

## C tokens, Identifiers and Keywords

C tokens, Identifiers and Keywords are the basics in a C program. All are explained in this page with definition and simple example programs.

### 1. C tokens:

- C tokens are the basic building blocks in C language which are constructed together to write a C program.
- Each and every smallest individual units in a C program are known as C tokens.
- C tokens are of six types. They are,
  1. Keywords (eg: int, while),
  2. Identifiers (eg: main, total),
  3. Constants (eg: 10, 20),
  4. Strings (eg: "total", "hello"),
  5. Special symbols (eg: {}, {}),
  6. Operators (eg: +, /, -, \*)

*C tokens example program:*

```
int main()
{
int x, y, total;
x = 10, y = 20;
total = x + y;
Printf ("Total = %d \n", total);
}
.
```

where,

- main – identifier
- {}, (,) – delimiter
- int – keyword
- x, y, total – identifier
- main, {, }, (, ), int, x, y, total – tokens

Do you know how to use C token in real time application programs? We have given simple real time application programs where C token is used. You can refer the below C programs to know how to use C token in real time program.

### 2. Identifiers in C language:

- Each program elements in a C program are given a name called identifiers.
- Names given to identify Variables, functions and arrays are examples for identifiers. eg. x is a name given to integer variable in above program.

*Rules for constructing identifier name in C:*

1. First character should be an alphabet or underscore.
2. Succeeding characters might be digits or letter.
3. Punctuation and special characters aren't allowed except underscore.
4. Identifiers should not be keywords.

### 3. Keywords in C language:

- Keywords are pre-defined words in a C compiler.
- Each keyword is meant to perform a specific function in a C program.
- Since keywords are referred names for compiler, they can't be used as variable name.

C language supports 32 keywords which are given below.

<i>auto</i>	<i>double</i>	<i>int</i>	<i>struct</i>	<i>const</i>	<i>float</i>	<i>short</i>	<i>unsigned</i>
<i>break</i>	<i>else</i>	<i>long</i>	<i>switch</i>	<i>continue</i>	<i>for</i>	<i>signed</i>	<i>void</i>
<i>case</i>	<i>enum</i>	<i>register</i>	<i>typedef</i>	<i>default</i>	<i>goto</i>	<i>sizeof</i>	<i>volatile</i>
<i>char</i>	<i>extern</i>	<i>return</i>	<i>union</i>	<i>do</i>	<i>if</i>	<i>static</i>	<i>while</i>