

Java Programming

Keywords

Java Keywords

abstract	class	finally	long	static	try
assert	continue	float	native	strictfp	void
boolean	default	for	new	super	volatile
break	do	if	package	switch	while
byte	double	implements	private	synchronized	
case	else	import	protected	this	LITERAL VALUES
catch	enum	instanceof	public	throw	true
catch	extends	int	return	throws	null
char	final	interface	short	transient	false

Dan McElroy



This presentation is offered under a [Creative Commons Attribution Non-Commercial Share](#) license. Content in this video can be considered under this license unless otherwise

Aug 2019

Context Sensitive Editors

Most of the editors used to write programs are context sensitive, meaning that they indent and change the color of the text based on the programming language.

Context Sensitive Editors

```
public class Grade {
    public static void main(String[ ] args) {
        int score = 83;
        char grade;
        // determine pass or fail
        if (score >= 70)
            grade = 'P';
        else
            grade = 'F';
        System.out.println ("Your grade is " + grade);
    } // end of public static void main(String [] args)
} end of public class Grade
```

NetBeans	Eclipse	Definition
blue	purple	Keyword (reserved)
gray	light-green	Comment
brown	blue	Literal value
black	black	No special meaning to Java

Keywords

Keywords are predefined reserved identifiers that have special meanings in Java. They cannot be used as identifiers or variable names in your program.

For example, the keyword **if** and **else** are part of the Java language.

Keywords

Don't worry about memorizing the list of key words for each language. As you write more programs, the keywords will become more familiar to you. The context sensitive editors help identify keywords if you accidentally try to use one of them as a variable name.

Keywords show up in:

blue on NetBeans

purple on Eclipse

violet on JDoodle

Java Keywords

abstract	class	finally	long	static	try
assert	continue	float	native	strictfp	void
boolean	default	for	new	super	volatile
break	do	if	package	switch	while
byte	double	implements	private	synchronized	
case	else	import	protected	this	<u>LITERAL VALUES</u>
catch	enum	instanceof	public	throw	true
catch	extends	int	return	throws	null
char	final	interface	short	transient	false

Keywords are defined in https://en.wikipedia.org/wiki/List_of_Java_keywords

The screenshot shows a web browser displaying the English Wikipedia page for "List of Java keywords". The page lists various Java keywords with their descriptions. The visible keywords include `abstract`, `assert`, `boolean`, `break`, `byte`, `case`, `catch`, `char`, and `class`. Each keyword has a detailed explanation of its usage and meaning in Java.

Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page
Print/export
Create a book
Download as PDF
Printable version
Languages

List [edit]

abstract
Abstracts are used to implement an abstraction in Java. A method with no definition must be declared as abstract and the class containing it must be declared as abstract. You cannot instantiate abstract classes. Abstract methods must be implemented in the sub classes. You cannot use the abstract keyword with variables or constructors.

assert (added in J2SE 1.4)
Assert describes a predicate (a true-false statement) placed in a Java program to indicate that the developer thinks that the predicate is always true at that place. If an assertion evaluates to false at run-time, an assertion failure results, which typically causes execution to abort. Optionally enable by ClassLoader method.

boolean
Defines a boolean variable for the values "true" or "false" only. By default, the value of boolean primitive type is false.

break
Used to end the execution in the current loop body.

byte
The `byte` keyword is used to declare a field that can hold an 8-bit signed two's complement integer.^{[4][5]} This keyword is also used to declare that a method returns a value of the primitive type `byte`.^{[6][7]}

case
A statement in the `switch` block can be labeled with one or more `case` or `default` labels. The `switch` statement evaluates its expression, then executes all statements that follow the matching `case` label; see `switch`.^{[8][9]}

catch
Used in conjunction with a `try` block and an optional `finally` block. The statements in the `catch` block specify what to do if a specific type of exception is thrown by the `try` block.

char
Defines a character variable capable of holding any character of the java source file's character set.

class
A type that defines the implementation of a particular kind of object. A class definition defines instance and class fields, methods, and inner classes as well as specifying the interfaces the class implements and the immediate superclass of the class. If the superclass is not explicitly specified, the superclass is implicitly `Object`.^[10] The `class` keyword can also be used in the form `Class.class` to get a `Class` object without needing an instance of that class. For example, `String.class` can be used instead of doing `new String().getClass()`.

You have reached the end of the discussion on Java Keywords