuz.

## Örnek 2:

- Bir şirketin ürünlerinin bilgilerinin tutulduğu ürünler tablosunu temsil eden değişkenleri oluşturunuz.

ürün_id	ürün_adi	ürün_adet	ürün_fiyat	ürün_tedarikci
3416	Kol saati	100	149.99	rolex

## print() metodu ile çıktı alma

- Kodlama yaparken kodların çalışma sonuçlarını bu metod ile takip edebiliriz.
  - print() alt alta yazmak için kullanılır.

String ifade içine **\$** ifadesi kullanılarak çıktıya değişken eklenebilir.

```
var ad = "Ahmet";
```

```
int yas = 10;
```

```
print("$ad Bursada $yas yıldır yaşamaktadır.");
```

**Ahmet Bursada 10 yıldır yaşamaktadır.**

# Değişkenleri Yazdırma

String ifade içine **`${}`** ifadesi kullanılarak işlem yapılabilir.

```
int a = 10;
```

```
var b = 20;
```

```
print("$a ve $b nin toplamı : ${a+b}");
```

10 ve 20 nin toplamı : 30 dir.

# Değişken Oluşturma Çeşitleri

*//Örnek 1*

```
var sayi = 10 ;  
int numara = 88 ;
```

*//Örnek 2 : Değişkenin değerini daha sonra değiştirebiliriz*

```
var fiyat = 12.99 ;  
double ucret = 23.59;
```

```
print(fiyat);  
print(ucret);
```

```
fiyat = 67.99;  
ucret = 10.45;
```

```
print(fiyat);  
print(ucret);
```

*//Örnek 3 : İşlem yaptıktan sonra başka bir değişkene veri aktarılabilir.*

```
var sayi1 = 10 ;  
int sayi2 = 20 ;
```

```
var toplam = sayi1 + 50 ;
```

```
int carpma = sayi1 * sayi2 ;
```

```
print(toplam);  
print(carpma);
```

*//Örnek 4 : Boş değişken oluşturup daha sonra değer aktarabiliriz.*

```
var sonuc1;  
int sonuc2;
```

```
sonuc1 = 333;  
sonuc2 = 555;
```

```
print(sonuc1);  
print(sonuc2);
```

*//Örnek 5 : Yan yana değişken oluşturma*

```
var s1,s2,s3;
```

```
s1 = 10;  
s2 = 20;  
s3 = 30;  
print("$s1 $s2 $s3");
```

```
int k1 = 44 ,k2 = 78 ;  
print("$k1 $k2");
```

*//Örnek 6 : Type Safety*

```
var deger = 100;  
deger = "merhaba";
```

## Type Safety – Tür Güvenliği

- Oluşturduğunuz değişkene farklı türde değişken atayamazsınız.

```
var deger = 100;  
deger = "merhaba";
```

# Değişkenin kapsamı (Global ve Local Değişken )

- Süslü parantez { } bizim kapsamımızı belirler. Değişkenin ulaşılabilirliği buna bağlıdır.

```
class Deneme {  
    int x = 10; //Global Değişkenler  
    int y = 20; //Global Değişkenler  
  
    void topla(){  
        int x = 40; //Local değişken Global değişkene baskın gelir.  
  
        x = x + y; // 40 + 20 = 60  
        print(x);  
    }  
  
    void carpma(){  
        x = x * y; // 10 * 20 = 200  
        print(x);  
    }  
}
```



Constant - Sabitler

# Constant - Sabitler

- Sabitler içerisine bir kere veri atıldığında bir daha değiştiremeyeceğiniz yapılardır.
- **final** ve **const** ismi ile kullanılırlar.
- **const** değişken oluşturduğumuz anda hafızada oluşur.
- **final** değişken oluşturulduktan sonra kod çalıştırıldığında hafızada oluşur.
- **const** sınıf içinde kullanılmaz.
- Genelde yaygın kullanım **final** dır.
- Sabit kullanmak memory yönetimini rahatlatır.
- Çünkü hafızada sabit için yer ayrılır ve değişim olmayacağı için açılan yer yeni bir değer almak için beklemez.
- Sadece kullanılma amaçlı değişkenler için kullanılması için uygundur.

```
final pi = 3.14;  
final String mesaj = "merhaba";  
const y = 20 ;  
const double a = 20.9 ;
```

# Kaçış Karakterleri

- Kaçış karakterleri String ifade içine bazı karakterleri yazmamızı sağlarlar.
- Bunun çıkış sebebi String ifadelerin **"** işareti ile başlayıp bitmesidir.

- En çok kullanılan kaçış karakterleri.

- **\\** – \ işareti
- **\t** – Bir tab boşluk bırakır
- **\n** – Bir alt satıra iner
- **\"** – Çift tırnak işareti
- **\'** – Tek tırnak işareti

```
var yazil = "Merhaba bu \"flutter\"\n\teğitiminde \\dart\\ dilini öğrenicez";  
print(yazil);
```

```
Merhaba bu "flutter"  
eğitiminde \dart\ dilini öğrenicez
```

# Yorum Satırları

- Yorum satırı kullanımının birçok amacı vardır.
- Kodunuza anlaşılır notlar yazmak.
- Bazı kod satırını geçici olarak gizlemek için kullanılabilir.

- **Satıra yorum ekleme ;**

// işareti ile yapılır.

Örn : // Bu nesne ile veri tabanına erişilebilir.

- **Blok yorumu ekleme;**

/\* ile açılır \*/ kapatılır.Tek satır değil birden fazla satır için kullanılabilir.

Örn : /\* Açıklama

Veri tabanı için gerekli nesneleri kullanmalıyız.

Bazı nesneler nil dönebilir dikkatli olunmalıdır. \*/

# Aritmetik Operatörler

- Matematiksel işlemleri yapmamızı sağlarlar.
- Parantezler işlemin önceliğini belirtmek için kullanılır.
  - Örn : A = 10 ve B = 20 olsun

Operator	Açıklama	Örnek
+	Toplama	$A + B = 30$
-	Çıkarma	$A - B = -10$
*	Çarpma	$A * B = 200$
/	Bölme	$B / A = 2$
%	Mod İşlemi	$B \% A = 0$

Örnekler : Aşağıdaki formülleri tanımlayınız.

- Daire alanını değişkenler oluşturarak hesaplayınız.
- $F = m \times a$  Uygulanan Kuvvet(F)= Cismin kütlesi(m) x cismin ivmesi (a)

$$\Delta x = \left( \frac{v + v_0}{2} \right) t$$

$$\Delta x = v_0 t + \frac{1}{2} a t^2$$

# Atama Operatörlerinin Kısaltımı

- Atama işlemlerini kolaylaştırma amaçlı kullanılırlar.
- Aritmetik operatörlerin hepsinde geçerlidir.
- Normal ifade ;

•  $a = a + 3$

$b = b * 3$

$c = c - 3$

$d = d / 3$

• **Kısayol** :  $a += 3$

**Kısayol** :  $b *= 3$

**Kısayol** :  $c -= 3$

**Kısayol** :  $d /= 3$

```
int y = 10;
```

```
y = y + 2;
```

```
y+=2;
```

```
print(y); //14
```

# Tür Dönüşümü

1. Sayısalardan sayısal dönüşüm
2. Sayısalardan metne dönüşüm
3. Metinden sayısal dönüşüm

- **toDouble() , toInt() , toString() , int.parse() , double.parse()**



## Sayısalardan sayısal dönüşüm

```
int i = 42;
```

```
double d = 42.45;
```

```
int sonuc1 = d.toInt();
```

```
double sonuc2 = i.toDouble();
```

## Sayısalardan Metne Dönüşüm

```
int i = 42;
```

```
double d = 42.45;
```

```
String str1 = i.toString();
```

```
String str2 = d.toString();
```

# Metinden Sayısala Dönüşüm

- Dönüşüm olurken dikkatli olunmalıdır çünkü metin içinde her zaman sayı yer almaz hata ihtimali yüksektir.

```
String yazi1 = "34";
```

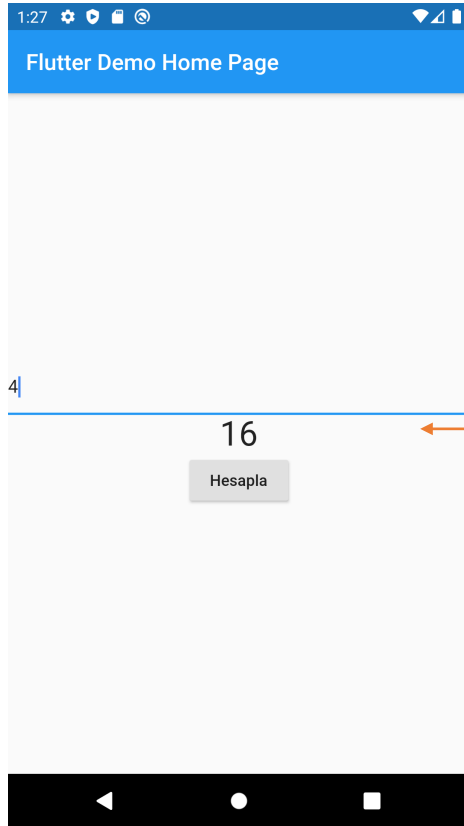
```
String yazi2 = "34.67";
```

```
int s1 = int.parse(yazi1);
```

```
double s2 = double.parse(yazi2);
```

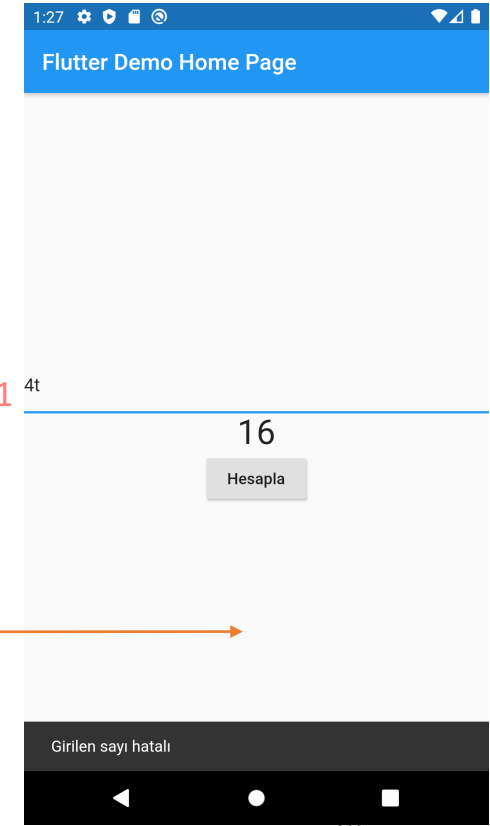
# Android Kullanım Alanı

## 1. Durum



```
TextField(  
  controller: tfController,  
  decoration: InputDecoration(hintText: "Veri"),  
) , // TextField  
Text(sonuc, style: TextStyle(fontSize: 30)),  
RaisedButton(child: Text("Hesapla"),  
  onPressed: () {  
    setState() {  
      String gelenVeri = tfController.text;  
  
      try{  
        int sayi = int.parse(gelenVeri); //String to Int  
  
        int hesap = sayi * sayi ;  
  
        sonuc = hesap.toString(); //Int to String  
  
      } catch(e){  
        scaffoldKey.currentState.showSnackBar(  
          SnackBar(content: Text("Girilen sayı hatalı")),  
        );  
      }  
    }  
  }  
), // RaisedButton
```

## 2. Durum



# Konsol Girdisi

```
import 'dart:io';

void main(){

    print("Adınızı Giriniz ");

    String isim = stdin.readLineSync()!;
    print("Adınız : $isim");

}

print("1. Sayıyı Giriniz ");
int sayi1 = int.parse(stdin.readLineSync()!);

print("2. Sayıyı Giriniz ");
int sayi2 = int.parse(stdin.readLineSync()!);

print("Toplam : ${sayi1+sayi2}");
```

Sadece string türünde girdi olabilir.

`stdin` import edilmelidir.

`readLineSync()` metodu ile veri konsoldan okunur.

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan