

Free Java Dersleri

Ders - 02

Data Nedir?
Java'da Data Turleri

Data Nedir ?

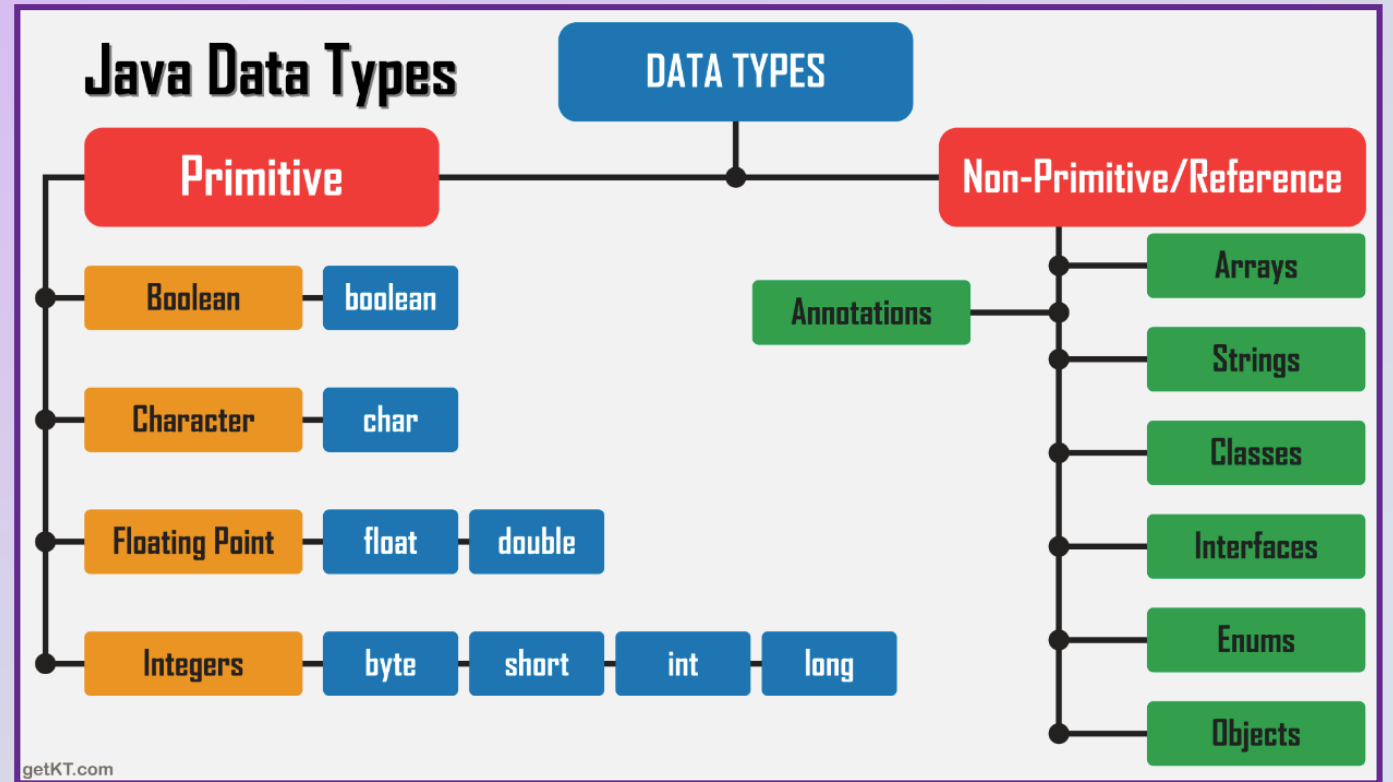
Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things.

Data (**Veri**), sayılar, kelimeler, ölçümler, gözlemler gibi bilgi içeren objelerin bir koleksiyonudur.

Yazacağımız her kod, yapacağımız her program **data**'yi almak, **data**'yi işlemek ve sonuç olarak bir **data** oluşturmak için kullanılır.

Yukarıda da tanımlandığı gibi data'nin içerdigi bilgi çok farklı olabileceğinden, tüm programlama dilleri farklı data türlerini kullanabilmek için **kendi kullanacakları data türlerini** tanımlamışlardır.

Hangi programlama dilini kullanacaksak, öncelikle o dillerde kullanabileceğimiz data türlerini öğrenmeliyiz.



Data Hafizada Nasil Saklanir (Store) ?

Her data hafizada(memory) bir yer kaplar.

Bir datanın hafizada saklanacagi en kucuk bolum bit'dir.

Her bir bit **1** veya **0** degerlerini icerir.

8 bit bir araya geldiginde bir **byte** olur.

Her **byte** $2^8 = 256$ farkli deger alabilir.

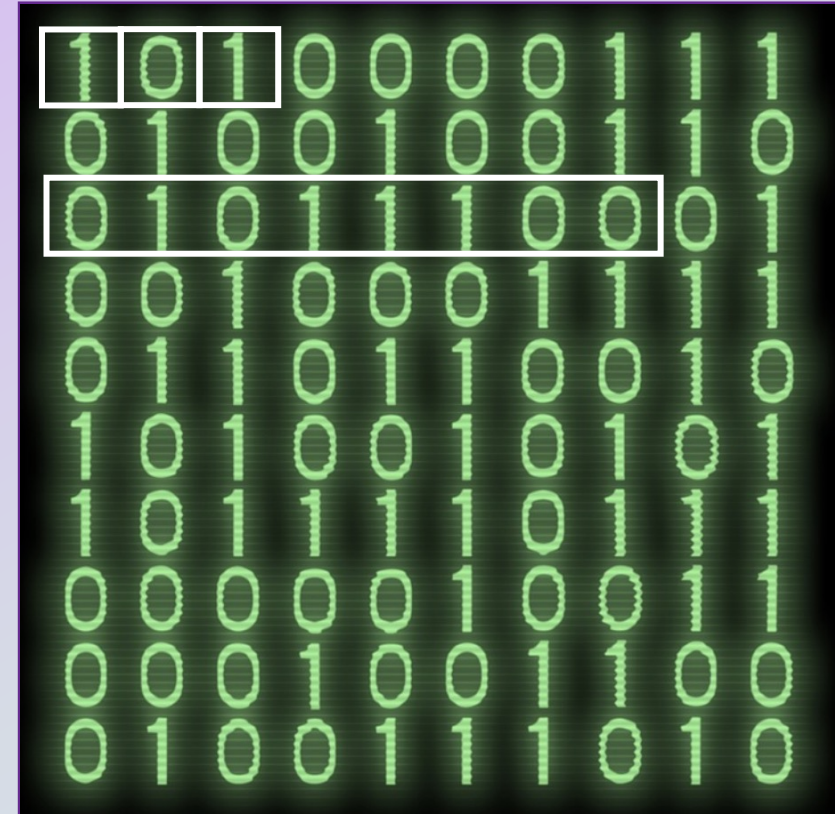
Sayi sistemimiz 10'luk sistem oldugu gibi hafiza da $2^{10}=1024$ 'un katlari seklinde yapilandirilmistir.

1024 byte = 1 KB

1024 KB = 1 MB

1024 MB = 1 GB

1024 GB = 1 TB



Variables (Java Datalari Nasil Kullanir) ?

Java hafizadaki datalari **variable** veya **objeler** yardimiyla kullanir.

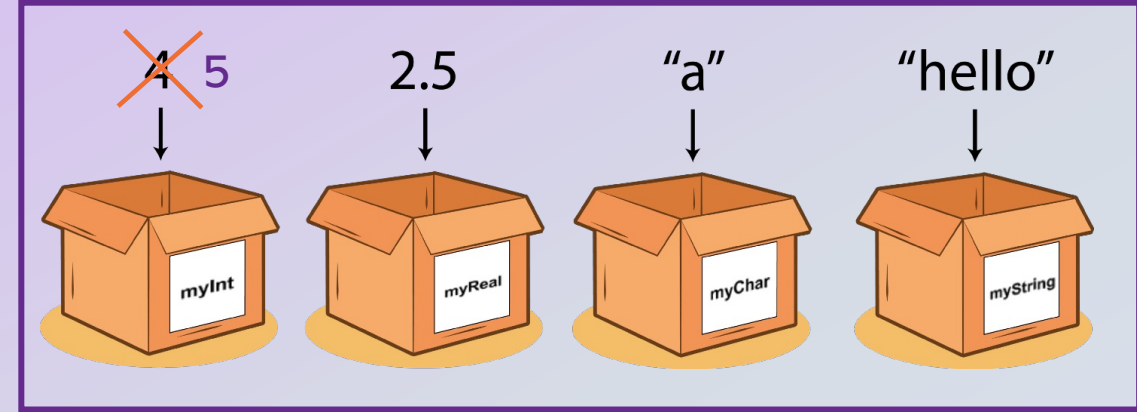
Bir datayi hafizada store etmek istedigimizde, Java hafizada o datayi tutabilecegi bir alan ayirir ve o alani isimlendirir.

Biz ne zaman o data uzerinde degisiklik yapmak istesek, ismini soylememiz yeterli olur.

Ornegin myInt'i bir artir dedigimizde, Java myInt ismindeki variable'i bulur, icindeki deger olan 4'u bir artirip 5 yapar.

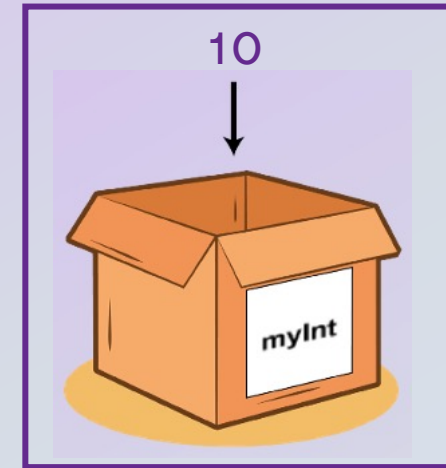
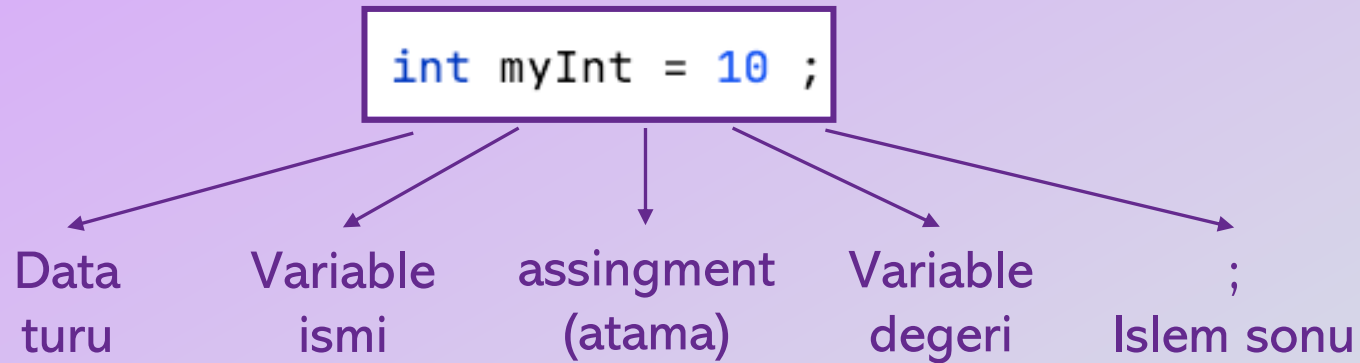
Bu islemden sonra myInt variable'inin degeri 5 olacaktır.

Ozetle; biz bir degeri store etmek istedigimizde data turune uygun bir variable olusturup ona bir isim veririz, ne zaman o datayi kullanmak istesek Java'ya variable ismini soylememiz yeterli olacaktır. myInt'i artir, myInt'i degistir, myInt'i sil vb...



Variable Nasıl Olusturulur ?

Variable olusturmak ve deger atamak icin Java'nin belirledigi syntax asagidaki gibidir.



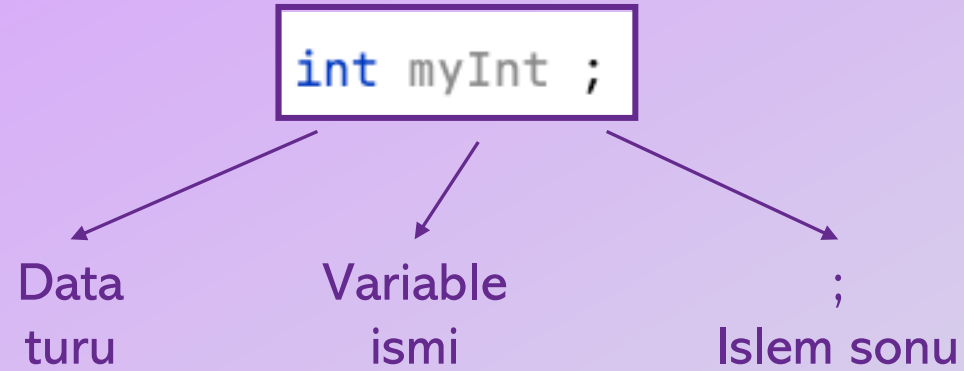
Bir variable olusturmak icin 2 islem vardır;

1- declaration (tanımlama) : esitligin sol tarafi

2- deger atama (assignment) : esitlik ve esitligin sol tarafi

Variable Declaration

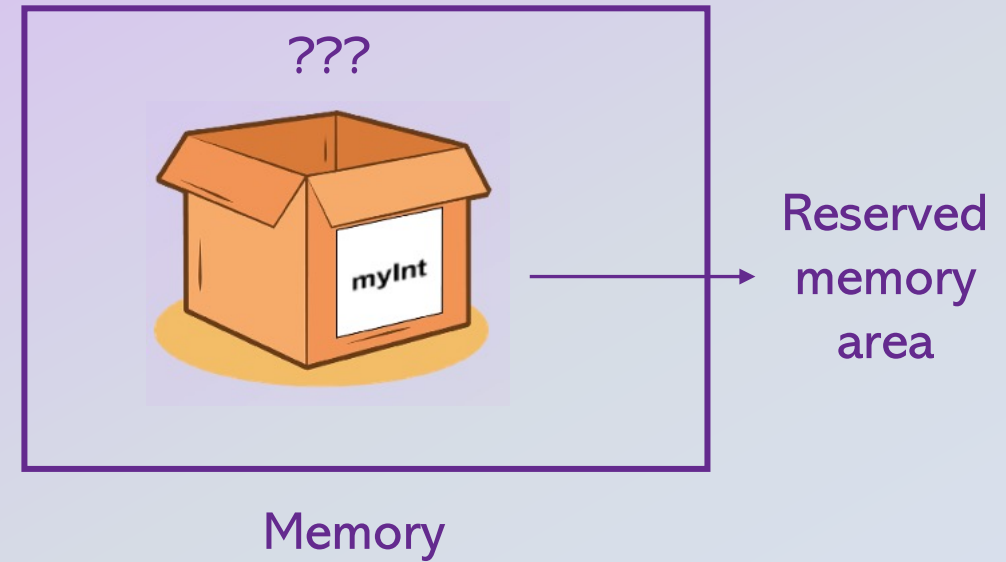
Java'nin datayı store edebilmesi için variable'a ihtiyacı vardır. Bir variable'ın oluşturulması için de mutlaka declaration gereklidir.



Declaration için data turu ve variable ismi yeterlidir.

Java'da declaration ve assignment farklı satırlarda yapılabilir.

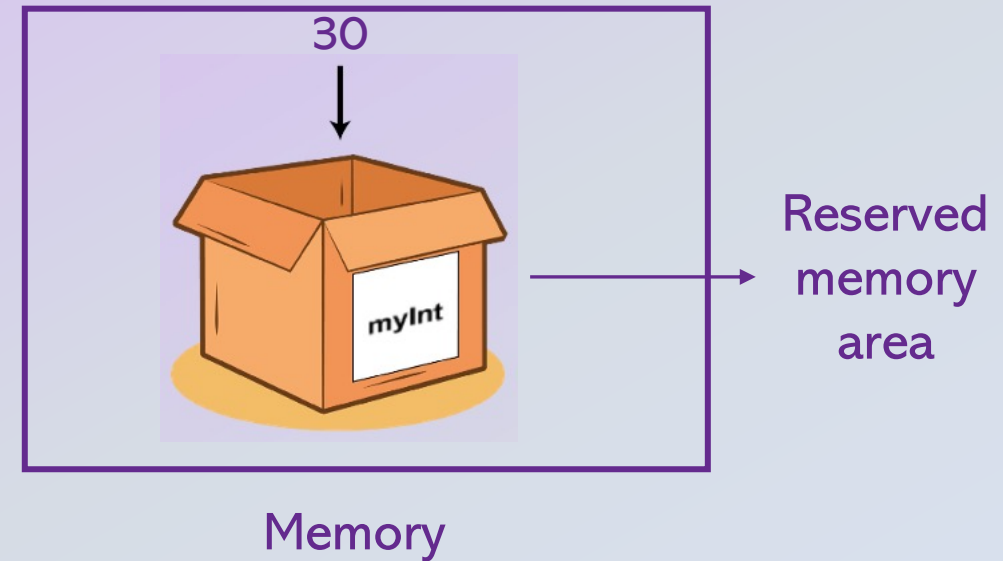
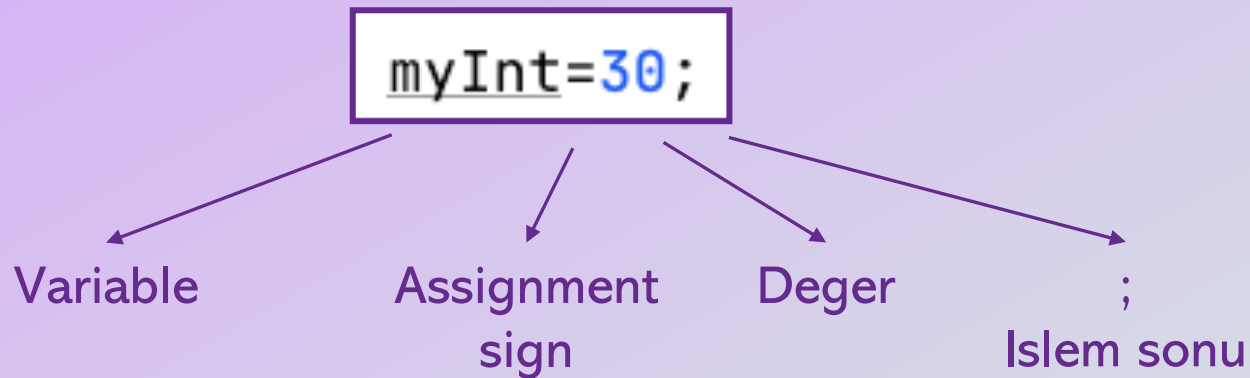
Ancak bir değer ataması olmadan variable'ın kullanılması mümkün değildir.



Variable Deger Atamasi (Assignment)

Deklare edilmiş bir variable'a deger atamaya assignment denir.

Declaration ve assignment farkli iki islemdir. Ikisi ayni satirda yapilabilecegi gibi, farkli satirlarda da yapilabilir.



Variable Deger Atamasi (Assignment)

Declaration ve assignment asagidaki sekillerde yapılabilir.

1- Declaration ve assignment ayni satirda yapılabilir

```
int not=80 ;  
String isim="John Doe";  
boolean ogrenciMi=true;  
double notOrt=89.3;
```

2- Once declaration, sonra assignment yapılabilir

```
int not;  
  
not= 90;  
  
not= (not + 80)/2;
```

NOT : Declaration sadece 1 kere yapılır, assignment ise istendigi kadar yapılabilir.

Variable Deger Atamasi (Assignment)

3- Ayni data turundeki birden fazla variable ayni satirda deklare edilip, sonra tek tek assignment yapılabilir

```
int not1,not2,ortNot;  
  
not1= 80;  
not2= 90;  
ortNot= (not1 + not2)/2;
```

4- Ayni data turunde birden fazla variable tek declaration ile olusturulabilir.

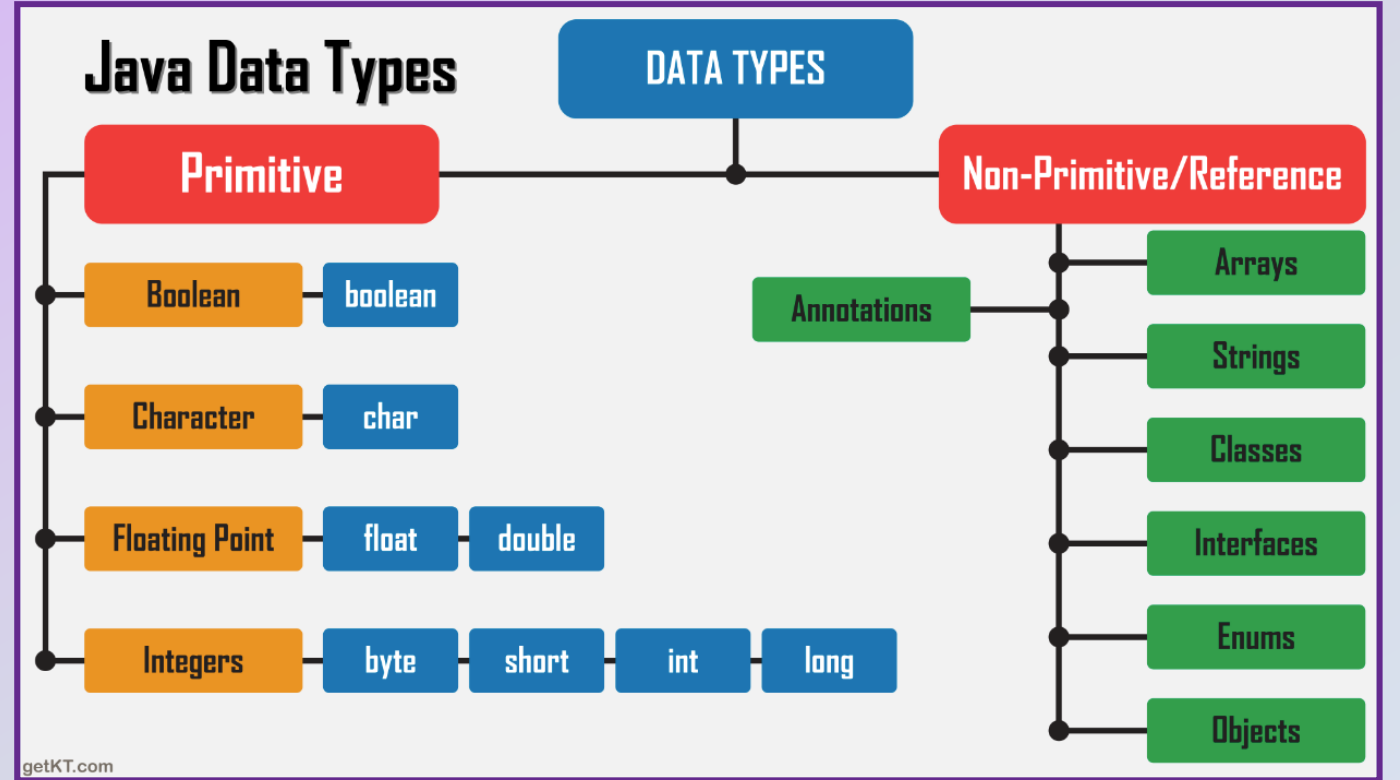
```
int not1=80,not2=90,ortNot= (not1 + not2)/2;
```

Java'da Data Turleri Nelerdir ?

Java'da temelde iki data turu kullanilir.

- 1- Primitive Data Turleri
- 2- Non-Primitive Data Turleri

Biz baslangicta primitive data turleri ve String kullanarak ilerleyecegiz, daha sonra diger non-primitive data turlerini ogrenecegiz.



Primitive Data Turleri

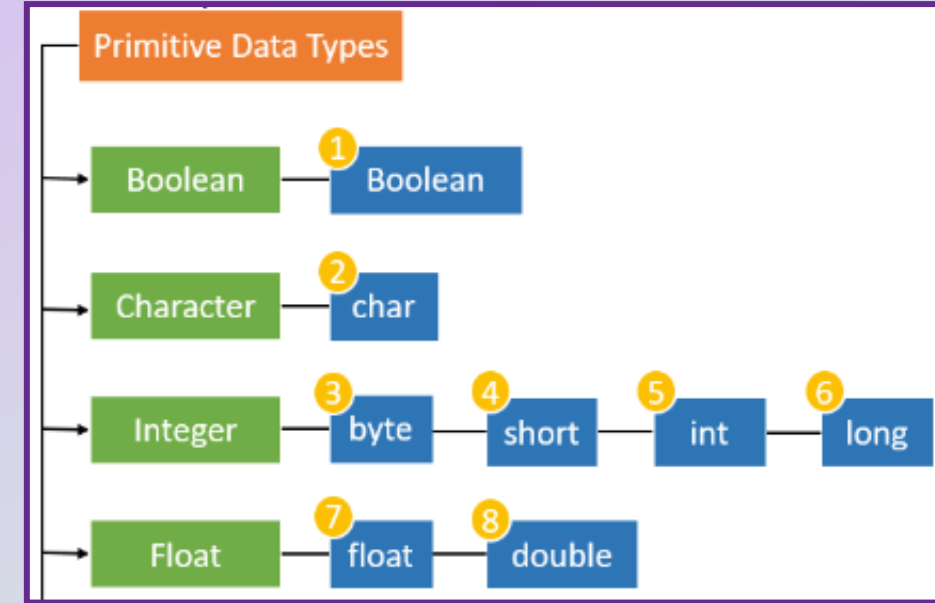
Java'da 8 primitive data turu kullanilir.

Primitive data turleri sadece deger store edebilirler ve hafizada kapladiklari alan her data turu icin sabittir.

1- Boolean : Mantiksal data sonuclarini store etmek icin kullanilir.

boolean data turundeki bir variable sadece 2 deger barindirabilir, **true** / **false**

Bilgisayar true icin 1, false icin 0 degerini tutar, dolayisiyla hafizada sadece **1 bit** yer kaplar



2- char : Tek bir karakter barindirir. Icerisinde harf, sayi veya ozel karakter olabilir.

char data turunun en belirgin farkliligi 'c' (tek tirnak) kullanmasidir. Bir data " kullanirsa char olmalidir.

```
char harf='A',sayi= '4',karakter='#';
```

hafizada **16 bit** yer kaplar

Primitive Data Turleri

Tam sayi barindiran primitive data turleri

Data Turu	Hafiza Boyut	minimum deger	maximim deger
3- byte	8 bit	$-2^7 = -128$	$2^7 - 1 = 127$
4- short	16 bit	$-2^{15} = -32.768$	$2^{15} - 1 = 32.767$
5- int	32 bit	$-2^{31} = -2.147.483.648$	$2^{31} - 1 = 2.147.483.647$
6- long	64 bit	$-2^{63} = -9.223.372.036.854.755.808$	$2^{63} - 1 = 9.223.372.036.854.755.807$

Bir variable icin hangi data turunu kullanacagimiz, uygulamamizin hafiza kullanimi icin onemlidir.

Ornegin, universitedeki ogrencilerin yaslarini barindiran bir variable icin byte yeterlidir.

Yaslari short, int veya long olarak da store edebiliriz ancak bu durumda kullanılacak hafiza miktarı katlanarak artacaktır.

Primitive Data Turleri

Ondalikli sayi barindiran primitive data turleri

Data Turu	Hafiza Boyut	min-max deger	Ondalikli basamak sayisi
7- float	32 bit	$\pm 3.40282347E+38F$	6-7 basamak
8- double	64 bit	$\pm 1.79769313486231570E+308$	15-16 basamak

Ondalikli sayilar icin hafiza durumu ve ondalik kismin uzunluguna gore data turu secilebilir.

Deger atamasi yaptigimizda Java'nin float ile double'i ayirt edebilmesi icin, float sayilarin yaninda **f** veya **F** kullanmamiz gerekir.

```
float a=20f;  
float b=6f;  
System.out.println(a/b); // 3.3333333
```

```
double c=20;  
double d=6;  
System.out.println(c/d); // 3.3333333333333335
```

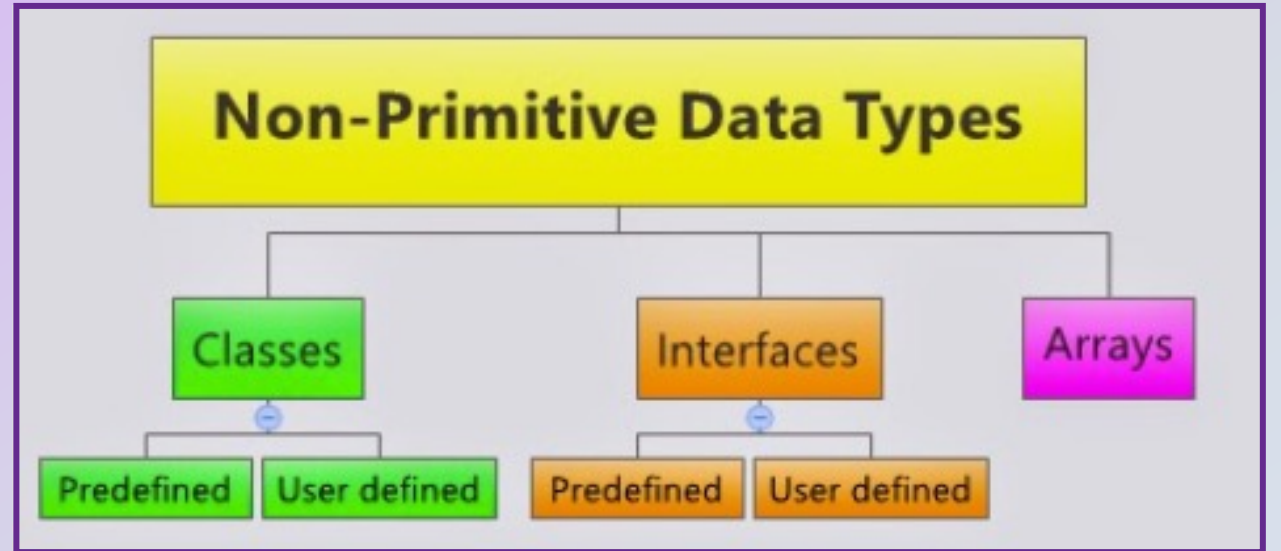
Non-Primitive Data Turleri

Primitive data turleri Java tarafından olusturulmustur ve biz yeni bir PRIMITIVE DATA TURU olusturamayiz.

Non-Primitive data turleri ise Java tarafından sinirlendirilmamislardir.

Java da bazi non-primitive data turleri olusturmustur, biz de yeni non-primitive data turleri olusturabiliriz.

Non-Primitive data turlerinin geneli icin **object** tabiri kullanılabilir, cunku tum non-primitive data turlerinin kendilerine ait bir class tarafından olusturulan objelerdir.



Class'lar OOP konsept cercevesinde bizim obje olusturdugumuz, kaliplar oldugundan, non-primitive data turleri bu class'lardan olusturulan objelerdir.

Class'lar method'lar da icerdiginden, non-primitive data turleri, olusturulduklari class'lardaki method'lari kullanabilirler.

Non-Primitive Data Turleri

Non-Primitive data turlerinden, simdilik String'i kullanacagiz. Ilerleyen derslerde Java'nin olusturdugu tum non-primitive data turlerini gorecegiz.

```
String isim= "John Doe";
```

String'i variable'lara deger atamak icin "" kullaniriz.

```
String isim= "John Doe";  
  
isim.  
  split(String regex)  
  toLowerCase()  
  substring(int beginIndex)  
  getBytes(StandardCharsets.UTF_8)  
  getBytes(String charsetName)  
  getBytes(Charset charset)  
  getBytes()  
  getBytes(int srcBegin, int srcEnd,  
  toLowerCase(Locale.ROOT)  
  toLowerCase(Locale locale)  
  toUpperCase(Locale.ROOT)  
  toUpperCase(Locale locale)  
  toUpperCase()  
  split(String regex, int limit)  
  substring(int beginIndex, int endIndex)  
  charAt(int index)
```

Non-primitive data turunde olusturulmus herhangi bir variable'in **ismini** yazip **.**'ya basarsaniz, o variable ile kullanabilecegimiz method'lari gorebilirsiniz.

Primitive vs Non-Primitive Data Turleri

İki data turu arasında 5 temel farklılık sayabiliriz.

Primitives	Non-Primitives
Tümü Java tarafından oluşturulmuştur.	Java tarafından oluşturulanlar olduğu gibi biz de oluşturabiliriz
Sadece değer içerirler, variable ile kullanılacak hazır method'ları yoktur.	İçerdikleri değerlerin yanında oluşturuldukları class'dan gelen hazır method'lar da barındırırlar.
Bir değer atanmadan oluşturulabilir ama kullanabilmek için mutlaka değer atanmalıdır.	Değer atanmadan null olarak işaretlenebilirler.
Data turu isimleri küçük harfle başlar (int, char vb)	Data turu isimleri büyük harfle başlar. (String vb..)
Primitive data türündeki variable'ların hafızada kapladıkları alan sabittir. Değeri küçük de olsa, büyük de olsa hafızada belirlenen miktarda alan ayrılır.	Hafızada kapladıkları alan sabit değildir. Data turu ve içerdigi datanın büyüklüğüne göre hafızada yer kaplarlar. (bir kelime veya binlerce kelime içeren String'lerin boyutları farklı olacaktır)

Variable'lar Icin Isim Verme Kurallari (Naming Convention)

Variable'lara isim verirken istedigimiz ismi secebiliriz ancak asagidaki kurallara uyulmasi gereklidir.

- 1- Variable isimleri buyuk-kucuk harf duyarlidir (**case sensitive**).
Not, NOT, not, nOT ... birbirinden farklıdır.
- 2- Variable isimlerinde **harf**, **rakam**, **_** ve **\$** kullanılabilir, boşluk veya * gibi özel karakterler kullanılamaz.
- 3- Variable isimleri harf ile başlamalıdır, rakam ile başlayamaz.
_ ve \$ ile başlayabilir ama kullanılması tavsiye edilmez.
- 4- Variable isimleri olarak keyword (Java'da tanımlı anahtar kelimeler) kullanılamaz.
for, int, short, class vb. variable ismi olamaz.
- 5- Variable isimleri küçük harfle başlar, birden fazla kelime içeriyorsa **camelCase** kullanılır, yani sonraki her kelimenin ilk harfi **buyuk harf**, diğer harfleri **kucuk harf** yapılır.

Hafıza Kullanımı

Java'da her method çalıştığında, o methoda ait bir stack ve ona ait bir stack memory alanı oluşturulur.

Ayrıca her method'un kullandığı heap memory vardır.

Stack memory'de primitive data türündeki variable'lar ve değerleri ile non-primitive data türündeki variable'ların referansları olurken, heap memory'de non-primitive variable'ların değerleri store edilir.

```
public static void main(String[] args) {  
    int a;  
    int b=20;  
    char c= '#';  
    String str1;  
    String str2="Java Guzeldir";  
    str1="Bence de Guzeldir";  
}
```

