

Identifier in Java

An **identifier in Java** is the name given to **Variables, Classes, Methods, Packages, Interfaces**, etc. These are the unique names used to identify programming elements. Every **Java Variable** must be identified with a unique name.

Example:

```
public class Test
{
    public static void main(String[] args)
    {
        int a = 20;
    }
}
```

In the above Java code, we have 5 identifiers as follows:

- **Test**: Class Name
- **main**: Method Name
- **String**: Predefined Class Name
- **args**: Variable Name
- **a**: Variable Name

Rules For Naming Java Identifiers

There are certain rules for defining a valid Java identifier. These rules must be followed, otherwise, we get a compile-time error. These rules are also valid for other languages like C and C++.

- The only allowed characters for identifiers are all alphanumeric characters([A-Z],[a-z],[0-9]), '\$'(dollar sign) and '_' (underscore). For example, "geek@" is not a valid Java identifier as it contains a '@', a special character.
- Identifiers should **not** start with digits(**[0-9]**). For example, "123geeks" is not a valid Java identifier.
- Java identifiers are **case-sensitive**.
- There is no limit on the length of the identifier, but it is advisable to use an optimum length of 4 - 15 letters only.
- **Reserved Words** can't be used as an identifier. For example, "int while = 20;" is an invalid statement as a while is a reserved word.

Note: Java has 53 reserved words (including 50 keywords and 3 literals), that are not allowed to be used as identifiers.

Examples of Valid Identifiers

```
MyVariable  
MYVARIABLE  
myvariable  
x  
i  
x1  
i1  
_myvariable  
$myvariable  
sum_of_array  
geeks123
```

Examples of Invalid Identifiers

My Variable // contains a space

123geeks // Begins with a digit

a+c // plus sign is not an alphanumeric character

variable-2 // hyphen is not an alphanumeric character

sum_&_difference // ampersand is not an alphanumeric character

Reserved Words in Java

Any programming language reserves some words to represent functionalities defined by that language. These words are called reserved words. They can be briefly categorized into two parts:

- **Keywords** (50): Keywords define functionalities. The below table shows list of 50 keywords.
- **literals** (3): Literals define value. The three literals are, **true**, **false** and **null**.

Identifiers are stored by **symbol tables** and used during the **lexical**, **syntax**, and **semantic** analysis phase of compilation.

List of Java Reserved Words

abstract	continue	for	protected	transient
Assert	Default	Goto	public	Try
Boolean	Do	If	Static	throws
break	double	implements	strictfp	Package

abstract	continue	for	protected	transient
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

Note: The keywords **const** and **goto** are reserved, even though they are not currently used in Java. In place of const, the final keyword is used. Some keywords like strictfp are included in later versions of Java.