

A Comparison Of Three Programming Languages:

**C Programming
Language**

**Java Programming
Language**

**Python Programming
Language**

"HELLO, WORLD!" PROGRAMS

C

```
#include <stdio.h>

int main() {
    printf("Hello, World!");
}
```

Java

```
public class HelloWorld {
    public static void main(String args[]) {
        System.out.print("Hello, World!");
    }
}
```

Python

```
# PROGRAM HelloWorld

print("Hello, World!")

# END.
```

	C	Java	Python
Variables	<pre>int IntegerVar = 1; float RealVar = 3.14; char CharVar = 'D'; C has no string type C has no boolean type</pre>	<pre>int IntegerVar = 1; float RealVar = 3.14; char CharVar = 'D'; String StringVar = "Hello"; boolean BoolVar = true;</pre>	<pre>IntegerVar = 1 RealVar = 3.14 CharVar = 'D' StringVar = "Hello" BoolVar = True</pre>
Escape Sequences	<pre>\b - backspace \f - form feed \n - newline \r - carriage return \t - tab \' - single quote character \" - double quote character \\ - backslash character</pre>	<pre>\b - backspace \f - form feed \n - newline \r - carriage return \t - tab \' - single quote character \" - double quote character \\ - backslash character</pre>	<pre>\b - backspace \f - form feed \n - newline \r - carriage return \t - tab \' - single quote character \" - double quote character \\ - backslash character</pre>
Print "Hello"	<pre>printf("Hello\n");</pre>	<pre>System.out.print("Hello\n");</pre>	<pre>print("Hello\n")</pre>
IF Statement	<pre>if (Condition) { // Statement(s); } else { // Statement(s); }</pre>	<pre>if (Condition) { // Statement(s); } else { // Statement(s); }</pre>	<pre>if (Condition): # Statement(s) else: # Statement(s) # Endif;</pre>

	C	Java	Python
CASE Statement	<pre>switch(var) { case x: // Statement(s) break; case y: // Statement(s) break; default: // Statement(s) }</pre>	<pre>switch(var) { case x: // Statement(s) break; case y: // Statement(s) break; default: // Statement(s) }</pre>	<pre>match VALUE: case x: STATEMENT(S) case y: STATEMENT(S) case _: STATEMENT(S) # EndCase;</pre>
WHILE Statement	<pre>while (Condition) { // Statement(s); }</pre>	<pre>while (Condition) { // Statement(s); }</pre>	<pre>while (Condition): # Statement(s) # EndWhile;</pre>
FOR Statement	<pre>for (INIT; COND; INC) { // Statement(s); }</pre>	<pre>for (INIT; COND; INC) { // Statement(s); }</pre>	<pre>for VAL in SET-OF-VALS: # Statement(s) # EndFor;</pre>
DO Statement	<pre>do { // Statement(s); } while (Condition);</pre>	<pre>do { // Statement(s); } while (Condition);</pre>	<p>Python does not have a DO statement</p>

	C	Java	Python																																																
Indexing Arrays	Index starts at 0 <table border="1"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>16</td><td>18</td><td>23</td><td>33</td><td>34</td><td>42</td><td>44</td><td>54</td></tr> </table>	0	1	2	3	4	5	6	7	16	18	23	33	34	42	44	54	Index starts at 0 <table border="1"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>16</td><td>18</td><td>23</td><td>33</td><td>34</td><td>42</td><td>44</td><td>54</td></tr> </table>	0	1	2	3	4	5	6	7	16	18	23	33	34	42	44	54	Index starts at 0 <table border="1"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>16</td><td>18</td><td>23</td><td>33</td><td>34</td><td>42</td><td>44</td><td>54</td></tr> </table>	0	1	2	3	4	5	6	7	16	18	23	33	34	42	44	54
0	1	2	3	4	5	6	7																																												
16	18	23	33	34	42	44	54																																												
0	1	2	3	4	5	6	7																																												
16	18	23	33	34	42	44	54																																												
0	1	2	3	4	5	6	7																																												
16	18	23	33	34	42	44	54																																												
Declaring Arrays	<code>int Age[3];</code>	<code>int[] Age;</code> <code>int Age[];</code> <code>int[] Age = new int[3];</code>	<code>Age = []</code>																																																
Initializing Arrays	<code>int Age[3] = {45,24,43};</code>	<code>int[] Age = {45,24,43};</code>	<code>Age = [45,24,43]</code>																																																
Array Length	<code>int ALen = sizeof Age / sizeof Age[0];</code>	<code>int ALen = Age.length;</code>	<code>ALen = len(Age)</code>																																																
Printing Arrays	<pre>for(i=0; I < ALen; i++){ printf("%d",Age[i]); }</pre>	<pre>import java.util.Arrays; : String APrint = Arrays.toString(Age); System.out.println(APrint);</pre>	<code>print(Age)</code>																																																
Reversing an Array	C does not have a built-in reverse array function.	<pre>import java.util.*; : Collections.reverse(Arrays.asList(Age));</pre>	<code>Age.reverse()</code>																																																

	C	Java	Python
Declaring a Method with no return value	<pre>void MethodName (Params) { // Method Code }</pre>	<pre>static void MethodName (Params) { // Method Code }</pre>	<pre>def MethodName (Params): # Method Code # END MethodName.</pre>
Declaring a Method with an integer return value	<pre>int MethodName (Params) { // Method Code return Var; }</pre>	<pre>static int MethodName (Params) { // Method Code return Var; }</pre>	<pre>def MethodName (Params): # Method Code return Var # END MethodName.</pre>
Declaring a Class	C does not have classes.	<pre>public class ClassName { // Class Code }</pre>	<pre>class ClassName: # Class Code # END ClassName.</pre>
Instantiating a Class	C does not have classes.	<pre>ClassName c1 = new ClassName () ClassName c2 = new ClassName ()</pre>	<pre>c1 = ClassName () c2 = ClassName ()</pre>

	C	Java	Python
Inheritance	C does not have inheritance.	<pre>class Parent { ... } class Child extends Parent { ... }</pre>	<pre>class Parent: ... # END ClassName. class Child(Parent): ... # END ClassName.</pre>