

iWebKit⁵ | *user guide*

*<http://www.iwebkit.net>
<http://news.iwebkit.net>
<http://demo.iwebkit.net>
<http://community.iwebkit.net>*

Christopher Plieger & Johan Van Wilsum

Start developing your own iPhone Web application right here! This guide will help you start a project and discover how to work with the framework from start to finish. You will learn what elements are available and how to customize and work with them.

Table Of Contents

How to get started with iWebKit.	3	Integration	12
Some basic principles	3	Introduction	12
Some elementary HTML knowledge	3	Detecting phone numbers	12
The top bar	4	Set up an e-mail	12
Getting started	4	Call a number/ add to contact	12
Adding the top bar and options	4	Open a YouTube video	12
Navigation: Arrows	4	Send an SMS to number	12
Navigation: Buttons	4	Open Google maps at a specific location	13
Navigation: trio/duo buttons	5	Open iTunes or Appstore and show a specific item	13
The content.	6	Embed an RSS feed	13
Introduction	6	Important notes and information	13
Gray titles	6		
Introduction to Page items	6		
Page items: Textboxes	6		
Page items: Menus	7		
Page items: iTunes classic list view	7		
Other special elements	9		
The list views.	9		
Introduction	9		
The Classic List	9		
The Appstore list	10		
The iTunes Music List	10		
The iPod list	11		
Full Screen & its options	11		
Introduction	11		
Changing the status Bar color	11		
Add a custom icon to the home screen	11		
Add a custom loading screen to your site	12		
Outgoing links	12		

How to get started with iWebKit.

Some basic principles

Hi and thanks for downloading iWebKit. Some of you might be surprised by the contents of this package. Indeed there is no "executable" file included to launch a "program". Well let me explain that for HTML newcomers.

Basically, it's you that will write your own website combining elements and text to form your own awesome iPhone Website! Now, I know what you're thinking, "pfff come on, what is this BS... let's just delete this junk" but please hold on! It is easier than it seems.

You need only one simple skill to succeed in making your website and that is: copy and paste. Seriously, yes you just need to copy and paste text, it's as simple as that! I can see that you seem happier! That's the spirit!

Throughout this document I will explain what every little bit of code does and where you should paste it. I will also show you what you can edit to add your own text or images.

Let's start with an example: `this is my first code`

You don't need to know what a `` is; all you will need to do is copy that element to the right spot and then edit the gray text. From now on all code will be dark orange; all white text is, most of the time, what visually appears on your site (text or images for example) and you will need to edit it.

Some elementary HTML knowledge

Alright now that you are all enthusiastic and ready to start let's first take a break to learn some basic html knowledge first. You don't need to remember all of this but just know that it's here if you are lost!

- ✓ The head of an HTML document is just like the header of a memo, the head of an HTML document contains special information, like its title for example. The head of a document is demarcated by `<head>` and `</head>` respectively. Once you have built your webpage you should fill in the title, description and keyword tags all included in the head.

- ✓ The body of an HTML document is filled with everything that is visible to the user. It is delimited by the `<body>` and `</body>` tags. All your text, images, and links will be in the body of the document.
- ✓ Bold and italic texts are both created by special tags. To make text bold just put the `text` tags around it. For italic text it's the same principle only you use `text` tags
- ✓ Creating links to other pages or sites is rather simple. Just put your link in any `href="link"` attribute included in an `<a>`/`` (you will see them in the elements). Just replace link with any url like `http://mysite.com/mypage.html`. Example: ``
- ✓ Inserting an image in your page is also easy and just a matter of adding one simple tag. Paste this tag in your textbox (or any other element) to add an image to your page: ``. Normally the image is automatically scaled to fit the width of the element. From time to time you will see `style="background-image: url('myimage.png')"` throughout this guide. Just replace the image with your own. It will have a specific function in the element.
- ✓ Center elements the right way, it isn't hard either. If you want to center an image or a paragraph, `<p>`/`</p>`, for example just add `class="center"` to that first element. A little example: `` another example: `<p class="center">my text</p>`
- ✓ Look out for Special characters because they can mess up your code. For example, you cannot use "<" or ">" in your code. You need to replace them with `<`; and `>`; respectively. Here is a list of some of the most used items and their equivalent in code:

<http://www.tedmontgomery.com/tutorial/HTMLchrc.html>

The top bar

Getting started

Ok right now you just know you have to copy and paste but now we are seriously going to begin designing a page. First off, open up the "index.html" file in the "framework" folder. Ok not in your browser but in a text editor like notepad for windows for example. Now concentrate on the body of the page.

Adding the top bar and options

If it isn't in your page yet, just paste this code right under the `<body>` tag:

✓ `<div id="topbar"></div>`

If you open up your page in safari or Google chrome you will notice the blue header. See easy right? Next you have the option to make it **black or transparent** by adding either `class="black"` or `class="transparent"` to the `<div>`

✓ `<div id="topbar" class="black"></div>`

✓ `<div id="topbar" class="transparent"></div>`

Ok now let's add a **title** to it. You can mainly use the title on pages that only have 1 or 2 navigation buttons, for aesthetic and practical reasons. Just add `<div id="title">Title</div>` between the `<div>` and the final `</div>` like this:

✓ `<div id="topbar"><div id="title">Title</div></div>`

Check your browser from time to time to check the result and see if you made any errors.

Navigation: Arrows

There are 3 different types of navigation available for iWebKit: **Arrows, buttons and trio/duobuttons**. Let's start with arrows. You have a choice between left and right arrows dependent on the page you are linking to. Simply paste one of these codes in the top bar `<div>` to create a **left or right pointing arrow**:

✓ `<div id="leftnav">text</div>`

✓ `<div id="rightnav">text</div>`

Just a quick note for people who have no idea what to put in here instead of "page.html", stop skipping pages and just read what I wrote about the `<a>` element!

You can also create a **"home" arrow** with a picture of a house in it like in the demo. Just add this image, instead of text, to the arrow:

``

✓ `<div id="leftnav"></div>`

Overall top bar code until now: Title, left arrow with home image, right arrow.

```
<div id="topbar">
  <div id="title">Title</div>
  <div id="leftnav">
    <a href="page.html" >
      
    </a>
  </div>
  <div id="rightnav">
    <a href="page.html" >text</a>
  </div>
</div>
```

Also included is a **breadcrumb system** that makes it possible to have multiple left or right arrows glued together. To achieve this effect simply add another `text` to you left or right arrow div like this:

✓ `<div id="leftnav">AB</div>`

✓ `<div id="rightnav">AB</div>`

Navigation: Buttons

You can add **buttons** just like arrows except that you cannot use the breadcrumb feature. The code you need to add is only slightly different for both left and right buttons and you must paste it in the top bar `<div>` like with

the navigation arrows:

- ✓ `<div id="leftbutton">Text or Image</div>`
- ✓ `<div id="rightbutton">Text or image</div>`

You can also insert **blue buttons** that you might want to use for "special actions" like an external link or a login button. To use it you need to slightly change the button code:

- ✓ `<div id="blueleftbutton">Text or Image</div>`
- ✓ `<div id="bluerightbutton">Text or image</div>`

Navigation: trio/duo buttons

There are **2 types of trio buttons: The ones nested inside of the top bar and those under it.** Take a look at the demo included to see what I mean.

Let's start by the one **included in the top bar.** You have the option to pick a three or two button navigation block. As with all the other navigation items, just paste the following codes in the top bar `<div>`:

- ✓ `<div id="duoselectionbuttons">TextText</div>`
- ✓ `<div id="triselectionbuttons">ATextBTextCText</div>`

As you have surely guessed the first code creates a two-button element whereas the second one makes a three button component.

Ok now let's deal with **the trio buttons under the top bar.** There is a huge difference between this code and the last one because you do not place it in the top bar `<div>` but under it! This is very important to remember!

These are the codes for both the two and three button component:

- ✓ `<div id="duobutton"><div class="links">TextText</div></div>`
- ✓ `<div id="tributton"><div class="links">ATextBTextCText</div></div>`

So a page with both tri buttons would look like this, notice that the "tributton" is placed under the topbar!

```
<div id="topbar">
  <div id="triselectionbuttons">
    <a href="A.html">AText</a><a href="B.html">BText</a></div>
  <a href="C.html">CText</a>
</div>
<div id="tributton"><div class="links">
  <a href="A.html">AText</a><a href="B.html">BText</a><a href="C.html">CText</a>
</div></div>
```

Now there is one **important** thing left. You need to mark the page that is active by changing the color of the corresponding button. Sounds confusing? Just take a look at the demo and you will see that one button is always in a different color to mark the page you're on.

This is actually very simple to do. Simply add: `id="pressed"` to the active `<a>` element:

- ✓ `<div id="triselectionbuttons">ATextBTextCText</div>`

In this case you are on the `A.html` page and the first `<a>` will be different. This works with "tributton's " and "triselectionbuttons".

The content.

Introduction

You have to admit that that first part was a piece of cake right? Simply copy and pasting! Well now we are getting into some more serious business. Just joking, this is still as easy! We are now going to talk about adding menu items, textboxes, etc... Everything that makes up your page! For this part of your site you will have to work inside of the `<div id="content"></div>` tags

Gray titles

Ok let's get started! Gray titles are those directly on the background of the page, you know often above menu items or textboxes. To create one simple paste the following code right in your content `<div>`:

✓ `my title`

Final result: `<div id="content">text</div>`
That was easy right?

Introduction to Page items

Here it comes, the most important element of iWebKit, the "**pageitem**". This component is essentially the outline of menu items, textboxes and iTunes lists. You simply need to "**fill it up**"!

Here is the code you need to add inside the content `<div>`:

✓ `<ul class="pageitem">`

Each page item will create a new group of elements you can organize the way you want to.

Here are the elements you can add together and rearrange:

Page items: Textboxes

A **textbox** is simply a blank item that you can type in, but I'm sure you guessed that already. As you might have understood this element must be placed **inside a pageitem** ``. The code is very simple, check it out:

✓ `<li class="textbox">`

So that's the element you type in but hold on I got some nice tools for you to work with. First of let's talk about paragraphs. **Paragraphs** are used to create a block of text and are needed inside a textbox to give your text the right placement. Simply add `<p>`finally some real text`</p>` inside a textbox.

Oh boy a `<p>` inside an `` inside an `` inside a `<div>` inside the `<body>`. But I'm certain for you it is as clear as a glass of boiling water. You are excited to continue aren't you?! Don't worry I'll recap in a sec.

Ok then, another element for in the textbox then; a blue title. With blue titles you can simply separate your huge chunk of text in smaller bits. You place them inside of a textbox and before a paragraph. This is the code you use:

✓ `my blue title`

Let's recapitulate; here is a gray title & textbox with a blue title and a paragraph:

```
<div id="content">
  <span class="graytitle">Title</span>
  <ul class="pageitem">
    <li class="textbox">
      <span class="header">Blue title</span>
      <p>My text</p>
    </li>
  </ul>
</div>
```

So that was the first element you can add to a page item, let's continue and talk about menu items now.

Page items: Menus

Menu items are the foundation of **navigation** on your website. It's thanks to them that your users will be able to quickly find what they are looking for. You place them in the same spot as a textbox, right inside the pageitem ``. You can add **multiple menu items** in the same page item and even **mix them with textboxes**.

Now that you know how to use iwebkit, more or less, let's speed things up a little. The code you have to use is the following:

```
✓ <li class="menu"><a href="page.html"><span class="name">Name</span><span class="arrow"></span></a></li>
```

This creates a simple menu item with a name and an arrow on the right side of it. You have probably seen those little icons in them, either on the included demo or the iPhone itself, well we have those too. We even got a whole library included in the "thumbs" folder! To add icons just use image tags and put it straight in the menu ``. Just use this code:

```
✓ 
```

Like icons you can also add a comment field next to the arrow on the right using this code:

```
✓ <span class="comment">Comment</span>
```

Ok let's visualize this alright? A menu item with a small icon and a comment.

```
<ul class="pageitem">
  <li class="menu">
    <a href="page.html">
      
      <span class="name">Item Name</span>
      <span class="comment">Comment</span>
      <span class="arrow"></span>
    </a>
  </li>
</ul>
```

Page items: iTunes classic list view

Alright, this is the only **list item** that you can embed inside of a page item and it can be really useful in many situations especially if you want to embed an image of your cat jumping in a box or fighting a bear. The code for this one is a little bit bigger and I won't be adding bits and pieces progressively this time.

I mean it is easy, right?! Just put this tag inside a page item like with textboxes or menu items:

```
✓ <li class="store"><a href="page.html"><span class="image" style="background-image: url('image.png')"></span><span class="comment">comment</span><span class="name">Name</span><span class="stars5"></span><span class="starcomment">5 ratings</span><span class="arrow"></span></a></li>
```

So as you see there are multiple things you can customize. First of the link of course as in every element, next the **image on the left side of the element**, It has to be at least a 90x90px square, but if it's bigger it will automatically resize to fit the background!

Next you can add a bit of **text to name your link and add a comment**. The comment is actually on top of the name explaining the order that I put the elements in.

Now the **star/rating system**; this is also easy to customize. By changing "stars5" to anything between **stars0** and **stars5** you can change the star rating of that particular element. Finally the star comment just displays whatever you want; in the iTunes store it displays the number of ratings for example.

You can of course remove any of these tags by just deleting the corresponding `` tag.

Page items: Form elements

The form elements included in iWebKit 5 are brand new and super easy to use. Just one quick note before we continue, I just provide the graphical side of the forms, to make them work is your job and that makes this section more for experienced users! Now let's hope I don't get any mails about that anymore, right?! Awesome! Just joking, I love your mails but post these kinds of questions on our forum! The community will be glad to help you out.

Anyway where were we, oh hell yeah, forms! Well I made custom checkbox, radio button, search field, select, text field, input and button. Seems like enough options for the average user right?!

Ok first of the **search box**, it is placed right under the top bar, so essentially between `<div id="topbar"></div>` and `<div id="content"></div>`, just like the tributton! This is the code you have to copy right in there:

- ✓ `<div class="searchbox"><form action="" method="get"><fieldset>input id="search" placeholder="search" type="text" /><input id="submit" type="hidden" /></fieldset></form></div>`

Next let's take a look at text inputs. You have a choice between three inputs. The first one is a **full blank line** with light grey text in the background to specify its function to the user. The second one is **half text and half input**, let me explain. On the left side it might say "Username" for example and on the right side the user can type it down. Finally you got the good old textbox for those who want to type a lot!

All three are included in a pageitem `` just like a textbox, a menu item or an iTunes list item and can be combined with all three of them in the same page item. Here are the codes of each of these in order of appearance:

- ✓ `<li class="bigfield">input placeholder="Username" type="text" />`
- ✓ `<li class="smallfield">span class="name">Nameinput placeholder="enter text" type="text" />`
- ✓ `<li class="textbox">textarea name="TextArea" rows="4"></textarea>`

By changing the placeholder attribute you choose what text the user sees in the text field before selecting it to type. Also if you add `type="password"` or to the two first inputs the text will be hidden, like for... a password, stupidly easy! Another type is `type="tel"` which will show a keypad instead of the regular keyboard.

Now let's do **checkboxes**, these are my favorite form elements because they look really awesome. It works in exactly the same way as the previous inputs; just insert it in a pageitem ``. This is the code you need:

- ✓ `<li class="checkbox">span class="name">my textinput name="checkbox" type="checkbox" /> `

Next **radio buttons**, these look sleek too. Each button has its own `` inside any pageitem `` of the page. You just need to make sure they got the same name set up so that they work together. Here is the code for 1 radio button, you need at least 2 for it to be useful, else just use a checkbox:

- ✓ `<li class="radiobutton">span class="name">Textinput name="GroupName" type="radio" value="A" />`

Next let's talk about **"selects"** when you open one up on an apple device like the iPhone, an automatic scroll button pops up from the bottom of the screen showing all the options available. This is not only very clean but also very good looking. As for the other elements just include this code inside of a pageitem ``:

- ✓ `<li class="select">select name="example">option value="1">option 1</option> <option value="2">option 2</option></select>span class="arrow">`

You can add as many `<option>`'s as you want as long as you change the value each time.

As a final point we are going to talk about the most basic form element ever: the button. Well as with all the previous components just include it in a pageitem ``.

Is it just me or am I repeating that sentence over and over again? Well at least you get it now and that was my goal! So here is the code!

```
✓ <li class="button"><input name="name" type="submit" value="Submit input" /></li>
```

The value attribute will determine the text written on the button and thanks to the type attribute you can choose between different buttons like a "submit" or "reset" button for example.

Other special elements

Let me quickly explain what this section is all about; this is the place where all items that I can't fit anywhere can be found. It's like a Tupperware box but a really awesome one where you can party and stuff... whatever.

Ok I might have exaggerated a bit because there is currently only one element in this section and that is the "double add" element. No way we're going to set up a great party with only one guest right?! So here it comes!

The **double add** is the element you can find at the top of the front page of the Appstore on your iPhone or iPod touch. Just take a look at the demo I've included, there is a double add on the Appstore list page! Alright we haven't spoken about that list yet but the special thing is that this structure can also be used on a regular page with regular page items. Just copy and paste the following code under your top bar code (and tributton/search-field code if you use those) and before the content **<div>**:

```
✓ <div id="doublead"><a href="page.html" style="background-image: url('pic1.png')"></a><a href="page2.html" style="background-image: url('pic2.png')"></a></div>
```

We will learn how to embed it in the Appstore list when the time is right. Oh and by the way you should use a 147 x 83px image for both ads. And no I did not make those numbers up, apple did, not my fault!

The list views.

Introduction

We are going to get started on some of the most awesome looking features included in iWebKit: The list views. Since version 4 I have added a couple of new ones that I am sure you will dearly love.

The Classic List

Aaaah the good old **classic list**; it's one of my favorites. Yup plain white border to border rectangles with text or even an image! You have 3 components to choose from and we are going to discover them gradually this time; because I'm in a good mood!

But first things first! To be able to build this list style you will have to replace your page's **<body>** tag with **<body class="list">**. This will simply tell the elements to behave like a classic list!

Well what makes up your page? You of course still need the top bar, next you have the content and finally the footer, but that last structure is not important. We are only going to work in the content **<div>** here.

To get started you are going to paste **** right in your content **<div>**, oh and I forgot to mention page items do not work here; kind of logical.

Now let's get to the elements. First of you can make **titles**, you know those gray bars you see across the screen to separate a list alphabetically for example? Ok the code for this is simple; just paste it in the **** you just created:

```
✓ <li class="title">your title</li>
```

That's all you need to know there!

Next up we will talk about a simple **list element with only text and a simple arrow**. As with the titles this is also just going to be an **** element that you will paste in the **** created before. Just use this code:

```
✓ <li><a href="page.html"><span class="name">name</span><span class="arrow"></span></a></li>
```

And last but certainly not least a list item with an image! Use it the same way as the previous list element.

- ✓ `<li class="withimage">Text or nameA comment`

As you certainly have noticed you not only have an extra tag for an image but also for a comment, I mean, why not use the extra space for something nice? Oh and by the way, the image can be any size as long as it is a square; it will automatically be stretched to fit the element. 90 x 90px is ideal if you go for best performance and quality.

Ok here comes a quick recapitulation: A list with 1 simple text list item

```
<body class="list">
  <div id="topbar"></div>
  <div id="content">
    <ul>
      <li>
        <a href="page.html">
          <span class="name">name</span>
          <span class="arrow"></span>
        </a>
      </li>
    </ul>
  </div>
</body>
```

The Appstore list

Ok, this is also a cool one, the **Appstore list** basically emulates, well, the Appstore obviously.

It works exactly like the classic list; start by replacing the `<body>` tag with this one: `<body class="applist">`.

Next just create an `` inside of the content `<div>` just like you did before and paste this code in it to create an applist element:

- ✓ `CommentText or name1 RatingFREE`

That's all there is to it, just repeat this code to build multiple items. Just like with iTunes classic list just replace "4" in `star4` with the number of stars you want to add to the element. The image gets automatically resized and shadows are also automatically generated. The best image size is 57x57px by the way.

Oh I almost forgot! Of course; couldn't you have told me that earlier? I just have to explain how to embed the double add in this list! Just use this special ``:

- ✓ `<li id="doublead">`

One important thing is that you have to place the double add `` as first element of your ``!

The iTunes Music List

Another nice list you might want to include is the **iTunes Music list**. You can find it in the iTunes store if you take a look at a specific album for example.

The code for this list is also very basic. As usual you need to change `<body>` to `<body class="musiclist">`. Then create the `` in the content `<div>` and then paste the following code in to create a list item:

- ✓ `1Name(1:33)`

As you see you get two new "abilities" on this list, add a number, for tracks for example as well as adding time.

The iPod list

This might be the most awesome one available in the demo because it can actually play music! You have to check it out if you haven't already! Anyhow this is how you make an iPod List.

As usual change `<body>` into `<body class="ipodlist">`, create the `` in the content `<div>` and paste this `` in there as many times as you want:

- ✓ `1Name4:11`

Ok this list has something special for you real coders out here, you can choose to automatically add an arrow to the element when you click it which is the default behavior but you can also do it manually. Just replace `` by

- ✓ ``
- ✓ ``

This feature is nice if you dynamically want to change the list.

Something else that you really should do is make the top bar black which looks a lot better, as a reminder: `<div id="topbar" class="black">`

Full Screen & its options

Introduction

Thanks to a little JavaScript hack Johan found it is possible to view your iWebKit site in Full screen mode! Try it out on the iWebKit Demo if you have no idea what I'm talking about.

How do you activate full screen?

open Safari

Visit for example: <http://m.iwebkit.net>

Touch the "+" and select "Add To Home Screen"

Confirm this and open your full screen web app straight from the home screen!

See? No task or navigation bars! It makes it almost like a native application right?! I will tell you all about it now

Enabling and disabling it

By default **full screen is enabled** on your site. Indeed it's this tag that defines the website's behavior: `<meta content="yes" name="apple-mobile-web-app-capable" />`. It is included in the `<head>` section of the document `content="yes"` means it is enabled. `content="no"` means it isn't.

Changing the status Bar color

The **status bar** is the small grey bar at the top of your full screen window; it displays the clock and battery level for example. You can leave it **gray** as it is, make it **black** and even make it **black-translucent**. Just place the elements in the `<head>` tag of your front page. Here are the codes in order of appearance:

- ✓ `<meta name="apple-mobile-web-app-status-bar-style" content="default" />`
- ✓ `<meta name="apple-mobile-web-app-status-bar-style" content="black" />`
- ✓ `<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />`

If you use the default one, just add nothing.

Oh and by the way, it cannot change between pages so just keep it on your front page or pages people are likely to web-clip.

Another small detail is that if you use the translucent one, the top bar will go behind it instead of under it which might look weird.

Add a custom icon to the home screen

You see the iWebKit logo when you clipped the demo site to your iPhone? That is simply called a **webclip icon**. Well you can make that too! You need a square image with a minimum size of 58x58px, in PNG format. Next simply add the following code to your `<head>` tag:

- ✓ `<link rel="apple-touch-icon" href="image.png" />`

Add a custom loading screen to your site

When you are loading the full screen version of the demo you will notice a nice image that says "iWebKit 5". That is a **custom load image** I added on purpose. To do the same simply make a 320x460px image and add this code to the <head> tag:

✓ `<link href="startup.png" rel="apple-touch-startup-image" />`

Outgoing links

This piece of information might be extremely valuable to you while you are developing. If you want to open an external URL in a safari window while you are in full screen you have to **add a class to your <a> tag**. Let me give you some examples: You have a link to your computer site on your mobile site and would like to open that one in safari; another example might be the wish to have an "exit full screen" button.

The same counts for links that open applications on the iphone like mail, phone, the appstore, etc. But you will discover those a little later on!

Also it can be handy if you use an onclick JavaScript event, just to avoid eventual interactions.

Integration

Introduction

Here we are going to talk about different ways to make your website communicate with iPhone software or other sites even making it possible to open up an application! In these examples you will need to use the previously discussed `class="noeffect"` so that the app's open in full screen too!

Detecting phone numbers

To **detect phone numbers** and transform them into links add this code to your <head>:

✓ `<meta name="format-detection" content="telephone=yes" />`

Change `telephone=yes` to `telephone=no` to disable it.

Set up an e-mail

You can send a **custom e-mail message** by clicking on a special made link of your website. Just make a change any an element you want inserting the following code in the href:

✓ `mailto:adress@server.com.`

Example: ``

You can also add a cc: `mailto:lol@test.com?cc=ex@ample.net`

Or add a subject: `mailto:lol@test.com&subject=my subject`

And a body `mailto:lol@test.com&body=my body text`

Of course you can combine all of them together too!

Call a number/ add to contact

This is another link transformation and it has a different behavior depending on the device you are using. On an iPhone it will call the number and on an iPod touch it will ask you if you want to save it in your contact book. You can use the code on any <a> element:

✓ `call now`

Open a YouTube video

This is also simple; just add the URL of a YouTube video to your link and it will open up when clicked!

✓ `Watch`

Send an SMS to number

This works the same as for the "call a number":

✓ `send`

Open Google maps at a specific location

Like with the previous items, this also is just something you add as a link. You can find the right code to use on <http://maps.google.com>:

✓ `map of america`

Open iTunes or Appstore and show a specific item

To get links to your favorite music or application go to iTunes, next your favorite album or application; right clicking on the image and click "copy link". Next just use it in a regular link:

✓ `open album`

Embed an RSS feed

This feature requires PHP to be enabled on your website. On some servers this technology isn't available or simply not turned on.

Next there are two steps you will need to follow to show your blog on your iWebKit website. First paste the following code right in the content `<div>`:

```
<?php include("php/rss.php");?>
```

You can still add other elements too like titles or form elements. Next you have to go to the php folder, open rss.php and look for this line:

```
$blog_url = "HTTP://YOURSITE.COM/FEED.XML";
```

Simply change that link to the url of your rss feed! Oh and one last thing, don't forget to change your page extension from .html to .php!

Important notes and information

Useful links and content

<http://developer.apple.com/safari/>

<http://www.w3.org/TR/>

<http://alistapart.com/articles/putyourcontentinmypocket>

<http://alistapart.com/articles/putyourcontentinmypocketpart2>

GNU © 2010 iWebKit