OBJECTIVE-C BASIC SYNTAX

http://www.tutorialspoint.com/objective_c/objective_c_basic_syntax.htm

You have seen a basic structure of Objective-C program, so it will be easy to understand other basic building blocks of the Objective-C programming language.

Tokens in Objective-C

A Objective-C program consists of various tokens and a token is either a keyword, an identifier, a constant, a string literal, or a symbol. For example, the following Objective-C statement consists of six tokens:

```
NSLog(@"Hello, World! \n");
```

The individual tokens are:

```
NSLog
@
(
"Hello, World! \n"
)
;
```

Semicolons ;

In Objective-C program, the semicolon is a statement terminator. That is, each individual statement must be ended with a semicolon. It indicates the end of one logical entity.

For example, following are two different statements:

```
NSLog(@"Hello, World! \n");
return 0;
```

Comments

Comments are like helping text in your Objective-C program and they are ignored by the compiler. They start with /* and terminate with the characters */ as shown below:

```
/* my first program in Objective-C */
```

You can not have comments with in comments and they do not occur within a string or character literals.

Identifiers

An Objective-C identifier is a name used to identify a variable, function, or any other user-defined item. An identifier starts with a letter A to Z or a to z or an underscore _ followed by zero or more letters, underscores, and digits $0to9$.

Objective-C does not allow punctuation characters such as @, $, and % within identifiers. Objective-C is a **case-sensitive** programming language. Thus, *Manpower* and *manpower* are two different identifiers in Objective-C. Here are some examples of acceptable identifiers:

```
mohd       zara    abc   move_name  a_123
myname50   _temp   j     a23b9      retVal
```

Keywords

The following list shows few of the reserved words in Objective-C. These reserved words may not be used as constant or variable or any other identifier names.

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double	protocol	interface	implementation
NSObject	NSInteger	NSNumber	CGFloat
property	nonatomic;	retain	strong
weak	unsafe_unretained;	readwrite	readonly

Whitespace in Objective-C

A line containing only whitespace, possibly with a comment, is known as a blank line, and an Objective-C compiler totally ignores it.

Whitespace is the term used in Objective-C to describe blanks, tabs, newline characters and comments. Whitespace separates one part of a statement from another and enables the compiler to identify where one element in a statement, such as int, ends and the next element begins. Therefore, in the following statement:

```
int age;
```

There must be at least one whitespace character *usuallyaspace* between int and age for the compiler to be able to distinguish them. On the other hand, in the following statement,

```
fruit = apples + oranges;   // get the total fruit
```

no whitespace characters are necessary between fruit and =, or between = and apples, although you are free to include some if you wish for readability purpose.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js