

# Aritmetik

# Aritmetik

Vi kan öka en variabel `fruits` med 1 genom att skriva:

```
fruits = fruits +1;
```

```
fruits += 1;
```

```
fruits ++;
```

VÄLJ DIN FAVORIT!

# Aritmetik

På motsvarande sätt kan vi minska variabeln fruits med 1 genom att skriva:

```
fruits = fruits -1;
```

```
fruits -= 1;
```

```
fruits --;
```

# Aritmetik

Addition (+), subtraktion(-),multiplikation(\*) och division (/) fungerar som i matematiken.

Prioriteringsreglerna gäller!

# Aritmetik

Ex:  $a+b/2$

$(a+b)/2$

$4-(3+2)$

# Aritmetik

MEN: Kom i håg att typerna inte förändras!

```
int x = 14;
```

```
int y = 3;
```

```
System.out.println(x/y);
```

Vad skrivs ut?

# Aritmetik

```
int x = 14;
```

```
int y = 3;
```

```
System.out.println(x%y);
```

Vad skrivs ut?

# Aritmetik

Det som kan vara nytt för många är **modulus** operatorn (%).  
Den ger oss resten vid en division:

$$\begin{aligned} 14/3 &= \\ (4+4+4) + 2 &= \\ 12 + 2 \end{aligned}$$

Skriver vi  $14/3$  så får vi svaret 12  
Skriver vi  $14/3$  får vi svaret 2

Detta kan användas för att exempelvis se om ett tal är udda eller jämt, eller för att plocka ut en viss siffra ur ett tal.



# Aritmetik

Heltalsdivision och rest ger oss dessa verktyg! ( $n$  är ett heltal)

$n/10$  :  $n$  utan sista siffran

$n\%10$ : sista siffran i  $n$ .

$n/100$  : ?

$n\%100$ : ?

$(n\%100)/10$ : ?

# Aritmetik

För mer avancerad matematik kan vi använda den inbyggda Math-klassen!

Method	Returns
<code>Math.sqrt(x)</code>	Square root of $x$ ( $\geq 0$ )
<code>Math.pow(x, y)</code>	$x^y$ ( $x > 0$ , or $x = 0$ and $y > 0$ , or $x < 0$ and $y$ is an integer)
<code>Math.sin(x)</code>	Sine of $x$ ( $x$ in radians)
<code>Math.cos(x)</code>	Cosine of $x$
<code>Math.tan(x)</code>	Tangent of $x$
<code>Math.toRadians(x)</code>	Convert $x$ degrees to radians (i.e., returns $x \cdot \pi/180$ )
<code>Math.toDegrees(x)</code>	Convert $x$ radians to degrees (i.e., returns $x \cdot 180/\pi$ )
<code>Math.exp(x)</code>	$e^x$
<code>Math.log(x)</code>	Natural log ( $\ln(x)$ , $x > 0$ )

# Aritmetik

Vi kan göra om till exempel **double** till **int** genom att "casta" men datorn kommer "klippa av" decimalerna när den gör detta så de avrundningsregler vi är vana vid gäller inte.

`Math.round()` - ger oss närmaste heltal.

Detta kan vi använda om vi vill undvika avrundningsfel när vi **castar** något från decimaltal till heltal

# Aritmetik

```
double x = 2.73456;  
int newX = (int)(x);
```

```
double roundedX = Math.round(x);  
int newRoundedX = (int)(x);
```