The C-Language Keywords and Symbols

Keywords

MEMORY
01.void
02.char
03.int
04.short
05.long
06.float
07.double
08.signed
09.unsigned
10.struct
11.union
12.enum
13.const
14.volatile
15.auto
16.extern
17.static
18.register
19.typedef
20.sizeof

CONTROL
21.return
22.if
23.else
24.switch
25.case
26.default
27.while
28.do
29.for
30.break
31.continue
32.goto

Symbols

CONTROL
#
< >
//
/* */
( )
{ }
;
,
“
‘
=
[ ]
:
?
.
\

MEMORY
&
*

LOGIC
==
!=
<
>
<=
>=

&&
||

!

MATH
*
%
/
+
-

BIT OP
|
&
^
~
<<
>>

loop

Loop is a PROGRAM structure that allow us to do repetitions based on a LOGIC_NUMBER evaluation. As long as the LOGIC_NUMBER is not 0, the body of a loop will be repeated

LOGIC_NUMBER is either 1 (TRUE) or 0 (FALSE)
(any numbers that is not 0 or 1 is considered as 1)

- **break** Keyword

- “break;” placed anywhere in the body, will jump to the code after the body closing curly bracket }. Regardless of the LOGIC_NUMBER

- can be used with switch/case, while, do/while and for structure

- **continue** Keyword

- “continue;” placed anywhere in the body will jump direct to LOGIC_NUMBER evaluation ignoring all the codes after it

- can be used with while, do/while and for structure

while and do/while

The "while" and "do/while" are very similar where both will do repetititons and both will exit the loop when LOGIC_NUMBER evaluation is 0

"while" will evaluate the LOGIC_NUMBER, then only it will run the codes in the body section

"do/while" will run the body section first, then only evaluate the LOGIC_NUMBER.

"while" structure
Part1:"while" Keyword
Part2:LOGIC_NUMBER – Numbers other than 0 and 1 will be considered as 1
Part3:while_body

Part2:LOGIC_NUMBER
– placed within the bracket ( ) pair
– either 1 or 0

Part1:"while" Keyword

```
while(LOGIC_NUMBER)
{


}
```

Part3:while_body
– multiple "instruction codes" within a Curly Bracket { } pair

"do while" structure
Part1:"do" Keyword
Part2:dowhile_body
Part3:"while" Keyword
Part4:LOGIC_NUMBER – Numbers other than 0 and 1 will be considered as 1

Part1:"do" Keyword

```
do
{


} while(LOGIC_NUMBER);
```

Part2:dowhile_body
– multiple "instruction codes" within a Curly Bracket { } pair

Part3:"while" Keyword

Part4:LOGIC_NUMBER
– placed within the bracket ( ) pair
– either 1 or 0
– followed by a semi-colon ; after the closing bracket)

Arduino IDE|Save PROGRAM as: **c_while_basic**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the while loop\n");
  while(while_logic) {
    counter++;
    if (counter <= 10) {
      Serial.print("\ncounter in while loop:");
      Serial.print(counter);
    } else {
      while_logic = 0; // next while(0) evaluation, will exit
    }
  }
  Serial.print("\n\nEnd of while");
}
void loop(){}
```

Arduino IDE|Save PROGRAM as: **c_do_while_basic**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the do while loop\n");
  do {
    counter++;
    if (counter <= 10) {
      Serial.print("\ncounter in do while loop:");
      Serial.print(counter);
    } else {
      while_logic = 0; // next while(0) evaluation, will exit
    }
  } while(while_logic);
  Serial.print("\n\nEnd of do while");
}               } while(while_logic);
void loop(){}
```

ATMEGA328/ARDUINO - C_LANGUAGE - LOOP
https://github.com/teaksoon/lmaewapm

Arduino IDE|Save PROGRAM as: **c_while_break**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the while loop\n");
  while(while_logic) {
    counter++;
    if (counter <= 10) {
      Serial.print("\ncounter in while loop:");
      Serial.print(counter);
    }
    else {
      break; // ignore while_logic, just exit while loop
    }
  }
  Serial.print("\n\nEnd of while loop");
}
void loop(){}
```

Arduino IDE|Save PROGRAM as: **c_do_while_break**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the do while loop\n");
  do {
   counter++;
    if (counter <= 10) {
      Serial.print("\ncounter in do while loop:");
      Serial.print(counter);
    }
    else {
      break; // ignore while_logic, just exit while loop
    }
  } while(while_logic);
  Serial.print("\n\nEnd of while loop");
}
void loop(){}
```

Arduino IDE|Save PROGRAM as: **c_while_continue**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the while loop\n");
  while(while_logic) {
    counter++;
    if ( is_even(counter) == 1) {
      Serial.print("\nskipped using continue");
      continue; // go back to evaluate while(while_logic)
    }
    if (counter <= 10) {
      Serial.print("\ncounter in while loop:");
      Serial.print(counter);
    } else {
      while_logic = 0; // next while(0) evaluation, will exit
    }
  }
  Serial.print("\n\nEnd of while loop");
}
void loop(){}

int is_even(int src_num) {
  if ( (src_num % 2) == 0)
    return 1;
  else
    return 0;
}
```

Arduino IDE|Save PROGRAM as: **c_do_while_continue**
Enter codes below and upload. Use the Serial Monitor to see results

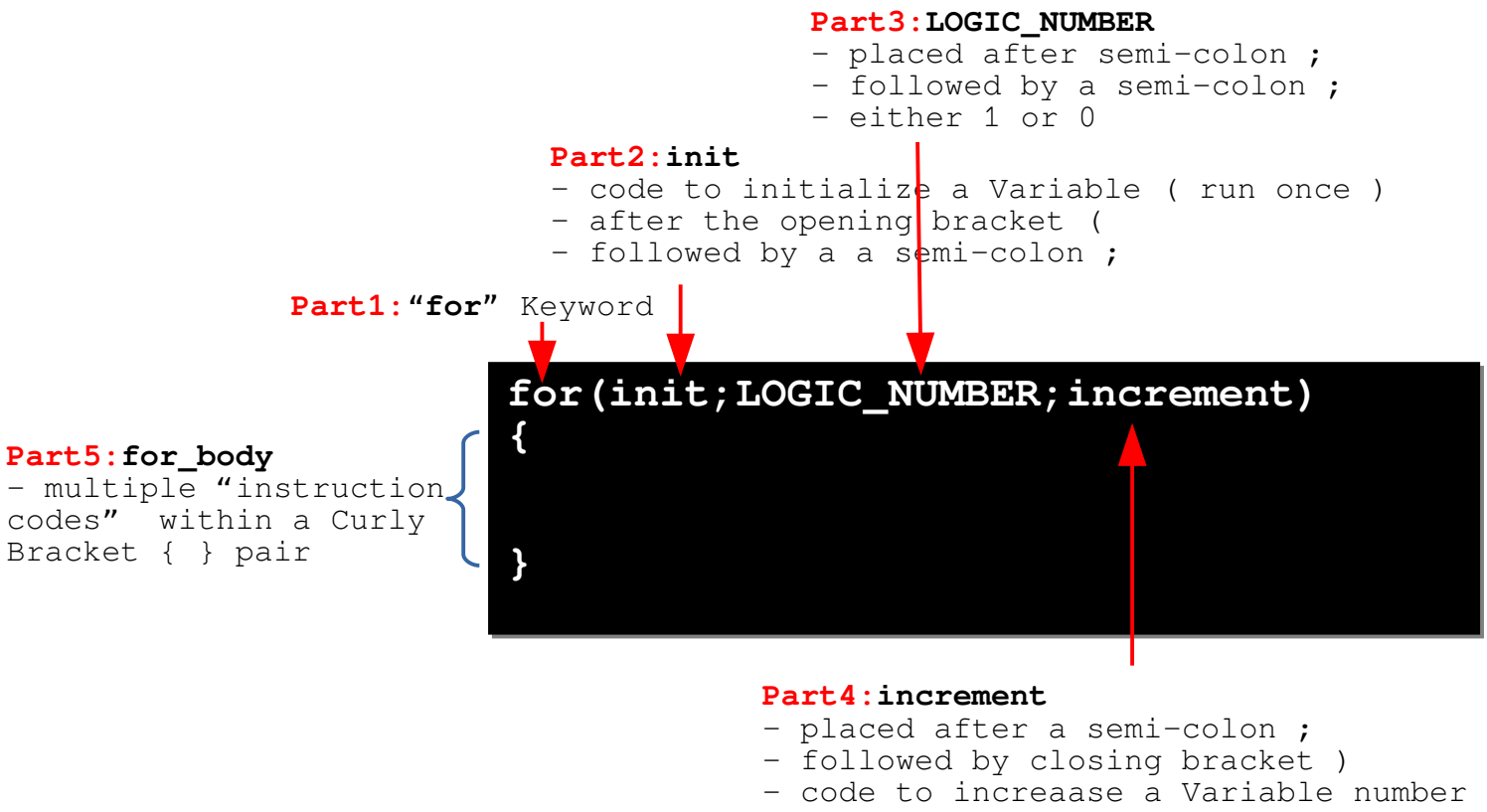
```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
int counter = 0;
int while_logic = 1;
  Serial.print("\n\nBefore entering the do while loop\n");
  do {
    counter++;
    if ( is_even(counter) == 1) {
      Serial.print("\nskipped using continue");
      continue; // go back to evaluate while(while_logic)
    }
    if (counter <= 10) {
      Serial.print("\ncounter in do while loop:");
      Serial.print(counter);
    } else {
      while_logic = 0; // next while(0) evaluation, will exit
    }
  } while(while_logic);
  Serial.print("\n\nEnd of do while");
}
void loop(){}

int is_even(int src_num) {
  if ( (src_num % 2) == 0)
    return 1;
  else
    return 0;
}
```

for

The "for" is also a loop where it will exit the loop when LOGIC_NUMBER evaluation is 0

Apart from the LOGIC_NUMBER evaluation, the "for" structure also have two other functions, Variable Initialisation and Variable Increment for each loop cycle

"for" structure
Part1:"for" Keyword
Part2:init – a variable intial value
Part3:LOGIC_NUMBER – Numbers other than 0 and 1 will be considered as 1
Part4:increment – increase variable value on each loop cycle
Part5:for_body



Arduino IDE|Save PROGRAM as: **c_for_basic**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
  Serial.print("\n\nBefore entering the for loop\n");
  for (int counter=1; counter <= 10; counter++) {
    Serial.print("\ncounter in for loop:");
    Serial.print(counter);
  }
  Serial.print("\n\nEnd of for");
}
void loop(){}
```

Arduino IDE|Save PROGRAM as: **c_for_break**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
  Serial.print("\n\nBefore entering the for loop\n");
  for (int counter=1; counter <= 10; counter++) {
    Serial.print("\ncounter in for loop:");
    Serial.print(counter);
    if (counter >= 5) {
      break; // will exit for loop immediately
    }
  }
  Serial.print("\n\nEnd of for");
}
void loop(){}
```

Arduino IDE|Save PROGRAM as: **c_for_continue**
Enter codes below and upload. Use the Serial Monitor to see results

```
void setup() {
  Serial.begin(9600);Serial.print("\n\nSerial Monitor(9600)...");
  Serial.print("\n\nBefore entering the for loop\n");
  for (int counter=1; counter <= 10; counter++) {
    if ( is_even(counter) == 1) {
      Serial.print("\nskipped using continue");
      continue; // go back to evaluate for(;LOGIC_NUMBER;)
    }
    Serial.print("\ncounter in for loop:");
    Serial.print(counter);
  }
  Serial.print("\n\nEnd of for");
}
void loop(){}

int is_even(int src_num) {
  if ( (src_num % 2) == 0)
    return 1;
  else
    return 0;
}
```