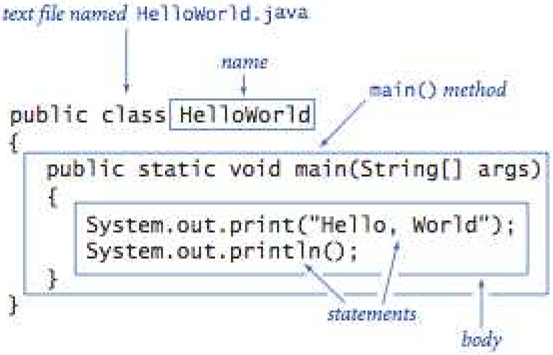
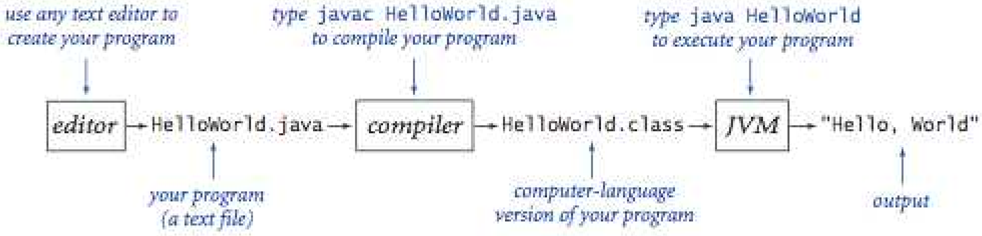
## Hello, World.



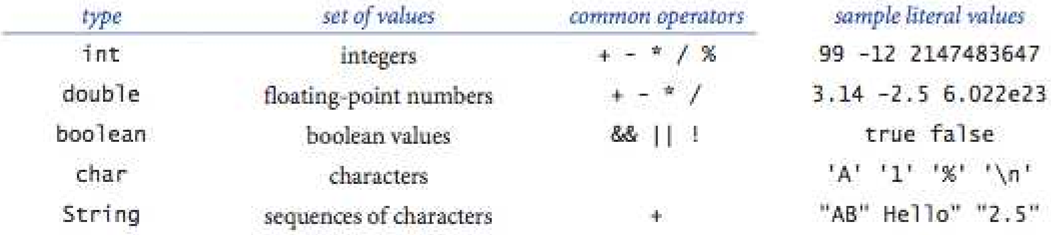
```
public class HelloWorld
{
   public static void main(String[] args)
   {
      System.out.print("Hello, World");
      System.out.println();
   }
}
```

## Editing, compiling, and executing.



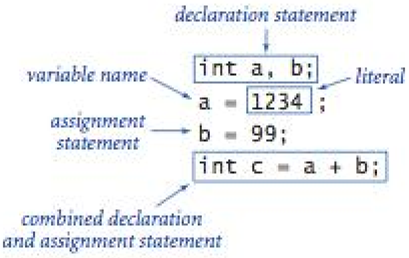
use any text editor to create your program
type javac HelloWorld.java to compile your program
type java HelloWorld to execute your program
editor
HelloWorld.java
compiler
HelloWorld.class
JVM
"Hello, World"
your program (a text file)
computer-language version of your program
output

## Built-in data types.



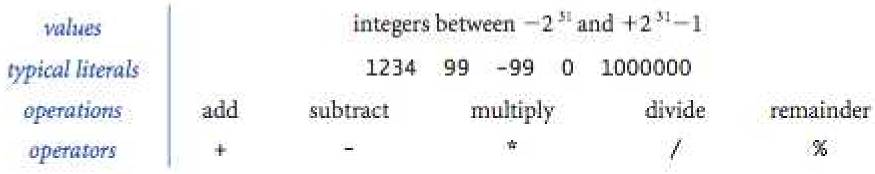
| *type* | *set of values* | *common operators* | *sample literal values* |
|---|---|---|---|
| int | integers | + - * / % | 99 -12 2147483647 |
| double | floating-point numbers | + - * / | 3.14 -2.5 6.022e23 |
| boolean | boolean values | && \|\| ! | true false |
| char | characters | | 'A' '1' '%' '\n' |
| String | sequences of characters | + | "AB" Hello" "2.5" |

## Declaration and assignment statements.

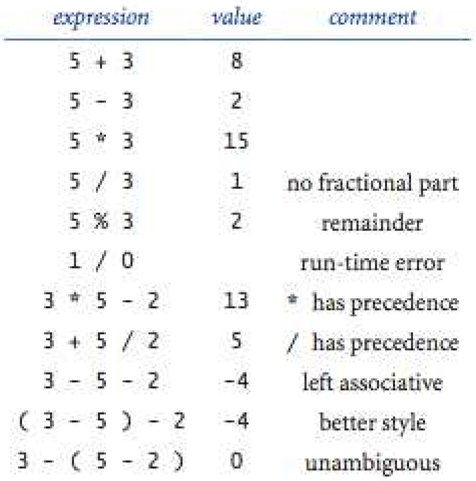


```
int a, b;
a = 1234;
b = 99;
int c = a + b;
```

Integers.

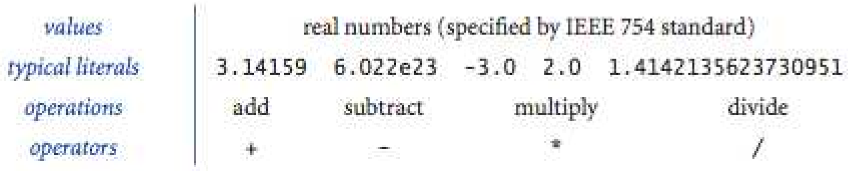


| | | | | | |
|---|---|---|---|---|---|
| *values* | integers between $-2^{31}$ and $+2^{31}-1$ | | | | |
| *typical literals* | 1234 99 -99 0 1000000 | | | | |
| *operations* | add | subtract | multiply | divide | remainder |
| *operators* | + | - | * | / | % |

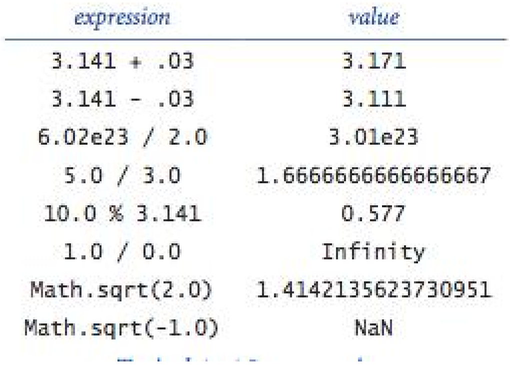


| *expression* | *value* | *comment* |
|---|---|---|
| 5 + 3 | 8 | |
| 5 - 3 | 2 | |
| 5 * 3 | 15 | |
| 5 / 3 | 1 | no fractional part |
| 5 % 3 | 2 | remainder |
| 1 / 0 | | run-time error |
| 3 * 5 - 2 | 13 | * has precedence |
| 3 + 5 / 2 | 5 | / has precedence |
| 3 - 5 - 2 | -4 | left associative |
| ( 3 - 5 ) - 2 | -4 | better style |
| 3 - ( 5 - 2 ) | 0 | unambiguous |

Floating-point numbers.

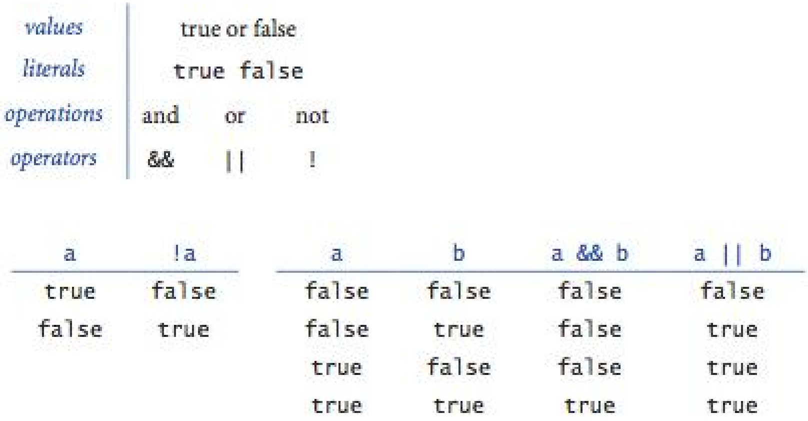


| | | | | |
|---|---|---|---|---|
| *values* | real numbers (specified by IEEE 754 standard) | | | |
| *typical literals* | 3.14159 6.022e23 -3.0 2.0 1.4142135623730951 | | | |
| *operations* | add | subtract | multiply | divide |
| *operators* | + | - | * | / |



| *expression* | *value* |
|---|---|
| 3.141 + .03 | 3.171 |
| 3.141 - .03 | 3.111 |
| 6.02e23 / 2.0 | 3.01e23 |
| 5.0 / 3.0 | 1.6666666666666667 |
| 10.0 % 3.141 | 0.577 |
| 1.0 / 0.0 | Infinity |
| Math.sqrt(2.0) | 1.4142135623730951 |
| Math.sqrt(-1.0) | NaN |

## Booleans.

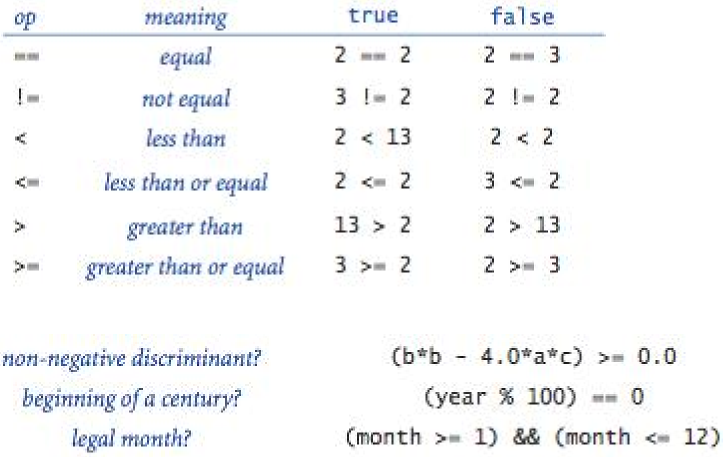


| | |
|---|---|
| *values* | true or false |
| *literals* | `true` `false` |
| *operations* | and or not |
| *operators* | && \|\| ! |

| a | !a |
|---|---|
| true | false |
| false | true |

| a | b | a && b | a \|\| b |
|---|---|---|---|
| false | false | false | false |
| false | true | false | true |
| true | false | false | true |
| true | true | true | true |

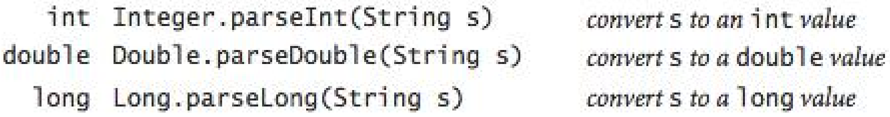
## Comparison operators.



| *op* | *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 |

| | |
|---|---|
| *non-negative discriminant?* | (b*b - 4.0*a*c) >= 0.0 |
| *beginning of a century?* | (year % 100) == 0 |
| *legal month?* | (month >= 1) && (month <= 12) |

## Parsing command-line arguments.



| | | |
|---|---|---|
| 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* |

## 연산자의 종류

| 이름 | | 연산자 | 결합규칙 |
|---|---|---|---|
| 단항연산자 | 후치 증감 | ＋＋, －－ | 좌 → |
| | 전치 증감 | ＋＋, －－ | 우 ← |
| | 단항 플러스/마이너스 (양수/음수) | ＋, － | 우 ← |
| 곱하기, 나누기, 나머지 | | *, /, % | 좌 → |
| 더하기, 빼기 | | ＋, － | 좌 → |
| 문자열 연결 | | ＋ | 좌 → |
| 대입 연산자 | | =, *=, /=, %=, +=, −= | 우 ← |
| 관계 연산자 | | >, >=, <, <=, ==, != | N/A |
| 논리 연산자 | | &, &&, \|, \|\|, ! | |

## 후치 증감 vs. 전치 증감

▶전치

```
x = 10;
y = ++x + 5;
System.out.println(“x = ”+x);
System.out.println(“y = ”+y);
```

▶출력

```
x = 11
y = 16
```

▶후치

```
x = 10;
y = x++ + 5;
System.out.println(“x = ”+x);
System.out.println(“y = ”+y);
```

▶출력

```
x = 11
y = 15
```