**Objective C Tutorial**

Objective-C is an object-oriented programming language.

It is the main programming language used by Apple for the OS X and iOS operating systems and their respective APIs, Cocoa and Cocoa Touch.

#import <Foundation/Foundation.h>

int main (int argc, **const** char \* argv[])

{

NSAutoreleasePool \* pool = [[NSAutoreleasePool alloc] init];

NSLog (@**"hello world"**);

[pool drain];

return 0;

}

**Objective-C Variables**

|  |  |  |
| --- | --- | --- |
| **Type** | **Storage size** | **Value range** |
| char | 1 byte | -128 to 127 or 0 to 255 |
| unsigned char | 1 byte | 0 to 255 |
| signed char | 1 byte | -128 to 127 |
| int | 2 or 4 bytes | -32,768 to 32,767 or -2,147,483,648 to 2,147,483,647 |
| unsigned int | 2 or 4 bytes | 0 to 65,535 or 0 to 4,294,967,295 |
| short | 2 bytes | -32,768 to 32,767 |
| unsigned short | 2 bytes | 0 to 65,535 |
| long | 4 bytes | -2,147,483,648 to 2,147,483,647 |
| unsigned long | 4 bytes | 0 to 4,294,967,295 |

**Arithmetic Operators**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Adds two operands |
| - | Subtracts second operand from the first |
| \* | Multiplies both operands |
| / | Divides numerator by denominator |
| % | Modulus Operator and remainder of after an integer division |
| ++ | Increment operator increases integer value by one |
| -- | Decrement operator decreases integer value by one |

**Relational Operators**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | Checks if the values of two operands are equal or not. |
| != | Checks if the values of two operands are not equal. |
| > | Checks if the value of left operand is greater than the value of right operand. |
| < | Checks if the value of left operand is less than the value of right operand. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand. |

**Logical Operators**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| && | Logical AND operator. If both the operands are non zero then condition becomes true. |
| || | Logical OR Operator. If any of the two operands is non zero then condition becomes true. |
| ! | Logical NOT Operator. Reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make false. |

**Assignment Operators**

|  |  |
| --- | --- |
| **Operator** | **Description** |
| = | Assignment operator, Assigns values from right side operands to left side operand |
| += | Add AND assignment operator, It adds right operand to the left operand and assigns the result to left operand |
| -= | Subtract AND assignment operator, It subtracts right operand from the left operand and assigns the result to left operand |
| \*= | Multiply AND assignment operator, It multiplies right operand with the left operand and assigns the result to left operand |
| /= | Divide AND assignment operator, It divides left operand with the right operand and assigns the result to left operand |
| %= | Modulus AND assignment operator, It takes modulus using two operands and assigns the result to left operand |
| <<= | Left shift AND assignment operator |
| >>= | Right shift AND assignment operator |
| &= | Bitwise AND assignment operator |
| ^= | bitwise exclusive OR and assignment operator |
| |= | bitwise inclusive OR and assignment operator |