Java Keywords Lists and Definitions PDF

tutorialsfield.com

There are 50 keywords in the Java Programming Language but only 48 are used. Two keywords **goto** and **const** are not used in the Java programming language. Now Let us see the list of all the keywords in Java with their definition one by one.

Note – All the keywords in Java Programming Language are in lower case characters .

Java Keywords Lists and Definitions

I have divided all these 50 keywords in Java Programming language according to their functionalities in different categories, so it will be easier for you to understand and learn all these Java Keywords.

#1 Data Type Keywords

Keywords which are used to declare variables of different data types.

1. byte : byte is a keyword used to store numbers ranging from **-128 to 127**.

```
byte num=13;
```

2. short : short keyword is used to store numbers ranging from **-32768 to 32767**.

```
short num= -3245;
```

3. int : int keyword is used to store numbers ranging from **-2147483648 to 2147483647**.

```
int num= 65783920;
```

4. long : long keyword is used to store numbrs ranging from **-9223372036854775808L to 9223372036854775807L**. A letter **'l'** or **'L'** should be added to tell the compiler that the numbers is a long and not an int.

```
long num= 6578392546544642L;
```

5. char : char keyword is used to store single character/letters or ASCII values. Its value range is from **'\u0000' to '\uffff'** or **0 to 65535**. All the characters must be enclosed within single quotes.

```
char var='A';
char name='\u0041';
```

6. boolean : boolean keyword is used to store values true or false.

```
boolean var=true;
```

7. float : float keyword is used to store fractional values ranging from **-3.4028235E38F to 3.4028235E38F**. A letter **'f'** or **'F'** should be added to tell the compiler that the fraction is a float and not a double.

```
float var=-3.40282E38F;
```

8. double : double keyword is used to store fractional values ranging from **-1.7976931348623157 x 10^{308} to 1.7976931348623157 x 10^{308}** .

```
double var = -1.234567543;
```

#2 Looping or Iterative statement Keywords

Keywords which are used for loops and Iterative statements.

9. for : for keyword is used to start the for loop. It is an entry-controlled loop in which the condition is checked first, and then the loop body gets executed only if the condition is satisfied. If the number of iterations is known already, then it is recommended to use for loop.

```
for(int i=0;i<=5;i++)
{
System.out.println(i);
}
```

10. while : while keyword is used to start the while loop. It is also an entry-controlled loop. If the number of iterations is not known already, then it is recommended to use while loop.

```
int i=0;
while(i<=5)
{
System.out.println(i);
i++;
}
```

11. do : do keyword is used to start a do-while loop. It is used along with the while to create a do-while loop. do-while loop is an exit-controlled loop in which, first of all the loop body gets executed and then the condition is checked while exiting from the loop body.

```java
int i=0;
do
{
  System.out.println(i);
i++;
}while(i<=5);
```

#3 Conditional statement Keywords

Keywords which are used for Conditional statements.

12. if : if keyword is used to create the if conditional statement. if the condition is true, then the if block gets executed.

```java
int i=0;
if(i==0)
{
        System.out.println("Correct");
}
```

13. else : else keyword is used when the if condition becomes false, and then the else block gets executed. It indicated the alternative branches in an if statement.

```java
int i=1;
if(i==0)
{
        System.out.println("Correct"); }
else
{
        System.out.println("Wrong");
}
```

14. switch : switch keyword is used to create the switch statement, which is used to execute different cases based on test value.

```java
switch (number)
{
        case 1:
        System.out.println("inside case 1");
        break;
        default:
        System.out.println("inside default case");
}
```

15. case : case keyword is used inside the switch-case block to create each case.

```java
      switch (number)
{
        case 1:
        System.out.println("inside case 1");
        break;
        case 2:
        System.out.println("inside case 2");
        break;
}
```

16. default : default keyword is used inside the switch-case block. If non of the cases matches, then the default block gets executed.

```
      switch (number)
{
        case 1:
        System.out.println("inside case 1");
        break;
        default:
        System.out.println("inside the default block");
}
```

17. continue : continue keyword is used to skip the statement following it inside the loop body and moves the control to the end of the loop body, and the control is automatically passed to the next iteration, and then the same loop body gets executed from the next round in usual manner.

```
    for(int i=0;i<=5;i++)
{
    if(i==3)
              continue;
    System.out.println(i);
}
```

18. break : break keyword is used to break out of a loop body or the switch block. The break statement is used for early exiting from the loop or switch statement at specified conditions.

```
    for(int i=0;i<=5;i++)
{
    if(i==3)
      break;
    System.out.println(i);
}
```

19. goto ** : Not in use and has no function.

#4 Function Keywords

Keyword which are used basically with some functions.

20. void : void keyword is used to specify a method which does not return any value.

```
void main()
```

21. return : return keyword in Java is used to return back the control along with the value from the called method(function) to the calling place.

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

#5 Object Keywords

Keywords which are related to objects in Java program.

22. new : new keyword in Java is used to create the instance of a class by dynamically allocating memory for a new object.

```
MyClass object = new MyClass();
```

23. this : this keyword in Java is used to refer to the current object inside a method or constructor. It is also used to distinguish between the instance variable and local variable.

```
public class MyClass
{
int a=50;
MyClass(int a)
{
this.a=a;
}

}
```

24. super : this keyword in Java is used to refer to the objects of the super class. It is used when we want to call the super class variable, method and constructor through sub-class object.

```
public class A
{
int a =30;
}
class B extends A
{
int a=40;
public void show()
{
System.out.println(a);
System.out.println(super.a);
}
}
```

25. instanceof : instanceof keyword is used to check if the given object is an instantiated object of the specified class or not. It returns the boolean value.

```
Test obj=new Test();
if(obj instanceof Test)
{
System.out.println("Yes");
}
else
{
System.out.println("No");
}
```

#6 Access Specifier Keywords

Keywords which are related to specify the accessibility or scope of a field, method, constructor or class.

26. **public :** public keyword in Java is used for classes, methods, variables and constructors, which makes them accessible from anywhere.

```
public class test{}
```

27. **private :** private keyword in Java is used to specify that a method, constructor or variable will be accessible only within the declared class.

```
private class test{}
```

28. **protected :** private keyword in Java is used to specify that a method, constructor or variable will be accessible within the package and outside the package through inheritance only.

```
protected class test{}
```

#7 Non-Access Modifier Keywords

keywords which do not have anything related to the level of access but they provide a special functionality when specified.

29. **final :** final keyword in Java is used with variables, methods and classes. A final variable is a constant value which cannot be changed. By making a method final, we cannot override it. By making a class final, we cannot extend it.

```
final class Test1 //final class
{
final int value =10; //final variable
final void show(){} // final method
}
```

30. **static :** static keyword in Java is used with variables, methods, blocks and classes. The static variables and methods get associated with the class as a whole rather than belonging to the individual object. The static keyword is used with variables and methods to make them constant for every instance of the class.

```
static int count; // static variable
public static void main(){} // static method
```

31. **abstract :** abstract keyword in Java is used to create abstract classes and abstract methods. The abstract methods do not have the body of themselves and must be overridden in all child classes. The class which contains an abstract method must be declared as an abstract class using the abstract keyword, and we can not instantiate that class.

```
      abstract class shape // abstract class
{
abstract double calculate(int x, int y); abstract method
}
```

32. synchronized : The synchronization keyword in Java is used to create a synchronization block that allows only a single thread to access the shared data or resources at a particular point of time. It specifies the critical sections or methods in multithreaded code.

```
synchronized void show(int num){}
```

33. transient : transient keyword in Java is used with instance variables to exclude them from the serialization process. It cannot be used with the static keyword.

```
transient int value=12;
```

34. volatile : volatile keyword in Java is used to indicate the visibility of variables modified by multiple threads during concurrent programming i.e every read or write of the volatile variable will be to the main memory and not the CPU cache.

```
static volatile int value=12;
```

35. strictfp : strictfp keyword in Java is used to make floating-point calculations platform independent.

Note : This keyword was added in JDK 1.2 version.

```
      strictfp double calculate(){
        double num1 = 10e+102;
        double num2 = 6e+08;
        return num1 + num2;
    }
```

36. native : native keyword in Java is used to create a method native which indicates that the method's implementation is also written in different languages like C and C++ in native code using JNI(Java Native Interface).

```
public native void test(){}
```

#8 Declarative Keywords

Keywords which are used to delcare something in Java progamming.

37. package : package keyword in Java is used to declare a Java package which includes classes, sub-packages, and interfaces.

```
package mypackage;
```

38. import : import keyword in Java is used to make classes and interfaces inside a package accessible to the current source code.

```
import javax.swing.JFrame;
```

39. class : class keyword in Java is used to declare a new class in Java.

```
class Test{}
```

40. interface : interface keyword in Java is used to declare an interface that only contains the abstract methods.

```
interface myInterface{
abstract void show();
}
class Test implements myInterface {
public void show()
{
System.out.println("Hello");
}
}
```

41. enum : enum keyword in Java is used to declare the enumerated(unchangeable) type, which are fixed set of constants. The constructors of enum are always private or default.

Note : This keyword was added in JDK 5.0 version.

```
class Test{
    enum WeekDays{
                    SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY;
                    }
    public static void main(String[] args){
        System.out.println(WeekDays.MONDAY);    }
}
```

42. const ** **:** Not in use and has no function.

#9 Inheritance Keywords

Keywords which are related to inheritance in Java programming

43. implements : implements keyword in Java is used to implement(kind of inheritance) an interface in a class. The interface contains only abstract methods, and to access those methods, another class must implement the interface using the implements keyword.

```
interface myInterface
{
abstract void show();
}
class Test implements myInterface
{
public void show()
{
System.out.println("Hello");
 }
}
```

44. extends : extends keyword in Java is used to inherit a new class from the base class using inheritance.

```
class Base{
public void showFun(){
System.out.println("In base class");
    }
}
class Child extends Base {
public void show(){
System.out.println("In child class");
    }
}
```

#10 Exception Keywords

Keyword which are related with exception handling in Java programming

45. throw : throw keyword in Java is used to throw an exception explicitly from a method or block of code. It is usually used with custom-made exceptions.

```
class Test{
        int a=47;
        int b=5;
        void divide()
        {
        if(b==5){
        throw new ArithmeticException();
        }
    }

    public static void main(String[] args){
            Test test = new Test();
            test.divide();
            }

    }
```

46. throws : throws keyword in Java is used to declare exceptions that can be thrown from a method during the program execution.

```
      void divide() throws ArithmeticException{
        int a=12;
        int b=0;
        System.out.println(a/b);
    }
```

47. try : try keyword in Java is used to create the try block, which is tested for any kind of exceptions while it is being executed. A try block must be followed by either catch or finally block.

```
try
    {
        int a=15;
        int b=0;
        System.out.println(a/b);
    }
    catch (ArithmeticException e)   {
        System.out.println("Divide by zero Error");
    }
```

48. catch : catch keyword in Java is used to create the catch block. When any exception occurs in the try block, it is caught by the catch block and is only used after the try block.

```
try
    {
        int a=15;
        int b=0;
        System.out.println(a/b);
    }
    catch (ArithmeticException e)   {
        System.out.println("Divide by zero Error");
    }
```

49. finally : finally keyword in Java is used to create the finally block, which is used after the try-catch block, and the finally block always gets executed whether an exception is found or not.

```
          finally
{
    System.out.println("This block will always execute whether exception is 
handled or not");
}
```

50. assert : assert keyword in Java is used to create an assertion(assumption), which is a condition that should be true during the program execution. during runtime if the assertion is false then an AssertionError is thrown.

Note : This keyword was added in JDK 1.4 version.

```
int age=20;
assert age>18: "age is greater than 18";
```

Points to Remember on Java Keywords

- true, false, and null are just like keywords, but they are actually literals
- You can’t use any of the above keywords as an identifier in your program