

Primitive Data Types

type	size	range of values
byte	8-bit <i>signed</i> 2's comp	(-128 -> 127)
short	16-bit <i>signed</i> 2's comp	(-32,768 -> 32,767)
int	32-bit <i>signed</i> 2's comp	(-2 ³¹ -> 2 ³¹ -1)
long	64-bit <i>signed</i> 2's comp	(-2 ⁶³ -> 2 ⁶³ -1)
long	64-bit <i>unsigned</i>	(0 -> 2 ⁶⁴ -1)
float	single-precision 32-bit <i>signed</i>	(-3.40282347 x 10 ³⁸ -> 3.40282347 x 10 ³⁸)
double	double-precision 64-bit <i>signed</i>	(-1.79769313486231570 x 10 ³⁰⁸ -> 1.79769313486231570 x 10 ³⁰⁸)
char	16-bit <i>unsigned</i> Unicode character	(0 -> 65,535)
boolean	<i>size not defined</i>	true / false

Hello World

```
{{noshy}}public class HelloWorld
{
    public static void
main(String[] args)
    {
        System.out.println("Hello
World!");
    }
}
```

Declaring and Initializing

Operations on Number Variables

Integer	sign		
		+	+99 -or- -99
	<i>add</i>	+	5 + 3 = 8
	<i>subtract</i>	-	5 - 3 = 2
	<i>multiply</i>	*	5*3 = 15
	<i>divide</i>	/	5/3 = 1 <i>no fractional part</i>
	<i>remainder</i>	%	5 % 3 = 2
Floating-Point Numbers	<i>add</i>	+	3.141 + 2.0 = 5.141
	<i>subtract</i>	-	3.141 - 2.0 = 1.111
	<i>multiply</i>	*	3.141 * 2.0 = 6.282
	<i>divide</i>	/	3.141 / 2.0 = 1.5705

Boolean Operations

Values	Literals	Operations	Operators
<i>true</i>	true	and	&&
<i>false</i>	false	or	
		not	!
a	b	a && b	a b
false (0)	false (0)	false	false
false (0)	true (1)	false	true
true (1)	false (0)	false	true
true (1)	true (1)	true	true

Comparison Operators

Printing and Parsing

Printing to console

```
System.out.println("String s");  
System.out.println("String s");  
print s followed by newline  
System.out.println(); print a newlin
```

Parse command-line args

```
int Integer.parseInt("String s")  
convert s to an int value  
double Double.parseDouble("String s");  
convert s to a double value  
long Long.parseLong("String s");  
convert s to a long value
```

Integers

```
int a, b; <-- Declare two integer variables
```

```
a = 100; <-- Initialize 'a' with a value of 100
```

```
b = 18; <-- Initialize 'b' with a value of 18
```

```
int c = a + b; <-- Declare and initialize c with the value of a plus b
```

Double

```
double a, b;
```

```
a = 1.57;
```

```
b = 9.8765;
```

```
double c = a + b;
```

It is the same for every **primitive** data type.

Operator	Meaning	true	false
==	equal	2 == 2	2 == 3
!=	not equal	3 != 2	2 != 2
<	less than	2 < 13	2 < 2
<=	less than or equal	2 <= 2	3 <= 2
>	greater than	13 > 2	2 > 13
>=	greater than or equal	3 >= 2	2 >= 3

Examples:

Check if a number is a multiple of 2

`(x % 2 == 0)` returns true if x is a multiple of 2

Check months

`(month >= 1) && (month <= 12)`
returns true if month is between 1 and 12



By **Kelvin** (EthanHann)

cheatography.com/ethanhann/

Not published yet.

Last updated 19th February, 2017.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>