**Definition:**

Reserviert nur den platz damit man später da etwas großes reinpacken könnte.

int b; variable definiert

**Deklaration:**

Nimmt die Definition und reserviert den gesamten notwendigen Speicherbereich

int a = 42; variable deklariert bzw. initialisiert

# Vereinbarungen

*[dienen dazu, Namen für Bestandteile von Programme, z.B.: … einzufügen]*

* Methoden
* Variablen *[dient als Speicherbereich zum einfügung und auslesung von Werten]*
  + Ausdrücke *[die Werte die in der Variable zu finden sind]*
* Typen

# Anweisungen

*[z.B.: System.out.print]*   
da zu gehören:

* Schleifen
* Bedingungen
* Zuweisungen
* Anweisungsfolgen
* Leere Anwesungen
* Prozedur-, Funktions-, Methoden-Aufrufe
* …

Ausdruck:

4 + 3;

Ausdruck Anweisung:

a = 4 + 3;

# Token

* Keywords[Datentypen (?)] ( int , class, if, goto ) [pre-defined/reserved words with special meaning/functions]
* Identifiers ( int totalSum ) [are used to name a Variable, Funktion, Class, … “camelCase , PascalCase”]
* Literals ( int totalSum = 42 ) [to set a variable a specific value]
  + static
  + final

[Ein Literal ist ein Wert der konkret definiert würde ( String a = "3" ) die 3]

[( a + 4 ) die 4 ist ein Literal]

* Strings ( “yolo”; “hello” )
* Comments ( //idk what this do , /\* … \*/)
* Separators/Punctuators ( ; , . = )
  + Demiters ( (), [], {} )
  + White Spaces ( blank, line break )
* Operators/Symbols

| Operator | Symbols |
| --- | --- |
| Arithmetic | + - / \* % |
| Unary | + + - - ! |
| Assignment | = += -= \*= /= %= ^=  |= <<= >>= >>>= (?) |
| Relation | = = != < > <= >= |
| Logical | && || |
| Ternary | (Condition) ? (Statement1) : (Statement2) |
| Bitwise | & | ^ ~ |
| Shift | << >> >>> |
| Zuweisung | = |

<https://www.javatpoint.com/java-tokens>

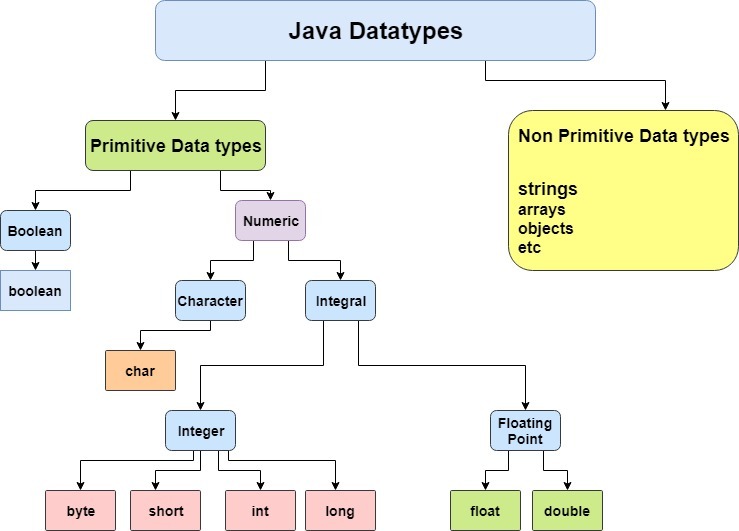
Python:

\*\* ⇒ for exponential

// ⇒ division as integer number

- end

**Primitive-type** [Elementare-Datentypen]



As the type of the Variable is seated (in the compile time), it can not be changed (statical type)

Sichtbarkeitsbereich (Scope) Local Scope

Ein Bereich, wo variable leben between { }

### 