**REVERSE A STRING**

NSString \*myString = "This is the string"

NSMutableString \*reversedString = [[NSMutableString alloc]init]; NSInteger charIndex = [myString length];

while (charIndex > 0)

{

charIndex--;

NSRange subStrRange = NSMakeRange(charIndex, 1); [reversedString appendString:[myString substringWithRange:subStrRange]];

}

NSLog(@"%@", reversedString); // outputs "ih"

**CREATE AN ARRAY**

NSArray \*array = @[@1, @2, @3, @4, @5, @6];

NSArray \*myArray = @[@"1", @"2", @"3", @"4", @"5"];

**REVERSE AN ARRAY**

NSArray \*reversed = [[anArray reverseObjectEnumerator] allObjects];

**ITERATE OVER AN ARRAY**

For OS X 10.4.x and previous:

int i;

for (i = 0; i < [myArray count]; i++)

{ id myArrayElement = [myArray objectAtIndex:i];

NSString \*value = (NSString \*)[appDelegate.bird\_arr objectAtIndex:rowClicked];

...do something useful with myArrayElement

}

For OS X 10.5.x (or iPhone) and beyond:

for (id myArrayElement in myArray) {

NSString \*value = (NSString \*)[appDelegate.bird\_arr objectAtIndex:rowClicked];

...do something useful with myArrayElement

}

**ITERATE OVER A DICTIONARY**

for(id key in \_recDict)

{

//myTable = [[TableObject alloc]init];

myTable = [\_recDict objectForKey:key];

myTable.tableColor = Clear;

//Call A Method

myTable = [myTable CreateTable:myTable

newTableFlag:false evalTableColor:true];

[self AddNewTable:myTable];

// do something with key and obj

NSLog(@"Table No: %@", myTable.tableNo);

}

**SORT AN ARRAY OF OBJECTS**

What I want to do seems pretty simple, but I can't find any answers on the web. I have an NSMutableArray of objects, let's say they are 'Person' objects. I want to sort the NSMutableArrayby Person.birthDate which is an NSDate.

NSSortDescriptor \*sortDescriptor;

sortDescriptor = [[NSSortDescriptor alloc] initWithKey:@"birthDate" ascending:YES];

NSArray \*sortDescriptors = [NSArray arrayWithObject:sortDescriptor]; NSArray \*sortedArray = [drinkDetails sortedArrayUsingDescriptors:sortDescriptors];

**SORT AN ARRAY OF STRINGS**

NSArray \*unsortedStrings = @[@"Verdana", @"MS San Serif", @"Times New Roman",@"Chalkduster",@"Impact"];

NSArray \*sortedStrings = [unsortedStrings sortedArrayUsingSelector:@selector(compare:)];

**CREATE A DELEGATE**

@protocol SomethingDelegate <NSObject>

@optional

-(void)something:(id)somethingdidFinishLoadingItem:(id)item;

-(void)something:(id)something didFailWithError:(NSError \*)error;

@end

@interface Something : NSObject

@property (nonatomic, weak) id <SomethingDelegate> delegate;

@end

@implementation Something {

struct {

unsigned int didFinishLoadingItem:1;

unsigned int didFailWithError:1;

}

delegateRespondsTo;

}

@synthesize delegate;

* (void)setDelegate:(id <JSSomethingDelegate>)aDelegate
* { if (delegate != aDelegate) { delegate = aDelegate; delegateRespondsTo.didFinishLoadingItem = [delegate respondsToSelector:@selector(something:didFinishLoadingItem:)]; delegateRespondsTo.didFailWithError = [delegate respondsToSelector:@selector(something:didFailWithError:)]; } } @end

**PROPERTIES**

@property (nonatomic, retain) [NSArray](http://developer.apple.com/documentation/Cocoa/Reference/Foundation/Classes/NSArray_Class/) \* sushiTypes;

@property (nonatomic, retain) [NSString](http://developer.apple.com/documentation/Cocoa/Reference/Foundation/Classes/NSString_Class/) \* lastSushiSelected;

**SINGLETON**

static NSMutableString \*\_tableNumber;

#pragma mark Singleton Methods

+ (id)sharedManager

{

static GlobalAccess \*sharedManager = nil;

static dispatch\_once\_t onceToken;

dispatch\_once(&onceToken, ^{

sharedManager = [[self alloc] init];

});

return sharedManager;

}

- (id)init {

if (self = [super init])

{

\_tableNumber = [[NSMutableString alloc] init];

\_btnDict = [[NSMutableDictionary alloc] init];

}

return self;

}

- (void)TableNumber:(NSString \*)tableNumber

{

\_tableNumber = [NSMutableString stringWithString: tableNumber];

}

- (NSString \*)TableNumber

{

return \_tableNumber;

}

**//TO CALL**

GlobalAccess \*sharedManager = [GlobalAccess sharedManager];

**INSTANTIATE A CLASS**

//Load the existing data in the database

DataAccess \*existingData = [[DataAccess alloc]init];

**NSURLCONNECTION**

// Create the request.

NSMutableURLRequest \*request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:@"http://google.com"]];

// Specify that it will be a POST request

request.HTTPMethod = @"POST";

// This is how we set header fields

[request setValue:@"application/xml; charset=utf-8" forHTTPHeaderField:@"Content-Type"];

// Convert your data and set your request's HTTPBody property

NSString \*stringData = @"some data";

NSData \*requestBodyData = [stringData dataUsingEncoding:NSUTF8StringEncoding];

request.HTTPBody = requestBodyData;

// Create url connection and fire request

NSURLConnection \*conn = [[NSURLConnection alloc] initWithRequest:request delegate:self];

**GCD**

dispatch\_async(dispatch\_get\_global\_queue(DISPATCH\_QUEUE\_PRIORITY\_HIGH, 0), ^{ *// 1* UIImage \*overlayImage = [self faceOverlayImageFromImage:\_image]; dispatch\_async(dispatch\_get\_main\_queue(), ^{ *// 2*

[self fadeInNewImage:overlayImage]; *// 3* }); });