

Ardunio

Sabitler

- Arduino dilinde önceden tanımlanmış ifadelerdir. Programların okunmasını kolaylaştırmak için kullanılırlar.
 - HIGH/LOW
 - INPUT/OUTPUT
 - Led BuiltIn
 - True/False

HIGH | LOW

```
int led= 13;  
digitalWrite(led, HIGH); // Ledi yakılıyor  
digitalWrite(led, LOW);  // Ledi söndürülüyor
```

Okuma veya yazma yaparken ***dijital pine verilen aktif veya/pasif olma durumunu ifade eder.***

HIGH ile pin çıkışı aktif edilirken, LOW ile pin çıkışı pasif yapılmaktadır. Pim aktif hâle getirildiğinde (HIGH yapıldığında) 5 Volt ile çalışan kartlarda 3 volttan daha yüksek bir voltaj mevcutken 3.3 Volt ile çalışan kartlarda 2 volttan daha yüksek bir voltaj bulunmaktadır. LOW durumunda ise (LOW yapıldığında) 5 Volt ile çalışan boardlarda 3 volttan daha düşük bir voltaj mevcutken 3.3 Volt ile çalışan boardlarda 2 volttan daha düşük bir voltaj bulunmaktadır.

INPUT | OUTPUT

```
int led=13;  
int buton=4;  
void setup()  
{  
  pinMode(led, OUTPUT); // Çıkış olarak tanımlandı  
  pinMode(buton, INPUT); // Giriş olarak tanımlandı  
}
```

- Temelde dijital pine verilen modunun giriş ya da çıkış olacağı belirlenmektedir.
- Dijital pinler INPUT veya OUTPUT olarak kullanılabilir. Dijital pinlerin elektriksel davranışı pinMode() ile değiştirilebilir.

LED_BUILTIN

- Arduino kartlarının bir çoğu, bir dirençle seri olarak bağlı bir LED'in bulunduğu bir pime sahiptir.
- Sabit LED_BUILTIN, yerleşik LED'in bağlandığı pinin numarasıdır. Genellikle bu LED dijital pin 13'e bağlanmıştır.

true | false

Arduino da doğruyu ve yanlış belirtmek için kullanılan iki sabit mantıksal tanımlamadır. Yanlış false 0 (sıfır) olarak tanımlanır. Doğru ise true 1 olarak tanımlanır. Ancak doğru olarak tanımlamanın daha geniş bir anlamı vardır. Boolean anlamında, sıfır olmayan herhangi bir tam sayı doğrudur. Dolayısıyla -1, 2 ve -200 tümüyle Boolean anlamında doğru olarak tanımlanır.

```
int m = true; // m doğru
int n = false; // n yanlış
int led=13; // led 13. pine tanımlandı
int buton=12; // buton 12. pine tanımlandı
void setup() { } //ana kurulum yapıldı
void loop() { // sonsuz döngü sağlandı
  if (buton == m) { // butona basıldı mı? Evet ise
    digitalWrite(led, HIGH); // led e 5v ver
    delay(1000); //1 saniye bekle
    digitalWrite(led, LOW); } //led söndür
  }
}
```

Veri Tipleri

- **boolean**
- **char**
- **unsigned char**
- **byte**
- **int**
- **unsigned int**
- **long**
- **unsigned long**
- **short**
- **float**
- **double**
- **string – char array**

void

Void anahtar sözcüğü yalnızca işlev bildirimlerinde sadece **fonksiyon tanımlanırken** kullanılır. Bu işlevin çağrıldığı işleve herhangi bir bilgi döndürmemesi beklenmez. Yani fonksiyonun değer döndürmeyeceği anlamına gelir.

```
// Eylemler ve fonksiyonlar "kurulum" ve  
// "döngü" içerisinde gerçekleştirilir.  
// Ancak hiçbir bilgi yada program bildirilmez  
void setup() {  
    // ...  
}  
void loop() {  
    // ...  
}
```

boolean

Bir boolean doğru veya yanlış olmak üzere iki değerden birini tutar. 0 veya 1 değerlerini “true” ve “false” olarak alır.

```
int LEDpin = 5;          // LED 5 nolu pine bağlanıyor
int switchPin = 13;      // anahtar 13 numaralı pine bağlanıyor
boolean running = false;
void setup()
{
    pinMode(LEDpin, OUTPUT);
    pinMode(switchPin, INPUT);
    digitalWrite(switchPin, HIGH);
}
void loop()
{
    if (digitalRead(switchPin) == LOW)
    { // anahtara basıldığında normal olarak çalışıyor
        delay(100);                // anahtar için gecikme sağlanıyor
        running = !running;        // değişkenler arasında geçiş yapılıyor
        digitalWrite(LEDpin, running); // LED yakılıyor
    }
}
```

char

Bir karakter değeri saklayan (1 bayt bellek alan) veri tipidir.

Karakter verisini tanımlamak için kullanılmaktadır.

Karakter harfleri, tek tırnak işaretleriyle yazılır: 'A' (birden fazla karakter için çift tırnak kullanılır: "ABC").

Ancak karakterler sayı olarak saklanır. (-128 ile +127 arası değer alır.)

```
char myChar = 'A';  
char myChar = 65;    // Her ikiside eşdeğerdir
```

unsigned char

1 baytlık belleği kaplayan işaretsiz bir veri türüdür. Bayt veri türü ile aynıdır. (0-255 arası değer alır.)

```
unsigned char myChar = 230;
```

byte

Bir bayt, 0'dan 255'e kadar 8 bitlik bir işaretsiz sayı verisi taşır.

```
byte b = B10010; // "B" binary biçimi (B10010 = 18 decimal)
```

int

Tamsayıları saklamak için kullanılan birincil veri türüdür. 16 bit işlemcilerde -32,768 ile 32,767 arası 32 bit işlemcilerde ise -2,147,483,648 ile 2,147,483,647 arasında değişen veri saklanabilir.

```
int ledPin = 13;  
  
int var = val;  
//var -int değişken adı  
//val -o değişkene atanan değer
```

unsigned int

Negatif sayıları saklamak yerine 16 bitlik işlemcilerde sadece 0 ile 65,535 arasında değişen 2 baytlık (16 bit) bir pozitif değeri saklar. 32 bit işlemcilerde 0 ila 4,294,967,295 arasında değişen 4 baytlık (32 bit) bir pozitif değeri saklar.

```
unsigned int ledPin = 13;  
  
unsigned int var = val;  
//var -işaretsiz int değişken adı  
//val -o değişkene atanan değer
```

long

Sayı saklamak için genişletilmiş boyut değişkenleridir ve 32 bit (4 bayt), -2,147,483,648 ile 2,147,483,647 arasında değişen değeri saklar.

Tamsayılar kullanılıyorsa sayıların en az biri “long” olmalı ve bir “L” tarafından takip edilmelidir.

```
long speedOfLight = 186000L;  
  
long var = val;  
// var -long değişken adı  
// val -o değışkene atanan değer
```

unsigned long

Unsigned long değişkenler sayı saklamak için genişletilmiş boyut değişkenleridir ve 32 bit (4 bayt) depolamaktadır. Standart "long"un aksine, unsigned long 0 ile 4,294,967,295 arasında değişen pozitif sayıları saklar, negatif sayıları saklamaz. unsigned long değişken adı = o değişkene atayan değer olarak yazılır.

```
unsigned long time;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.print("Time: ");
  time = millis();
  // program başlatıldıktan sonra zamanı yazdırır
  Serial.println(time);
  // bir saniye bekleniyor
  delay(1000);
}
```

short

Short, 16 bitlik (2 baytlık) bir veri türüdür. -32,768 ile 32,767 arasında değişen değeri saklar.

```
short ledPin = 13;  
  
short var = val;  
// var -short değişken adı  
// val -o değışkene atanan değer
```

float

Ondalıklı sayılar için kullanılan veri türüdür. Ondalıklı sayılar, tamsayılara göre daha yüksek çözünürlüğe sahip oldukları için genellikle analog ve sürekli değerleri yaklaştırmak için kullanılırlar. Ondalık sayıları $3.4028235E + 38$ ile $-3.4028235E + 38$ arasında olabilirler. Bunlar 32 bit (4 bayt) bilgi olarak saklanırlar.

```
float myfloat;  
float sensorCalbrate = 1.117;  
  
float var = val;  
// var -float değişken adı  
// val -o değişkene atanan değer
```

double

Double, ondalıklı sayılar için kullanılan veri türüdür.

Burada hassasiyet 2 kat yüksektir.

“double” uygulaması tam olarak “float” ile aynıdır.

Uno ve diğer ATMEGA tabanlı kartlarda 4 bayt yer kaplar.

Arduino Due'da “double” 8 bayt (64 bit) hassaslığa sahiptir.

array

Bir dizi, bir dizin numarasıyla erişilen değişkenlerin bir toplamıdır.

Diğer bir ifadeyle her birine indeks numarası ile ulaşılan aynı türdeki veri topluluğudur. Dizi, bilgisayar belleğinde aynı isim altında genellikle aynı tipten çok sayıda veriyi bir arada saklayan veri yapısıdır.

mySensVal=

```
int myInts [6];  
int myPins [] = {2, 4, 8, 3, 6};  
int mySensVals [5] = {2, 4, -8, 3, 2};  
char message [8] = "merhaba";  
  
mySensVals [0] = 10; // Bir diziye değer atamak  
x = mySensVals [4] ; // Bir diziden bir değer almak
```

string - char array

Yazı verisi depolamak için kullanılır. Karakter dizisidir. Metin dizeleri string ile gösterilir. Yandaki gösterimlerin tümü geçerli kullanım şekillerine örnektir.

```
char Str1 [15];  
char Str2 [8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o'};  
char Str3 [8] = {'a', 'r', 'd', 'u', 'i', 'n', 'o', '\\ 0'};  
char Str4 [] = "arduino";  
char Str5 [8] = "arduino";  
char Str6 [15] = "arduino";
```