

Objective-C Cheat Sheet

Declaring Variables

Primitive Variables

type name
↓ ↓
`int number;`

Non-Primitive Variables (Objects)

type name
↓ ↓
`@property (strong, nonatomic) UIButton *typeButton;`

Instance Variables or Local Variables

type name
↓ ↓
`UIButton * typeButton;`

Initializing Variables

Primitive Variables

name value
↓ ↓
`number = 0;`

Non-Primitive Variables (Objects)

name type allocate initialize
↓ ↓ ↓ ↓
`typeButton = [[UIButton alloc] init];`

Calling Methods

name method name
↓ ↓
`[typeButton resignFirstResponder];`

name method & argument name 1 method & argument name 2
↓ ↓ ↓
`[typeButton setTitle:@"Hello" forState:UIControlStateNormal];`
argument 1 argument 2

Control Flow

If Statement

condition
↓

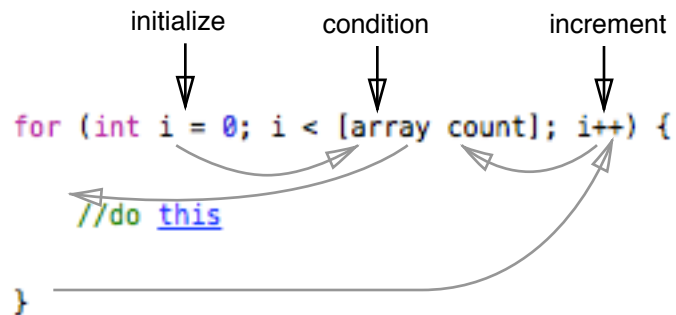
```
if (typeButton == nil) {  
    //this happens  
}
```

Switch Statement

For Loop

initialize condition increment
↓ ↓ ↓

```
for (int i = 0; i < [array count]; i++) {  
    //do this  
}
```



While Loop

Custom Class & Method

Header File

Public Methods and Properties Declared Here

```
class name(type)      subclass
    ↓                ↓
@interface NewViewController : UIViewController

return type           method and Argument name   argument   method and Argument name   argument
    ↓                ↓                ↓                ↓                ↓
-(void) newMethodsWithArgument:(int) argumentNum andArgument:(NSString *) argumentName;

argument type           argument type

@end
```

Implementation File

```
@interface NewViewController ()
@end

private methods and properties go here

@implementation NewViewController

return type           method and Argument name   argument   method and Argument name   argument
    ↓                ↓                ↓                ↓                ↓
-(void) newMethodsWithArgument:(int)argumentNum andArgument:(NSString *)argumentName {

    // do something here
    argument type           argument type

}

@end
```