

CS 696 Mobile Phone Application Development
Fall Semester, 2010
Doc 19 Network part 1
Nov 4, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

URL Loading System Programming Guide, Apple documentation, <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/URLLoadingSystem/URLLoadingSystem.html>

Event-Driven XML Programming Guide, Apple documentation, <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/XMLParsing/Articles/HandlingElements.html>

iPhone Programming: The Big Nerd Ranch Guide, Conway & Hillegass, Big Nerd Ranch, 2010, Chapter 21 Web Services

More iPhone 3 Development: Tackling iPhone SDK3, Mark & LaMarche, Apress, 2009, Chapter 10 Working with Data from the Web

Accessing Data from network

Web pages (UIWebView)

URL

Web services

GameKit & Bluetooth

Bonjour

Socket Programming

Displaying Web

UIWebView

Displays page
Does most of the work

NSURL

URL to load

NSURLRequest

Caching
timeout
Http request properties

UIWebViewDelegate

Information about loading of page



Roger Whitney's
Courses

[To Roger
Whitney's
Home Page](#)

San Diego
State

University --
This page last
updated 15-
Aug-2010

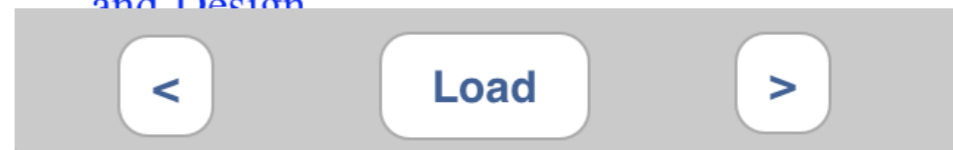
If you wish to know more about my classes
the following courses contain all class notes,
lectures, assignments, syllabus and exams.

Fall Semester

Spring Semester

2010-2011

[CS535 OO
Programming
and Design](#)





View Set up



Roger Whitney's
Courses

[To Roger
Whitney's
Home Page](#)

San Diego
State

University --
This page last
updated 15-
Aug-2010

```
@interface WebViewViewController :  
    UIViewController <UIWebViewDelegate>{
```

```
}
```

```
IBOutlet UIWebView *webpage;
```

```
IBOutlet UIButton *forwardButton;
```

```
IBOutlet UIButton *backButton;
```

If you wish to know more about my classes
the following courses contain all class notes,
lectures, assignments, syllabus and exams.

Fall Semester

Spring Semester

2010-2011

[CS535 OO
Programming
and Design](#)



Load



```
- (IBAction) back;
```

```
- (IBAction) showPage;
```

```
- (IBAction) forward;
```

```
@end
```

Starting

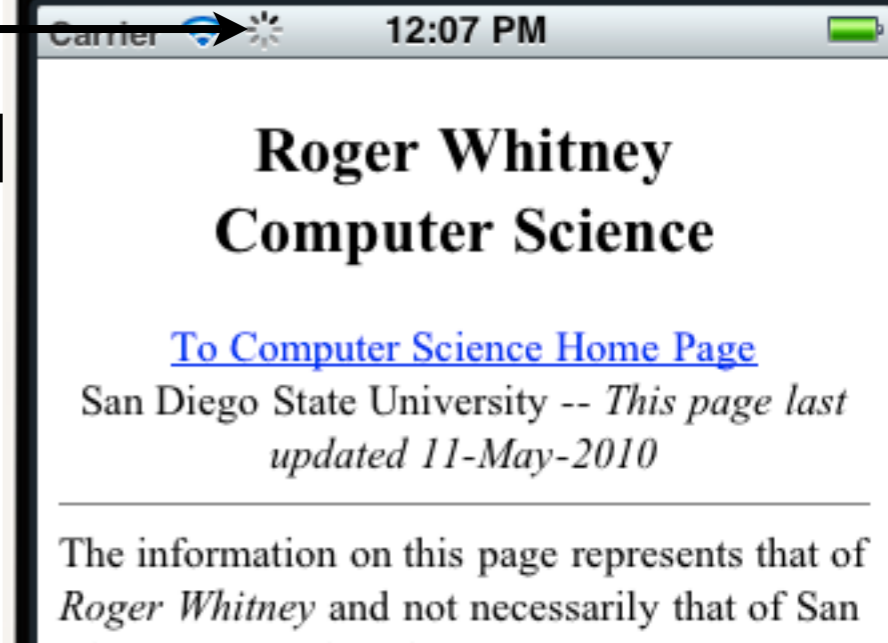
```
- (void)viewDidLoad {
    [super viewDidLoad];
    backButton.hidden = YES;
    forwardButton.hidden = YES;
}

- (IBAction) showPage {
    webpage.delegate = self;
    [webpage loadRequest:[NSURLRequest requestWithURL:
        [NSURL URLWithString:@"http://www.eli.sdsu.edu/"]]];
}
```

UIWebViewDelegate Methods

Showing network traffic

- (void)webViewDidStartLoad:(UIWebView *)webView {
 [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;
}
- (void)webViewDidFinishLoad:(UIWebView *)webView {
 [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;
 [self hideShowNavigationButtons];
}



UIWebViewDelegate Method

Loading a string

```
- (void)webView:(UIWebView *)webView didFailLoadWithError:(NSError *)error {
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;

    // report the error inside the webview
    NSString* errorString = [NSString stringWithFormat:
        @"<html><center><font size=+6 color='red'>An error occurred:<br>
        %@</font></center></html>",
        error.localizedDescription];
    [webpage loadHTMLString:errorString baseURL:nil];
    [self hideShowNavigationButtons];
}
```


Navigation

- (IBAction) back {
 [webpage goBack];
 [self hideShowNavigationButtons];
}
- (IBAction) forward {
 [webpage goForward];
 [self hideShowNavigationButtons];
}
- (void) hideShowNavigationButtons {
 backButton.hidden = !webpage.canGoBack;
 forwardButton.hidden = !webpage.canGoForward;
}

Dealloc

```
- (void)dealloc {  
    [super dealloc];  
    webpage.delegate = nil;  
    [webpage release];  
}
```

URL

Supports

ftp://
http://
https://
file:///

proxy servers
Authentication
Cookies

Synchronous
Asynchronous

Major Classes

NSURL

URL to load

NSData

Store data until done reading

NSURLRequest

Caching

timeout

Http request properties

NSURLConnection

Handles connection with server

NSURLConnection delegate methods

Handle asynchronous reading of data

Asynchronous Examp

Connect to Server

Read Data asynchronously

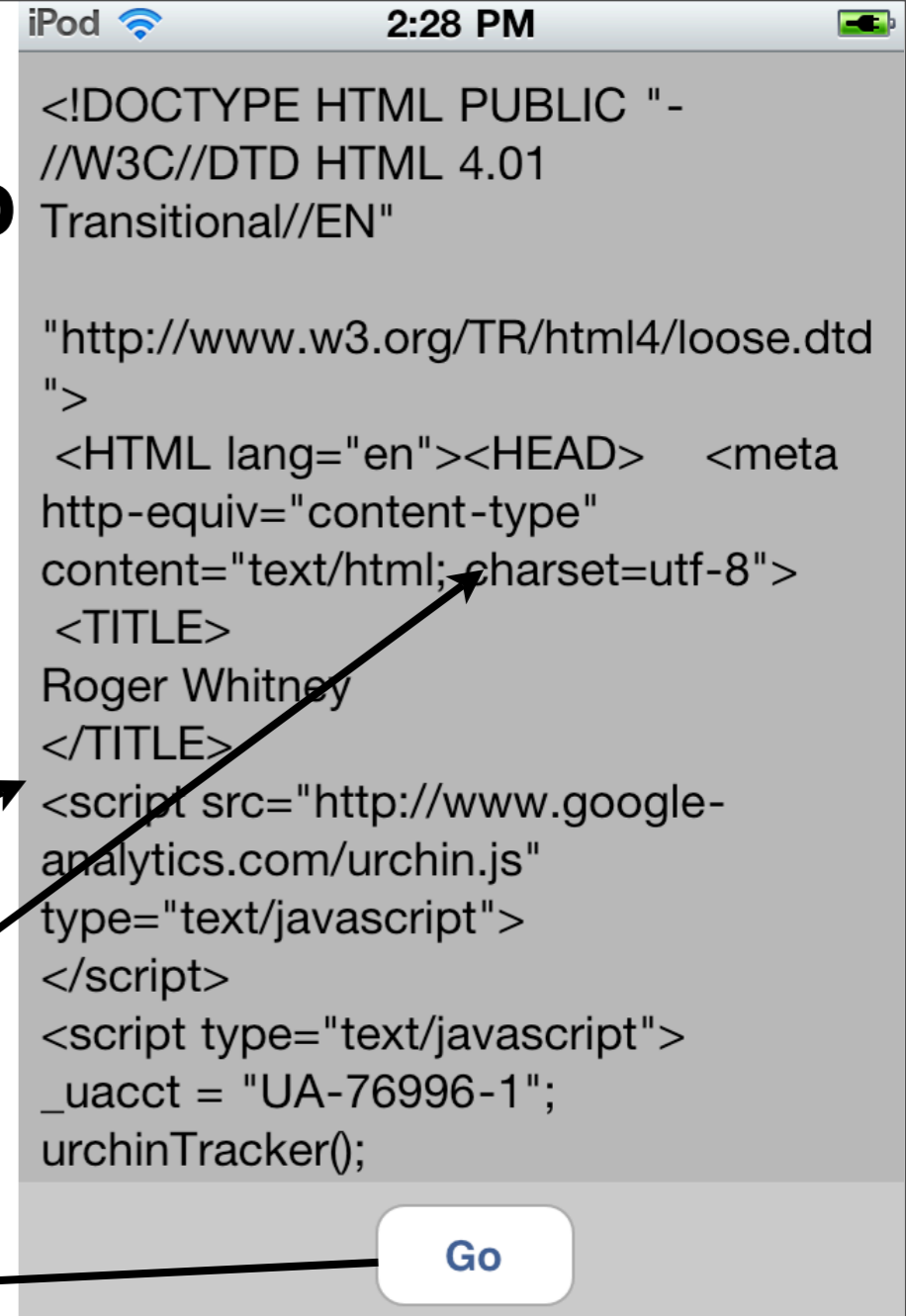
Display resulting data

```
iPod 2:28 PM
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML lang="en"><HEAD> <meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>
Roger Whitney
</TITLE>
<script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
</script>
<script type="text/javascript">
_uacct = "UA-76996-1";
urchinTracker();

```

Go

Asynchronous Examp



```
@interface URLConnectionExampleViewController :  
UIViewController {  
}  
property - NSMutableData *receivedData;  
property - IBOutlet UITextView * textview;  
property - IBOutlet UIActivityIndicatorView *spinner;  
  
- (IBAction) fetchUrl;  
@end
```

Starting the connection

```
- (IBAction) fetchUrl {
    [spinner startAnimating];
    NSURLRequest *request = [[NSURLRequest alloc] initWithURL:
        [NSURL URLWithString:@"http://www.eli.sdsu.edu/"]];
    NSURLConnection *connection = [[NSURLConnection alloc] initWithRequest:request
        delegate:self];

    if (connection) {
        NSMutableData *data = [[NSMutableData alloc] init];
        self.receivedData=data;
        [data release];
    }
    else {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Error"
            message:@"Error connecting to remote server"
            delegate:self
            cancelButtonTitle:@"Ok"
            otherButtonTitles:nil];

        [alert show];
        [alert release];
    }
    [request release];
}
```

Reading the data

```
- (void)connection:(NSURLConnection *)connection  
    didReceiveResponse:(NSURLResponse *)response {  
    [receivedData setLength:0];  
}
```

```
- (void)connection:(NSURLConnection *)connection  
    didReceiveData:(NSData *)data {  
    [receivedData appendData:data];  
}
```


When done reading data

```
- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    NSString *payloadAsString = [[NSString alloc] initWithData:receivedData
        encoding:NSUTF8StringEncoding];
    textView.text = payloadAsString;
    [payloadAsString release];

    [connection release];
    self.receivedData = nil;
    [spinner stopAnimating];
}
```

When there is an error

```
- (void)connection:(NSURLConnection *)connection  
  didFailWithError:(NSError *)error {  
  [connection release];  
  self.receivedData = nil;
```

```
  UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Error"  
                message:[NSString stringWithFormat:@"Connection failed!  
                %@", [error localizedDescription]]  
                delegate:self  
                cancelButtonTitle:@"Ok"  
                otherButtonTitles:nil];  
  
  [alert show];  
  [alert release];  
  [spinner stopAnimating];  
}
```

Nice example but

Displaying html as text not very useful

html is not useful when we want data

Web Services & Rest

Basic Idea

URL - request for data

Request sent to server

Response contains data requested

SOAP

Complex

Rest

Simpler

Returns XML or JSON

Geocoding Example

<http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false>

Returns geocoding data for given address

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<GeocodeResponse>
  <status>OK</status>
  <result>
    <type>street_address</type>
    <formatted_address>1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA</formatted_address>
    <address_component>
      <long_name>1600</long_name>
      <short_name>1600</short_name>
      <type>street_number</type>
    </address_component>
    ...
  </address_component>
  <geometry>
    <location>
      <lat>37.4227820</lat>
      <lng>-122.0850990</lng>
    </location>
    <location_type>ROOFTOP</location_type>
    <viewport>
      <southwest>
        <lat>37.4196344</lat>
        <lng>-122.0882466</lng>
      </southwest>
      <northeast>
```

XML & Parsing

NSXMLParser provided by Apple

<http://weblog.bignerdranch.com/?p=48>

Event-Driven XML Programming Guide

libxml2

Preferred by many

C library

Have to download & install

<http://www.json.org/>

JavaScript Object Notation

data-interchange format

rfc 4627

Maps to/from strings

null

true, false

number

string

array

objects

Implementations in

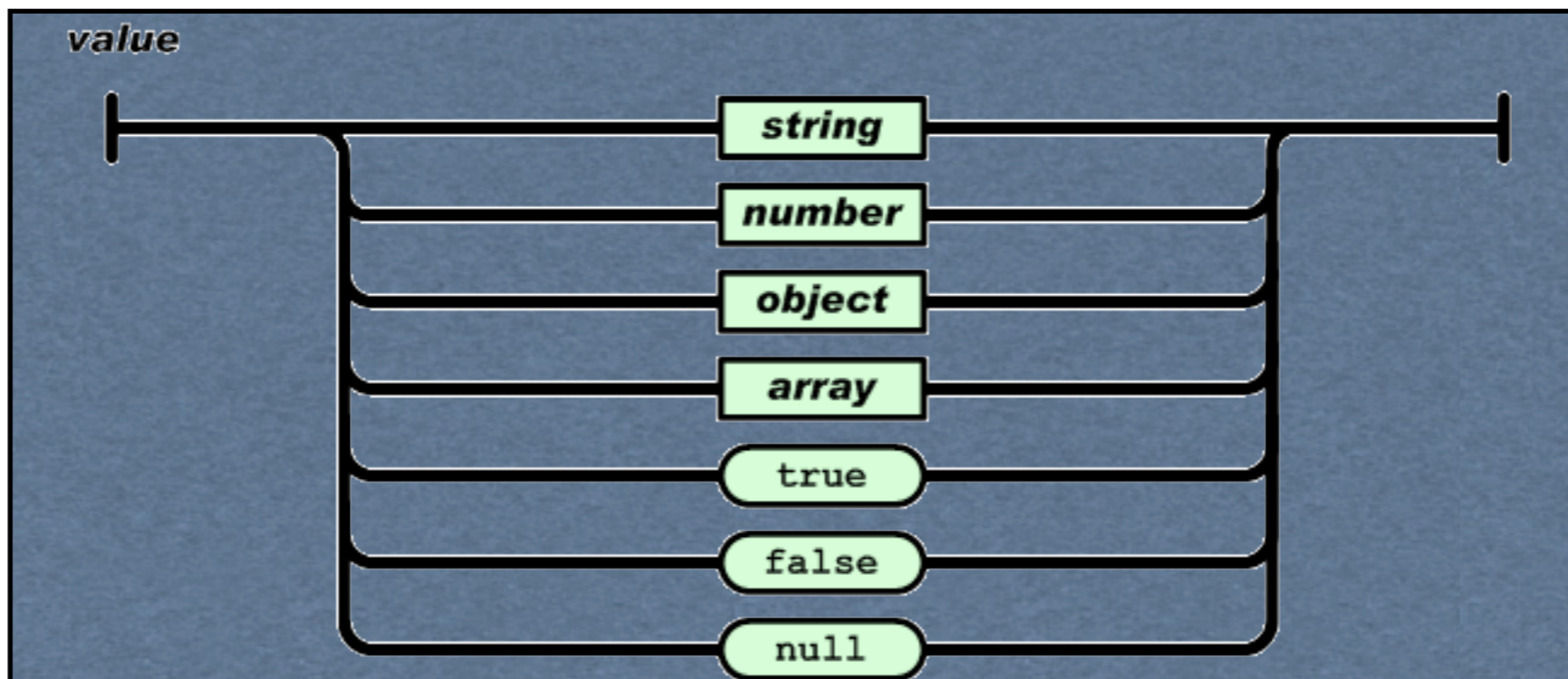
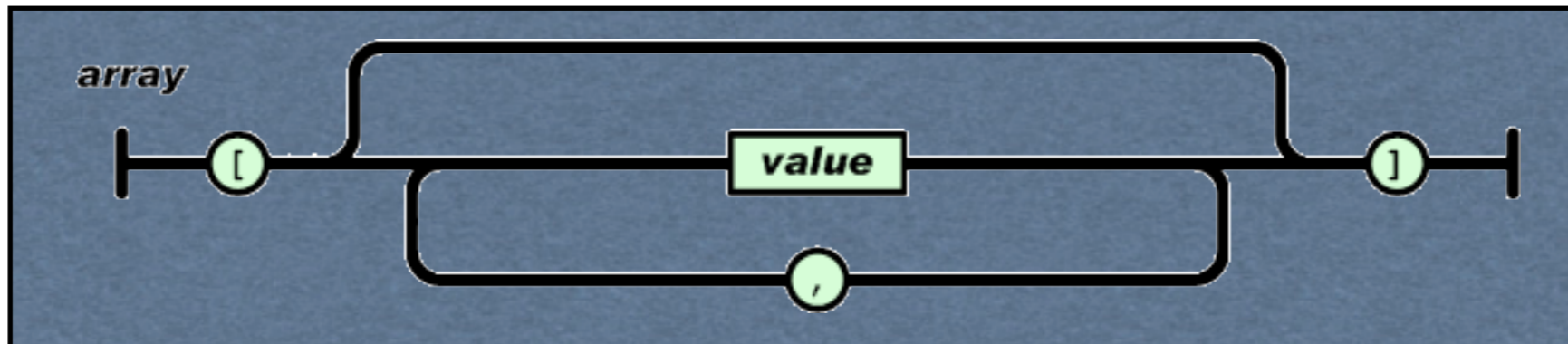
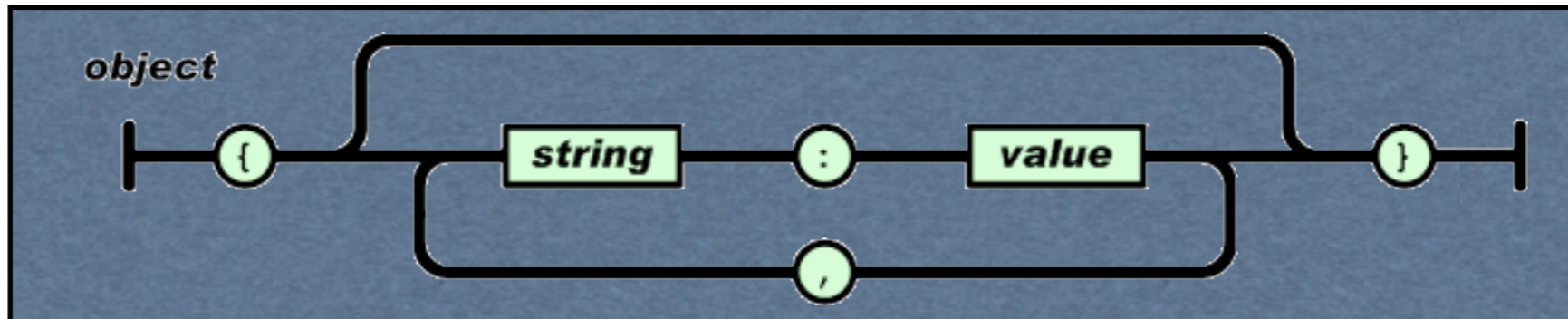
C, C++, C#, D, E, Java, Objective C

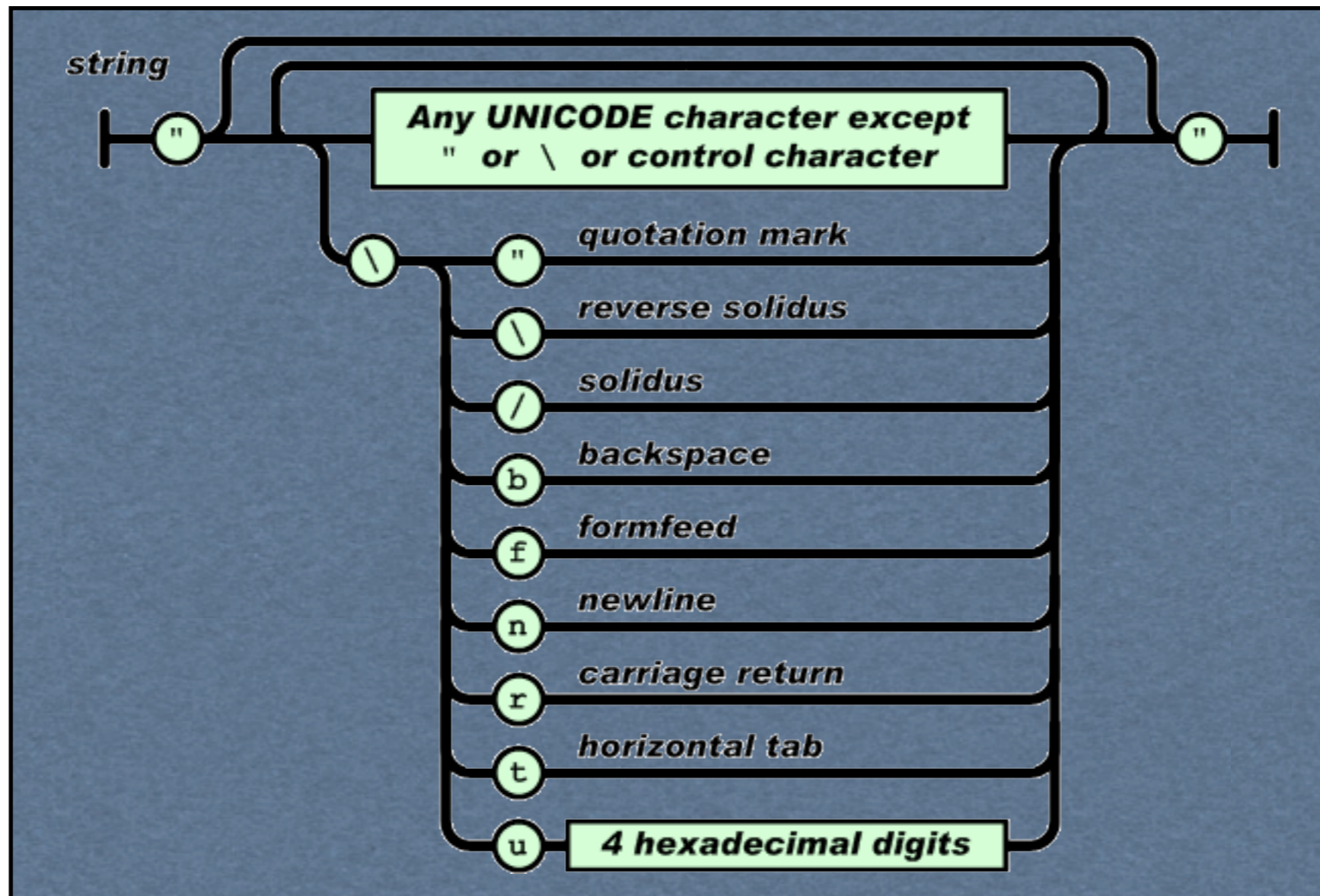
Cold Fusion, Delphi, Erlang, Haskell

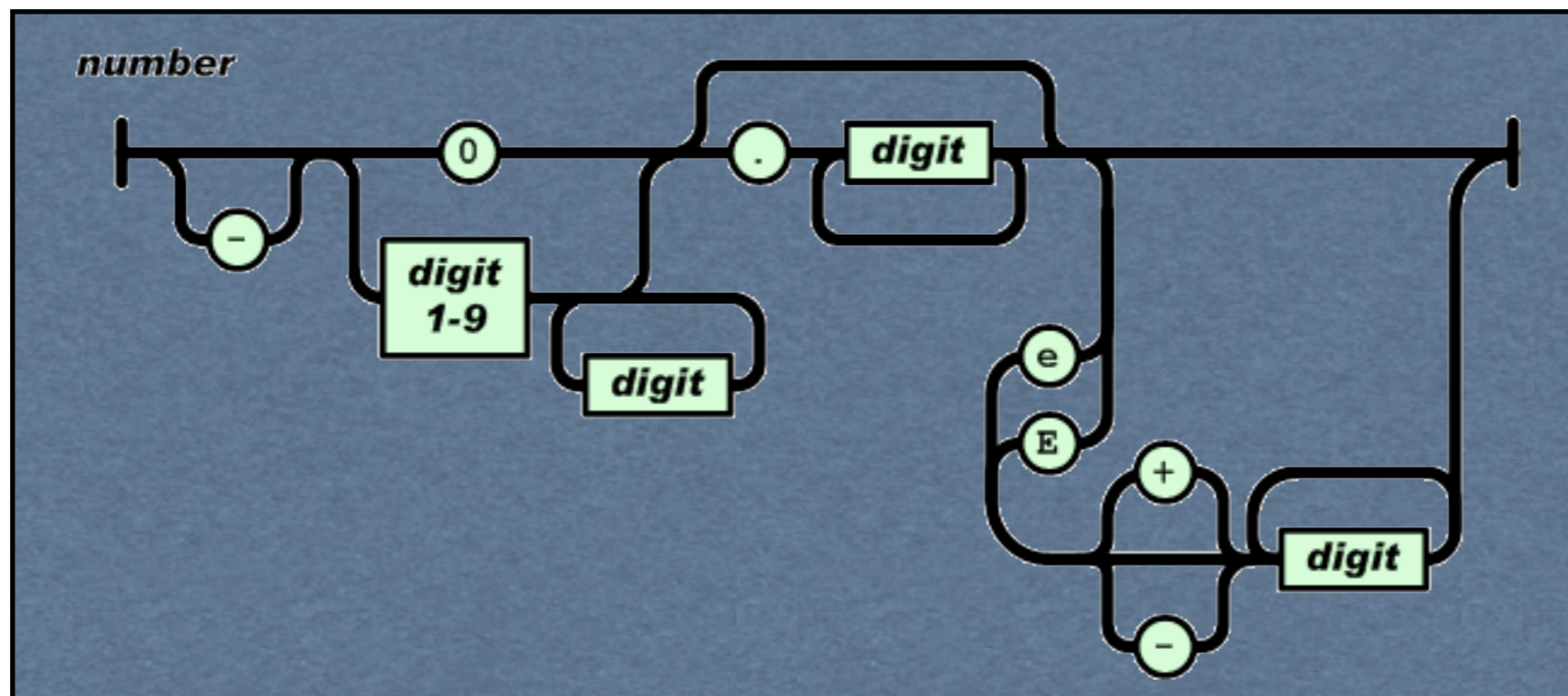
JavaScript, Lisp, LotusScript, Perl,

PHP, Pike, Prolog, Python, Ruby, Smalltalk

Source: <http://www.json.org/>







```
[ 23, "cat", 65.324]
```

```
{ "size" : 12, "color" : "red" }
```

```
{ "sizes" : [10, 12, 14], "colors" : ["red", "yellow", "blue"], "price" : 12.99 }
```

JSON & iPhone

Framework

<http://code.google.com/p/json-framework/>

Tutorial

<http://mobileorchard.com/tutorial-json-over-http-on-the-iphone/>

Converting JSON to objects

```
NSArray * result = [@"[ 23, 92, 65.324]" JSONValue];
```

```
NSDictionary * secondResult = [@"{ \"size\" : 12, \"color\" : \"red\" }" JSONValue]
```

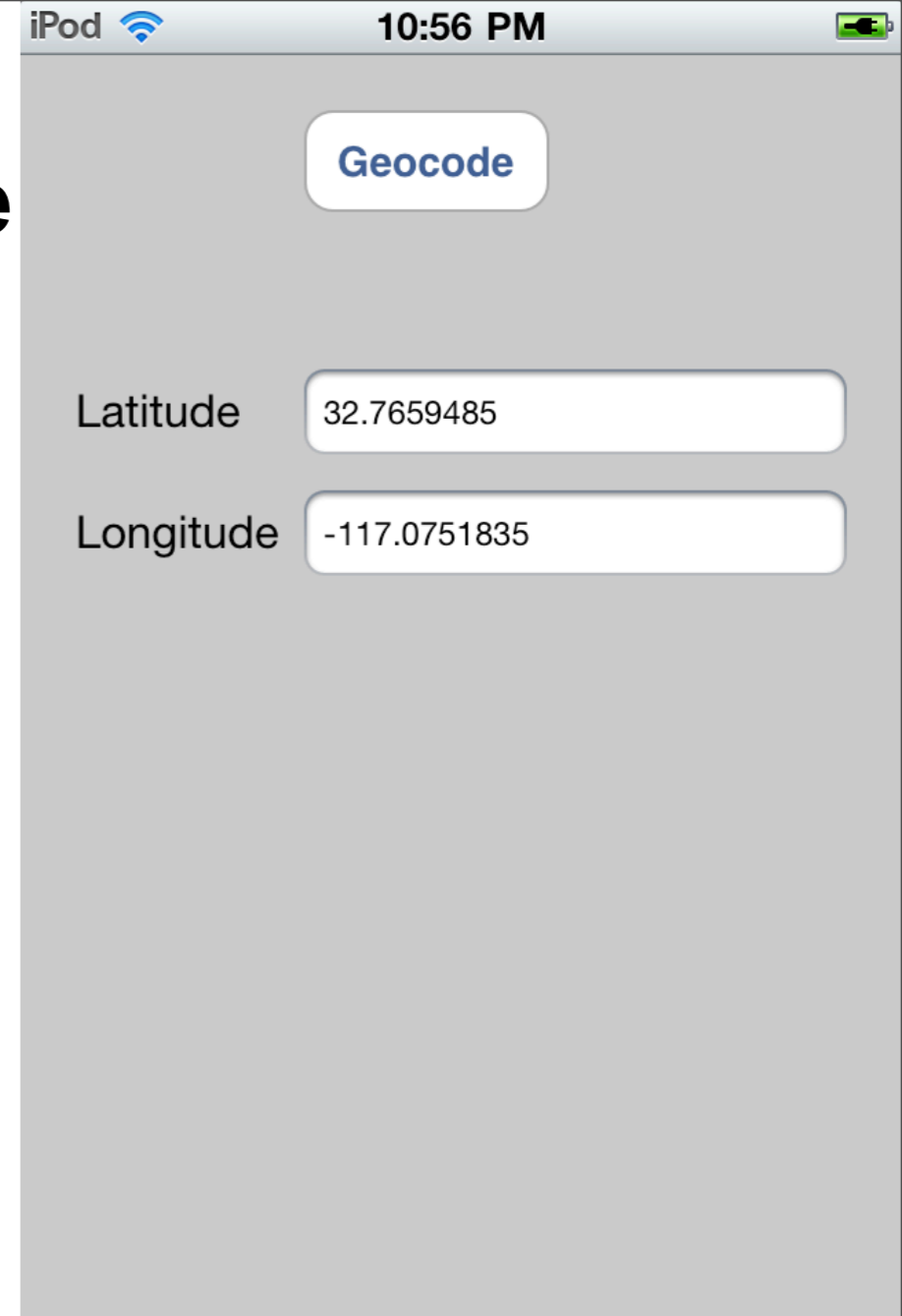
```
NSError *error;
```

```
SBJSON *json = [[SBJSON new] autorelease];
```

```
NSArray * result = [json objectWithString: @"[ 23, 92, 65.324]" error:&error];
```

Geocoding Example

Use Google Geocoding service to find latitude & longitude of SDSU



Request & Response

<http://maps.googleapis.com/maps/api/geocode/json?address=5500+Campanile+Drive,+San+Diego,+CA&sensor=true>

```
{
  "status": "OK",
  "results": [
    {
      "types": [ "street_address" ],
      "formatted_address": "5500 Campanile Way, San Diego, CA 92115, USA",
      "address_components": [
        {
          "long_name": "5500",
          "short_name": "5500",
          "types": [ "street_number" ]
        },
        {
          "long_name": "Campanile Way",
          "short_name": "Campanile Way",
          "types": [ "route" ]
        }
      ], ...
    },
    {
      "long_name": "92115",
      "short_name": "92115",
      "types": [ "postal_code" ]
    }
  ],
  "geometry": {
    "location": {
      "lat": 32.7659485,
      "lng": -117.0751835
    },
    "location_type": "ROOFTOP",
    "viewport": {
      "southwest": {
        "lat": 32.7628009,
```


Making the Request

```
- (IBAction) geocode {
    [spinner startAnimating];
    NSURLRequest *request = [[NSURLRequest alloc] initWithURL:[NSURL URLWithString:@"http://maps.googleapis.com/
maps/api/geocode/json?address=5500+Campanile+Drive, San+Diego, +CA&sensor=true"]];
    NSURLConnection *connection = [[NSURLConnection alloc] initWithRequest:request
                                                                    delegate:self];

    if (connection) {
        NSMutableData *data = [[NSMutableData alloc] init];
        self.receivedData=data;
        [data release];
    }
    else {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Error"
                                                            message:@"Error connecting to remote server"
                                                            delegate:self
                                                            cancelButtonTitle:@"Ok"
                                                            otherButtonTitles:nil];

        [alert show];
        [alert release];
    }
    [request release];
}
```

Reading Data

- (void)connection:(NSURLConnection *)connection
 didReceiveResponse:(NSURLResponse *)response {
 [receivedData setLength:0];
 }
- (void)connection:(NSURLConnection *)connection
 didReceiveData:(NSData *)data {
 [receivedData appendData:data];
 }

Extracting Latitude & Longitude

```
- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    NSString *payloadAsString = [[NSString alloc] initWithData:receivedData
encoding:NSUTF8StringEncoding];
    NSDictionary * answer = [payloadAsString JSONValue];

    NSArray * results = [answer objectForKey:@"results"];
    NSDictionary * result = [results objectAtIndex:0];
    NSDictionary * geometry = [result objectForKey:@"geometry"];
    NSDictionary * location = [geometry objectForKey:@"location"];

    NSNumber * latitude = [location objectForKey:@"lat"];
    NSNumber * longitude = [location objectForKey:@"lng"];
    longitudeView.text = [longitude description];
    latitudeView.text = [latitude description];
    [payloadAsString release];

    [connection release];
    self.receivedData = nil;
    [spinner stopAnimating];
}
```

XML Example

Sending & receiving request the same

Parsing data different

NSXMLParser (SAX parser)

NSXMLParserDelegate

NSXMLParserDelegate

– parser:didStartElement:namespaceURI:qualifiedName:attributes:

Called when start parsing an element

– parser:foundCharacters:

Called when find characters in element

Can be called multiple times in same element

– parser:didEndElement:namespaceURI:qualifiedName:

Called when ending an element

At End of Connection

```
- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    NSXMLParser * parser = [[NSXMLParser alloc] initWithData:receivedData];
    [parser setDelegate:self];
    [parser parse];                // parser blocks until done
    [parser release];
    [connection release];
    self.receivedData = nil;
    [spinner stopAnimating];
}
```

Delegate method

```
- (void)parser:(NSXMLParser *)parser
  didStartElement:(NSString *)elementName
  namespaceURI:(NSString *)namespaceURI
  qualifiedName:(NSString *)qualifiedName
  attributes:(NSDictionary *)attributeDict {

    if ([elementName isEqual:@"location"]) {
        inLocationElement = YES;
        elementString = [[NSMutableString alloc] init];
    }
}
```

The characters

```
- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string {  
    [elementString appendString:string];  
}
```


Getting the latitude & longitude

```
- (void)parser:(NSXMLParser *)parser
didEndElement:(NSString *)elementName
namespaceURI:(NSString *)namespaceURI
qualifiedName:(NSString *)qName {

    if (inLocationElement & [elementName isEqual:@"lat"]) {
        latitudeView.text = elementString;
        [elementString release];
        elementString = [[NSMutableString alloc] init];
    }
    if (inLocationElement & [elementName isEqual:@"lng"]) {
        longitudeView.text = elementString;
        [elementString release];
        elementString = [[NSMutableString alloc] init];
    }
    if ([elementName isEqual:@"location"]) {
        inLocationElement = NO;
        [elementString release];
        elementString = nil;
    }
}
```