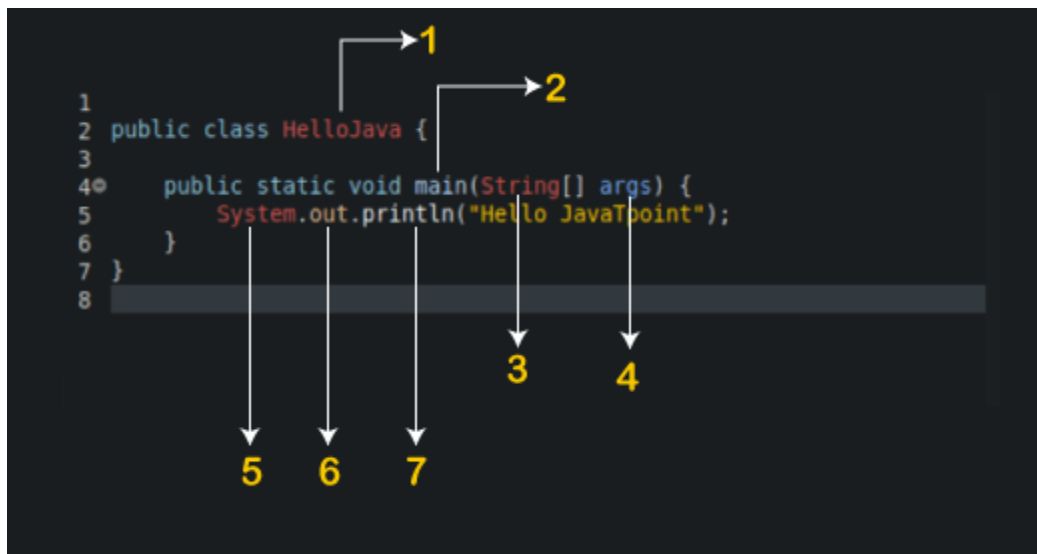


Identifiers in Java

Identifiers in Java are symbolic names used for identification. They can be a class name, variable name, method name, package name, constant name, and more. However, In [Java](#), There are some reserved words that can not be used as an identifier.

For every identifier there are some conventions that should be used before declaring them. Let's understand it with a simple Java program:

1. **public class** HelloJava {
2. **public static void** main(String[] args) {
3. System.out.println("Hello JavaTpoint");
4. }
5. }



From the above example, we have the following Java identifiers:

1. HelloJava (Class name)
2. main (main method)
3. String (Predefined Class name)
4. args (String variables)
5. System (Predefined class)

6. out (Variable name)
7. println (method)

let's understand the rules for Java identifier:

Rules for Identifiers in Java

There are some rules and conventions for declaring the identifiers in Java. If the identifiers are not properly declared, we may get a compile-time error. Following are some rules and conventions for declaring identifiers:

- A valid identifier must have characters [A-Z] or [a-z] or numbers [0-9], and underscore(_) or a dollar sign (\$). for example, @javatpoint is not a valid identifier because it contains a special character which is @.
- There should not be any space in an identifier. For example, java tpoint is an invalid identifier.
- An identifier should not contain a number at the starting. For example, 123javatpoint is an invalid identifier.
- An identifier should be of length 4-15 letters only. However, there is no limit on its length. But, it is good to follow the standard conventions.
- We can't use the Java reserved keywords as an identifier such as int, float, double, char, etc. For example, int double is an invalid identifier in Java.
- An identifier should not be any query language keywords such as SELECT, FROM, COUNT, DELETE, etc.

Java Reserved Keywords

Java reserved keywords are predefined words, which are reserved for any functionality or meaning. We can not use these keywords as our identifier names, such as class name or method name. These keywords are used by the syntax of Java for some functionality. If we use a reserved word as our variable name, it will throw an error.

In Java, every reserved word has a unique meaning and functionality.

Consider the below syntax:

1. **double** marks;

in the above statement, double is a reserved word while marks is a valid identifier.

Below is the list of reserved keywords in Java:

abstract	continue	for	protected	transient
Assert	Default	Goto	public	Try
Boolean	Do	If	Static	throws
break	double	implements	strictfp	Package
byte	else	import	super	Private
case	enum	Interface	Short	switch
Catch	Extends	instanceof	return	void
Char	Final	Int	synchronized	volatile
class	finally	long	throw	Date
const	float	Native	This	while

Although the const and goto are not part of the Java language; But, they are also considered keywords.

Example of Valid and Invalid Identifiers

Valid identifiers:

Following are some examples of valid identifiers in Java:

- TestVariable
- testvariable
- a
- i
- Test_Variable
- _testvariable

- \$testvariable
- sum_of_array
- TESTVARIABLE
- jtp123
- JavaTpoint
- Javatpoint123

Invalid identifiers:

Below are some examples of invalid identifiers:

- Test Variable (We can not include a space in an identifier)
- 123javatpoint (The identifier should not begin with numbers)
- java+tpoint (The plus (+) symbol cannot be used)
- a-javatpoint (Hyphen symbol is not allowed)
- java_&_Tpoint (ampersand symbol is not allowed)
- Java'tpoint (we can not use an apostrophe symbol in an identifier)

We should follow some naming convention while declaring an identifier. However, these conventions are not forced to follow by the Java programming language. That's why it is called conventions, not rules. But it is good to follow them. These are some industry standards and recommended by Java communities such as Oracle and Netscape.

If we do not follow these conventions, it may generate confusion or erroneous code.