Identifiers in Java

**Java Identifiers**

In programming languages, identifiers are used for identification purposes. In Java, an identifier can be a class name, method name, variable name, or label. For example :

public class Test

{

public static void main(String[] args)

{

int a = 20;

}

}

In the above java code, we have the following identifiers namely:

* **Test** : class name.
* **main** : method name.
* **a** :  variable name.

**Rules for defining 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 such as C, 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 contain '@', which is a special character.
* Identifiers should **not** start with digits(**[0-9]**). For example "123geeks" is a 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 while is a reserved word. There are **53** reserved words in Java.

**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**  
  
Any programming language reserves some words to represent functionalities defined by that language. These words are called reserved words. They can be briefly categorised into two parts: **keywords**(50) and **literals**(3).  
  
Keywords define functionalities, and literals define a value.

List of keywords or reserved words:

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

Identifiers are used by symbol tables in various analyzing phases(like lexical, syntax, semantics) of a compiler architecture.  
  
**Note :** The keywords const and goto are reserved, even though they are not currently used. In place of const, the final keyword is used. Some keywords like [strictfp](https://www.cdn.geeksforgeeks.org/strictfp-keyword-java/" \t "_blank)are included in later versions of Java.