

C Keywords

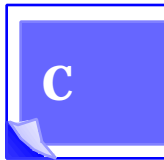


C Keywords

There are several keywords that are reserved as a part of the C programming language. These words shouldn't be used for anything other than their intended uses:

auto	C variable default storage class. auto int x;
break	An unconditional exit command. Used to immediately exit a for, while, switch, or do...while statement. while(TRUE) { commands if(done == TRUE) { break; } }
case	Used in conjunction with the switch statement. switch(x) { case 'a': commands break; case 'b': commands break; }
char	The smallest C data type. int char ch = 'A';
const	A C data modifier. const float pi = 3.14159;
continue	A reset command used to jump to the next instance in a for, while, or do...while statement. for(ctr = 1; ctr <100; ctr++) { if(ctr % 2) continue; commands; }
default	A case within a switch that is used when none of the other cases are met. switch(x) { case 'a': commands

	<pre> break; case 'b': commands break; default: commands break; } </pre>
do	Used in conjunction with the <code>while</code> statement. This allows looping to occur until the <code>while</code> statement evaluates to FALSE. The loop is executed at least one time.
	<pre> do { commands; }while (condition); </pre>
double	A data type used to hold double-precision floating point values.
	<code>double num = 912340000000.123;</code>
else	Used to signal commands that should be executed when an <code>if</code> statement evaluates to FALSE.
	<pre> if(condition) { commands } else { commands } </pre>
enum	A data type that can be used to create variables that accept a limited number of predetermined values.
	<code>enum card_suit = { CLUB, DIAMOND, HEART, SPADE };</code>
extern	A data modifier that signals that the variable will be declared in another part of the program.
	<code>extern global_variable;</code>
float	A data type used to hold floating-point values.
	<code>float num = 123.456;</code>
for	An iterative looping command that allows a portion of code to be executed a specific number of times. A <code>for</code> statement contains <i>initialization</i> , <i>incrementation</i> , and <i>conditional</i> sections.
	<pre> for(ctr = 1; ctr <= 10; ctr++) { commands } </pre>

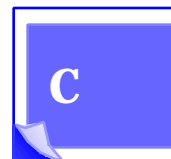




C Keywords

<code>goto</code>	<p>Transfers program control to a label elsewhere in the program.</p> <pre>goto label ; commands label : commands</pre>
<code>if</code>	<p>Allows for branching within program flow. If the expression following the <code>if</code> evaluates to TRUE, then the following command(s) are executed. Also used in conjunction with the <code>else</code> command.</p> <pre>if(ctr == 1) { commands } else { commands }</pre>
<code>int</code>	<p>A C data type used to hold a small integer value. Specific values that can be held are dependent upon CPU's integer size. Generally stores numbers in 2 bytes.</p> <pre>int nbr = 150;</pre>
<code>long</code>	<p>A C data type used to hold larger integer values. Generally stores numbers in 4 bytes.</p> <pre>long nbr = 100000;</pre>
<code>register</code>	<p>A C data modifier used to signal that a C variable should be stored in one of the CPU's registers, if available.</p> <pre>register int number;</pre>
<code>return</code>	<p>A C command that causes a function to end and return to the calling function. The <code>return</code> statement may return a single value to the calling function.</p> <pre>function() { commands; return; }</pre>
<code>short</code>	<p>A C data type used to hold a small integer value. Generally stores numbers in 2 bytes.</p> <pre>short num = 123;</pre>

signed	<p>A C data modifier that allows both positive and negative numbers to be stored in a variable.</p> <pre>signed int num = -100;</pre>
sizeof	<p>A C operator (rather than a command) that returns the size of its argument in bytes.</p> <pre>sizeof(x);</pre>
static	<p>A C data modifier that allows a variable to retain its contents.</p> <pre>static int ctr = 0;</pre>
struct	<p>A C construct that allows the grouping of several C data types.</p> <pre>struct name { char first[15]; char last[20]; };</pre>
switch	<p>A C command that allows program flow to be channeled to various conditions. case statements are used to present the conditions.</p> <pre>switch(x) { case 'a': commands break; case 'b': commands break; default: commands break; }</pre>
typedef	<p>A C modifier that allows new names to be created for data types.</p> <pre>typedef pos_nbr unsigned int; pos_nbr x; x = 100;</pre>
union	<p>A construct that allows multiple C data types to be stored in the same memory location.</p> <pre>union x { long nmb; char str[4]; };</pre>
unsigned	<p>A C modifier that allows a variable to hold only positive numbers.</p> <pre>unsigned int nbr = 100;</pre>





C Keywords

<code>void</code>	Used to create a generic pointer that can be cast to any type. Can also be used with function declarations to signify that there is not a returned value. <code>void function();</code> or <code>void char *ptr;</code>
<code>volatile</code>	Opposite of <code>const</code> . A data modifier that allows a variable's contents to be altered. <code>volatile int x = 100;</code>
<code>while</code>	A looping command that executes until its condition evaluates to TRUE or until a <code>break</code> command causes an exit. <code>while(condition)</code> { commands }