# Defined Types

## BOOL

**DECLARED IN**      objc/objc.h

**SYNOPSIS**                                                                                   typedef char **BOOL**;

**DESCRIPTION**      This type carries the basic boolean distinction between YES and NO (true and false).

## Class

**DECLARED IN**      objc/objc.h

**SYNOPSIS**                                                                              typedef struct objc_class ***Class**;

**DESCRIPTION**      Class is the data type for class objects.   The **objc_class** structure it refers to holds
information compiled from the class definition; details of its contents can be found in Chapter 15,
ªRun-Time System.º

Every object has an **isa** instance variable of this type, which enables the object to identify its class.

Class objects can also be assigned to type **id**.   But just as instances of a class can be statically typed
by using the class name, class objects can be more particularly typed with the Class data type.

## id

**DECLARED IN**      objc/objc.h

**SYNOPSIS**                                                                        typedef struct objc_object {
        Class isa;
    } ***id**;

**DESCRIPTION**      The **id** data type designates an Objective C object of any class.   All objects, including
both instances and class objects, can be assigned to this type.

## IMP

**DECLARED IN**      objc/objc.h

**SYNOPSIS**                                                                              typedef id (***IMP**) (id, SEL, ...);

**DESCRIPTION**      This is the data type returned by Object's **methodFor:** method to identify a method
implementation.   It's defined as a pointer to a function that returns an **id** and takes an object (**self**)

and a selector (**_cmd**) as its first two arguments.

## SEL

**DECLARED IN**   objc/objc.h

**SYNOPSIS**                                                                                 typedef struct objc_selector ***SEL**;

**DESCRIPTION**   The SEL type identifies method selectors.   Valid SEL values are assigned only by the run-time system.   They are never 0.

## STR

**DECLARED IN**   objc/objc.h

**SYNOPSIS**                                                                                 typedef char ***STR**;

**DESCRIPTION**   This type is a rarely used shorthand for a character string.   It's mainly of historical interest.

# Symbolic Constants

### Boolean Constants

**DECLARED IN**   objc/objc.h

**SYNOPSIS**                                                   YES   (BOOL)1
  NO                                                (BOOL)0

**DESCRIPTION**   YES and NO are the standard values assigned to BOOL variables.

### Empty Objects

**DECLARED IN**   objc/objc.h

**SYNOPSIS**                                                       nil     (id)0
  Nil                                                (Class)0

**DESCRIPTION**   **nil** is the common notation for a NULL object.   **Nil** is sometimes used for a NULL class object, but **nil** typically serves this purpose as well.