

Page No.:

Date:

Page No.:

Date:

SOURCE
Print "Hi"

COMPILER
(based on Platform)

byteCode
class

PROCESSOR

SOURCE
Print "Hi"

COMPILER
(common)

byteCode
class

PROCESSOR

JVM (Interpreter)
based on Platform

JIT (Just in time compiler)

JIT JVM works on runtime. It converts byte code into native code. JVM line by line execute code. so JIT used to speed up.

JVM

Libraries,
Files

JRE

Java
tools

Java virtual machine

Java runtime Environment

JDK

Java development kit
includes tools, compiler

IDE (Integrated Development Environment)

class User { class

String userName;

String pwd;

register() { }

login() { } method

}

packages (contains classes)

java.lang/java.util/java.io inbuilt

we can create our own package

IDE + JRE

eclipse > File > Java Project > Enter Name > Finish > Don't (create)

expand > Src > New > Class > Finish > Write Code > Run

[C]

IDE + JRE

[C]

JDK

- notepad enough

JDK

File Explorer > C: > java > click > New > text Document > write code
 > Save As - name.java > CMD > cd C:\java > C:\java> java C:\name.java
 new file will be created (if not) > java > java First
 Re. java Re. run code to run the First class

QED java not recognised as command?

Program Files > java > JDK > bin < copy link

Search Bar > Edit the System environment variables > System Variables >
 path > Edit > New > Paste that link > ok > ok

Why this error? because cmd prompt not java path given
 address so in this way we define path like a class static method.

import java.io.*; ^{→ commonly class contains import java.io}
 it is module used for i/o related task
 we don't need to import standard packages
 like System.out.print("Hello");

Filename
 1. int * public class First {
 public static void main(String args[]) {
 System.out.print("print this");
 }
 }

System.out.println("first line")

System.out.println("second line")

datatype variable
 int ticketPrice; — variable declaration

ticketPrice = 800; — value assigning

System.out.println(ticketPrice);

int ticket = 10; — initialisation

int amount, total, avg;

DATA TYPES

datatype	Bytes	Desc
byte	1	-128 to 127
short	2	-32768 to 32767
int	4	-2 ³¹ to 2 ³¹ -1
long	8	-2 ⁶³ to 2 ⁶³ -1
float	4	upto 7 decimal digits
double	8	upto 15 decimal digits
boolean	1	is a false
char	2	single character

ASCII Table

Character Encoding Binary data, convert characters to messages
 store information ASCII standard

Case Sensitive — System.out... ≠ system.out

Camel case — ticketPrice (space suggest capital & lower case)

int price = 900;

boolean ispaid = true;

char c = 't';

Same rules for Variable name: \$, -, num at mid, no space

// command single line

/* multiline command */

Literals

int amount = 900; — numerical literals

boolean soldOut = false; — boolean literals

char ch = 'a'; '*' 'A' — character literals

"hello world" — string literals

JDK

File Explorer > C: > java > click > New > text Document > write code
 > Save As - name.java > CMD > cd C:\java > C:\java> java -c name.java
 new file will be created (bytecode) > java First
 re: JRE re: no need to mention First.class

ERO javac not recognised as command?

program Files > java > JDK > bin <- copy link

Search Env > Edit the System environment Variables > System Variables >
 path > Edit > New > Paste that link > ok > ok

Why this error? because cmd prompt ~~is~~ javac path original
 program so in this way we define path like express static method.

import java.io.*; ^{to know class compile import Objavio.}
 io is module used for i/o related task
 we don't need to import standard packages
 like System.out.println("Hello");

import * public class First { ^{Filename}
 public static void main(String args[]) {
 System.out.print("print this");
 }
 }

System.out.println("first line")
 System.out.println("second line")

datatype variable
 int ticketPrice; — variable declaration
 ticketPrice = 800; — value assigning
 System.out.println(ticketPrice);
 int ticket = 10; — initialisation
 int amount, total, avg;

DATA TYPES

datatype	Bytes	Desc
byte	1	-128 to 127
short	2	-32768 to 32767
int	4	-2 ³¹ to 2 ³¹ -1
long	8	-2 ⁶³ to 2 ⁶³ -1
float	4	upto 7 decimal digits
double	8	upto 15 decimal digits
boolean	1	true or false
char	2	single character

ASCII Table

Character Binary ASCII Convert decimal to binary
 store decimal ASCII standard value to binary

Case Sensitive — System.out... ≠ system.out
 Camel case — ticketPrice (space assign capital & lowercase)

int price = 900;
 boolean ispaid = true;
 char c = 't';

Some rules for variable name: \$, -, num at end, no space...
 // command single line
 /* multiline command */

Literals

int amount = 900; — numerical literals
 boolean soldOut = false; — boolean literals
 char ch = 'o'; 'x' 'A' — character literals
 "hello world" — string literals

escape sequences

```
System.out.println("Hello\nWorld"); // Hello
System.out.println("\nHello world\n"); // "Hello world"
```

```
int amount = 010; // Octal number // 8
          0x8; // Hexa decimal // 10
```

```
long lng = 71; or 71L; // (consider long)
int n = 0b0101101; // JDK 7.0 (store binary)
int n2 = 123_456_789; // (improve readability)
```

```
float pi = 3.14159f; // (standard notation)
float pi2 = 3.14159e-05; // (scientific notation) (e, -)
```

Type Casting

```
byte b = 5; // 1 byte
```

```
int n; // 4 byte
```

```
n = b; // automatic type conversion or implicit
```

even if it's different type it works BUT!
 Retrieve using variable 20000 Byte given - error

```
byte b; // 1 byte
```

```
int n = 10; // 4 byte
```

```
b = n;
```

ERROR
 cannot convert int to byte

So WHAT TO DO?

```
b = (byte)n; // explicit type conversion // 10
```

```
byte b;
```

```
int n = 100000;
```

```
b = (byte)n;
```

unreasonable 9p
 7 < -96

byte 20000 byte 100000

store original (garbage)

```
import java.util.Scanner;
```

```
public class inputdemo {
    public static void main (String args[]) {
        Scanner scanner = new Scanner (System.in);
        System.out.println("what is your name?");
        String name = scanner.nextLine(); // for String
        System.out.println("what is your rating from 1 to 5");
        short rating = scanner.nextInt(); // for Integer
        System.out.println("hello " + name);
        System.out.println("you rated us " + rating); } }
```

```
System.out.println("what is your rating from 1 to 5");
```

```
Short rating = scanner.nextShort();
```

```
scanner.nextLine(); // the number scan always write this to avoid empty value
```

```
System.out.println("Enter email"); // email or enter key empty
```

```
String email = scanner.nextLine(); // email or enter key empty
```

Operators

1) Arithmetic Operators (+, -, *, /, %)

```
int a = 10, b = 2, c; // float a = 10, b = 2, c;
c = a/b; // c = a/b // 5.5
```

2) Compound Assignment

```
a = a + 3;
```

```
a += 3; // a = 3; // a += 3;
```

3) increment Decrement

```
a++; // (post increment) rules:
```

```
++a; // (pre increment) same as javascript
```

```
a--; // (post decrement)
```

```
--a; // (pre decrement)
```


iv) Relational Operator (<, >, <=, >=, !=)

even character have it check it out

v) Bitwise Operator

& AND << Left shift
| OR >> Right shift
~ NOT >>> Right Shift Zero fill
^ EXOR

int a=9, b=12;

a & b < 8
a | b < 13
~ a < -10

byte a=4;

System.out.println(a << 28); //promoted to int < 1073741824

System.out.println((byte)(a << 28)); < 0
result byte is 0 because casting million type cast

int a=-5;

a >> 1 1110 0111 -3
a >>> 1 0011 0011 2147483645
these are sign bit +, -

a = a & 1 a & 1

a = a << 1 a << 1

System.out.println("Right shift " + (a >> 1));

System.out.println((a >> 20) & (a << 30));