Java tutorial in hindi

Java Notes For Quick Reference!

1. // [comment]
Single line comment.

2. /* [comment] */
Multi line comment.

3. public
This can be imported publically.

4. import [object].*
Imports everything in object.

5. static
Going to be shared by every [object].

6. final
Cannot be changed; common to be defined with all uppercase.

7. double
Integer with numbers that can have decimals.

8. ;
Put after every command.

9. String
Just a string of characters.

10. Private
Can only be changed by a method.

11. int
Can store numbers from 2^-31 to 2^31.

12. fields are attributes

13. boolean
Can have true or false as the value.

14. { }
These are used to start and end a function, class, etc.

15. byte

Course Content
Show Player

☐ 1. Learn JavaScript In One Video In Hindi (2018)
Free YouTube Video

☐ 2. Learn Python In Hindi In One Video - हिंदी में (Latest Tutorial)
Free YouTube Video

☐ 3. Learn Php In One Video In Hindi - हिंदी में (Latest PHP Tutorial 2018)
Free YouTube Video

☐ 4. 15 Minute Python Tutorial For Beginners In Hindi (Full & Complete Crash Course)
Free YouTube Video

☐ 5. Learn Bootstrap In Hindi In One Video - हिंदी में (Latest Tutorial 2019)
Free YouTube Video

☐ 6. Learn HTML In One Video
Free YouTube Video

☐ 7. Learn jQuery In Hindi In One Video - हिंदी में (Latest Tutorial 2018)
Free YouTube Video

☐ 8. Create A Responsive Website Using HTML, CSS And Bootstrap 4 In Hindi
Free YouTube Video

20. protected
Can only be accessed by other code in the package.

21. Scanner
This lets you get user input.

22. new [object constructor]
This will let you create a new object.

23. System.in
This lets you get data from the keyboard.

24. public [class]()
This will be the constructor, you use it to create new objects.

25. super()
This will create the superclass (the class it's inheriting).

26. extends [class]
Makes the object a subclass of [object], [object] must be a superclass.

27. ++
Will increment the amount.

28. --
Will decrement the amount.

29. += [amount]
Increment by [amount]

30. -= [amount]
Decrement by [amount]

31. *= [amount]
Multiply by [amount]

32. /= [amount]
Divide by [amount]

33. System.out.println([text])
Will print something to the output console.

34. +
Can be used for concatenation. (ex. "6" + [var_here])

35. public static void main(String[] args)
This is your main function and your project will start in here.

36. System.out.print([text])

41. !
This means not.

42. ||
This means or.

43. ==
This means equal to.

44. <
This means less than.

45. >
This means greater than.

46. >=
This means greater than or equal to.

47. [inputVarHere].hasNextLine()
This will return if there is a next line in the input.

48. this
Refer to the class that you are in.

49. [caller].next[datatype]()
This will get the [datatype] that you somehow inputted.

50. Create getters and setters
This will create the get methods and set methods for every checked variable.

51. [caller].hasNext[datatype]()
This will return if it has the correct datatype within the input.

52. overloading
If you have different parameters you can call them whatever way you want.

53. parameters
These are the inputs of your function.

54. ([datatype])[variable]
This will convert [variable] into [datatype]. Also known as casting.

55. Math.random()
Generate an extremely percise string of numbers between 0 and 1.

56. Primitives
Just the basic data types which are not objects.

57. [x].toString()

62. switch([variable])
This will do stuff with specific cases. (e.g. switch(hi){ case 2: (do stuff)})

63. case [value]:
This will do stuff if the case is the case.

64. break
Put that when you want to leave the loop/switch; should be at end of case.

65. default [value]:
This will do stuff if none of the cases in the switch statement was made.

66. for ([number]; [condition]; [operation])
This will start at [number] and then do [operation] until [condition] is met.

67. continue
This will just go back to the enclosing loop before reaching other code.

68. while ([condition])
This will basically do something while [condition] is true.

69. void
This means no return type.

70. return
This will return something when you call it to where it was called from .

71. do { } while ([condition])
Guarantees it will execute once even if [condition] isn't met.

72. printf("%[type] stuff here bah bla", [variable here])
This will let you use [variable here] with %s being where.

73. System.out.printf([text])
Another way to print? // didn't quite get but ok then

74. [type] [returntype] [name]([parameters]) {
This is a way to create a method.

75. [type][[indexes]]
This will create an array with [indexes] amount of indexes; default infinite.

76. int[] something = new int[20];
This will just make an array of ints with 20 ints in it.

77. for ([object] [nameOfObject] : [arrayOfObject]) {
This will iterate through all of the arrayOfObject with object in use incrementing by 1 until done.

78. [object][[1]][[2]][[3]] [name] = {[value] [value] [value] \n [value] [value] [value]}

```java
package com.codewithharry;

import java.util.Scanner;

public class Main {

    static int sum(int a, int b){
        return a+b;
    }

    public static void main(String[] args) {
     // Write your code here
//        System.out.println("Hello World");
     /* Variables
     Just Like:
        -Water - Bucket
        -Masala - box
        -Lunch - LunchBox
     In Java:
     Variables are containers which store data values
     String, int, float, char, boolean
     How to declare variables:
     syntax - <dataType> <variableName> = <value>;
      */
     String name = "Harry";
```

← Previous

Next →