

Primitive Datatype, Wrapperklasse

Ziel:

- Ich kann die **Primitiv Data Type** und **Wrapper-Klasse** unterscheiden und erklären.
- Ich kann **Wertzuweisung** und das **Zuweisen von Referenzen** unterscheiden und erklären.



Inhalte

Inhalte	1
statisch typisierte Programmiersprache	1
primitive data types von Java	2
Wrapper-Klassen	2

statisch typisierte Programmiersprache

Java ist eine **statisch typisierte Programmiersprache**. Das bedeutet, dass zu dem Zeitpunkt, wo das Java-Programm erstellt wird, bekannt sein muss, welchen Typ eine Variable hat.

Typen können so genannte **primitive data types** sein oder beliebig komplexe **Klassen**.

```
int alter = 25; // Type von alter ist int
```

oder beliebig komplexe Klassen

```
Person hans = new Person("Hans", "Muster", 25); //Type von hans ist Person
```

Das **Beispiel mit PHP** zeigt eine **schwache, dynamische Typisierung**:

```
$x = 23 + "2"  
// Zahl puls String, doch welcher Datentyp hat nun x?
```

primitive data types von Java

`byte, short, int, long, float, double, boolean, char`

- ⇒ Alle Primitive Datatypes sind klein geschrieben!
- ⇒ Beim Anlegen einer Variablen vom Typ eines *Primitive Data Type* wird unmittelbar Speicherplatz belegt.
Es braucht kein *new*.
Es wird kein Objekt instanziiert.
- ⇒ Bei Zuweisungen werden Werte kopiert.
- ⇒ Bei Vergleichen werden Werte verglichen

```
int alterHans = 25;
int alterMonika = alterHans; //Wert wird kopiert

if( alterHans == alterMonika) { //Wert wird verglichen
```

Die *primitive data types* verwendet man, wenn man einfache Werte verwalten will.

Wrapper-Klassen

Zu jedem Primitive Data Type gibt es auch eine Wrapper-Klasse.
Die Wrapper Klasse kapselt in sich einen Primitive Data Type.

`byte` → `java.lang.Byte`
`short` → `java.lang.Short`
`int` → `java.lang.Integer`
usw.

```
int alterVonHans = 25;    //Primitiv Data Type

//Wrapper Klasse, Objekt instanziiieren, Referenz zu weisen
Integer alterVomVaterVonHans = new Integer(60);
```

Mit den Wrapper-Klassen kann man die Vorteile von Klassen nutzen (Methoden), ohne ganz auf die Vorteile von *primitive data types* (Umgang mit Werten) verzichten zu müssen.

Wrapper Klassen sind für eine einfache Wertzuweisungen vorbereitet.

```
Integer alter = 34;
```

Doch man muss aufpassen.

Obwohl das obige aussieht wie eine Zuweisung eines Wertes, entsteht in Wirklichkeit ein Objekt.

Merke: Wrapperklassen sind etwas speziell in Ihrem Verhalten.