

# Operatori aritmetici, relazionali, di assegnazione

---

- Di assegnazione: = += -= \*= /= &= |= ^=
- Di assegnazione/incremento: ++ -- %=
- Operatori Aritmetici: + - \* / %

Operatore	Significato
+	addizione
-	sottrazione
*	motiplicazione
/	divisione
%	resto
++var	preincremento
--var	predecremento
var++	postincremento
var--	postdecremento

Operatore	Significato
=	addizione
+=	addizione assegnazione
-=	sottrazione assegnazione
*=	motiplicazione assegnazione
/=	divisione assegnazione
%=	resto assegnazione

- Relazionali: == != > < >= <=

Operatore	Significato
<	minore di
<=	minore di o uguale a
>	maggiore di
>=	maggiore di o uguale a
==	uguale a

Operatore	Significato
!=	non uguale / diverso

---

## Operatori per Booleani

- Bitwise (interi): & | ^ << >> ~

Operatore	Significato
&&	short circuit AND
	short circuit OR
!	NOT
^	exclusive OR

### Attenzione:

- Gli operatori logici agiscono solo su booleani
    - Un intero NON viene considerato un booleano
    - Gli operatori relazionali forniscono valori booleani
- 

## Operatori su reference

Per i puntatori/reference, sono definiti:

- Gli operatori relazionali == e !=
    - N.B. test sul puntatore NON sull'oggetto
  - Le assegnazioni
  - L'operatore "punto"
  - NON è prevista l'aritmetica dei puntatori
- 

## Operatori matematici

Operazioni matematiche complesse sono permesse dalla **classe Math** (package java.lang)

- `Math.sin (x)` calcola  $\sin(x)$
- `Math.sqrt (x)` calcola  $x^{(1/2)}$
- `Math.PI` ritorna pi
- `Math.abs (x)` calcola  $|x|$
- `Math.exp (x)` calcola  $e^x$
- `Math.pow (x, y)` calcola  $x^y$

### Esempio

- `z = Math.sin (x) - Math.PI / Math.sqrt(y)`
-

## Caratteri speciali

Literal	Represents
<code>\n</code>	New line
<code>\t</code>	Horizontal tab
<code>\b</code>	Backspace
<code>\r</code>	Carriage return
<code>\f</code>	Form feed
<code>\\</code>	Backslash
<code>\"</code>	Double quote
<code>\ddd</code>	Octal character
<code>\xdd</code>	Hexadecimal character
<code>\udddd</code>	Unicode character