

De Java programmeertaal

Variabelen && Types

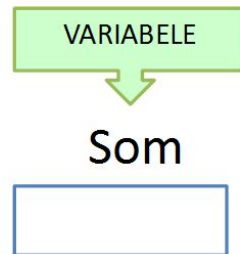
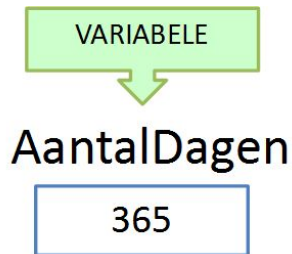
A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

Inhoud

- Wat is een variabele, een literal en een type?
- Soorten data types
 - Primitieve datatypes
 - Referentie datatype
- Uitleg scope + soorten
- Declaratie van variabelen
- Toekenning van waarden
 - Booleans
 - Karakters
 - Escape codes
 - Strings
 - Gehele getallen
 - Floating point
- Naamgeving
- Constanten
- Type conversie

Wat is een
variabele, een
literal en een type?

Variabelen en literals



LITERAL = Letterlijke waarde

Som = AantalDagen + 35

Soorten data types

Soorten Types in Java

- Het primitieve datatype:

Bevat 1 enkele waarde

- Referentie datatype:

Bevat een verwijzing naar een reeks variabelen of een object

Primitieve datatype

Gehele getallen

TYPE	OMSCHRIJVING	MIN	MAX
byte	Byte getalwaarde 8 bit	-128	127
short	Short integer 16 bit	-32768	32767
int	Integer 32 bit	-2^{31}	$2^{31} - 1$
long	Long integer 64 bit	-2^{63}	$2^{63} - 1$

Reële getallen

TYPE	OMSCHRIJVING	MIN	MAX
float	Single precision floating point	-3.4^E+38	3.4^E+38
double	Double precision floating point	-1.7^E+308	1.7^E+308

Andere types

TYPE	OMSCHRIJVING	MIN	MAX
boolean	Booleaanse variabele	Niet waar = false	Wel waar = true
char	Karakter		

Declaratie van variabelen

Verplicht voor het gebruik!

Type variabelenaam;

```
int leeftijd;  
int aantalKinderen;  
boolean gehuwd;
```

Type variabelenaam1, variabelenaam2, ;

```
int leeftijd, aantalKinderen;  
boolean gehuwd;
```

Toekenning van literal waarden

Type variabelenaam = startwaarde;

```
int leeftijd = 23;  
int aantalKinderen = 5;  
boolean gehuwd = false;
```

Type varnaam1 = waarde, varnaam2 = waarde, ;

```
int leeftijd = 23, aantalKinderen = 5;  
boolean gehuwd = false;
```

Boolean

Toekennen van booleaanse waarden

- hebben steeds de waarde **true** or **false**
- indien geen waarde meegegeven bij declaratie steeds **default false**

true en **false** zijn **gereserveerde** woorden in Java

Karakters

Escape codes

ESCAPE CODE	BETEKENIS
<code>\n</code>	Newline: nieuwe regel
<code>\r</code>	Return
<code>\t</code>	Tabulator
<code>\b</code>	Backspace: terugkeertoets
<code>\f</code>	Formfeed: nieuwe pagina
<code>\</code>	Enkelvoudig aanhalingsteken
<code>\"</code>	Dubbel aanhalingsteken
<code>\\</code>	Backslash: schuine streep

String literals

String tekst = "De bananen zijn krom";

→ Reeks karakters tussen dubbele
aanhalingstekens

Komt later nog uitgebreid aanbod in
onze cursus.

Gehele getallen

Literals voor gehele getallen

→ Worden standaard weergegeven in decimale notatie.

Andere notaties:

Talstelsel	Notatie	Grontal	Geldige cijfers
decimaal	34	10	0 1 2 3 4 5 6 7 8 9
octaal	O42	8	0 1 2 3 4 5 6 7
hexadecimaal	Ox22	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

Literals voor gehele getallen

- Worden standaard als integer (32 bit) beschouwd.
- Om een literal als **long** te beschouwen, zet men er een **L** achter:
 - ◆ `long aLong = 123L;`

SINDS JAVA 7: Underscore:

- `long aLong = 123_000_000L;`

Floating point literals

→ Worden beschouwd als getallen met drijvende komma:

- ◆ Met een decimaal punt:
 - 1.235
- ◆ Met wetenschappelijke notatie:
 - 45E+3
- ◆ Met de letter D (double):
 - 459D
- ◆ Met de letter F (float):
 - 5687F

Naamgeving

- Reeks Unicode-karakters
- Begint met een letter, een dollar-teken (\$) of underscore (_)
- Geen gereserveerd woord (Cursus p. 52)
- Moet uniek binnen zijn scope zijn.

Conventions in Java

Gewoonte (convention)

- **Variabelen:** eerst **kleine** letter, ieder bijkomend woord met hoofdletter. [**lower CamelCase**]
- **Klassen:** met **hoofdletter** + bijkomende woorden ook in hoofdletters [**Upper CamelCase**]

Constanten – Final variables

Eenmaal een waarde toegekend kan deze **niet meer worden veranderd**

→ Gewoonlijk in hoofdletters.

→ Scheiding meerdere woorden: underscore

Voorbeeld:

```
final int CONSTANTE_WAARDE = 7;
```


Type conversie

Casting

Automatisch toegepast zolang geen verlies van gegevens.

Enkele vb:

```
int integer = 5;  
long aLong;  
aLong = integer;
```



Typeconversie

Is dit correct?

```
int anInteger;
```

```
long aLong = 6;
```

```
anInteger = aLong;
```

Typeconversie

Is dit correct?

```
int anInteger;
```

```
long aLong = 6;
```

```
anInteger = aLong;
```

FOUT !

Typeconversie

```
int anInteger;  
long aLong = 6;  
anInteger = aLong;
```

FOUT !

```
int anInteger;  
long aLong = 6;  
anInteger = (int) aLong;
```

OK
(OP EIGEN RISICO)

Typeconversie

- Opmerking:
 - **char** = **tekenloos** 16-bits getal
 - **short** = 16-bits getal **MET teken**

daaruit volgt:

char kan niet gecast worden naar een short,
wordt indien nodig wel gecast naar een int !

Typeconversie

IMPLICIT CAST:

```
long l = 10;  
short s = 5;  
int i = s;
```

EXPLICIT CAST:

```
float f = (float) 10.2;  
int i = 100;  
short s = (short) i;
```