* <https://developer.apple.com/documentation/uikit/uiwebview?language=objc>
* Class(类)：
* UIWebView
* A view that embeds web content in your app.(将网页内容嵌入您的应用中的视图。)
* Overview
* **Note**

In apps that run in iOS 8 and later, use the [WKWebView](https://developer.apple.com/documentation/webkit/wkwebview?language=objc) class instead of using UIWebView. Additionally, consider setting the [WKPreferences](https://developer.apple.com/documentation/webkit/wkpreferences?language=objc) property [javaScriptEnabled](https://developer.apple.com/documentation/webkit/wkpreferences/1536203-javascriptenabled?language=objc) to NOif you render files that are not supposed to run JavaScript.

* **Important**

An iOS app linked on or after iOS 10.0 must include in its Info.plist file the usage description keys for the types of data it needs to access or it will crash. To access a user’s photo data specifically, it must include [NSPhotoLibraryUsageDescription](https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW17) and [NSCameraUsageDescription](https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW24).

* Use the [loadHTMLString:baseURL:](https://developer.apple.com/documentation/uikit/uiwebview/1617979-loadhtmlstring?language=objc) method to begin loading local HTML files or the [loadRequest:](https://developer.apple.com/documentation/uikit/uiwebview/1617957-loadrequest?language=objc) method to begin loading web content. Use the [stopLoading](https://developer.apple.com/documentation/uikit/uiwebview/1617974-stoploading?language=objc) method to stop loading, and the [loading](https://developer.apple.com/documentation/uikit/uiwebview/1617978-loading?language=objc) property to find out if a web view is in the process of loading.

If you allow the user to move back and forward through the webpage history, then you can use the [goBack](https://developer.apple.com/documentation/uikit/uiwebview/1617975-goback?language=objc) and [goForward](https://developer.apple.com/documentation/uikit/uiwebview/1617940-goforward?language=objc) methods as actions for buttons. Use the [canGoBack](https://developer.apple.com/documentation/uikit/uiwebview/1617931-cangoback?language=objc) and [canGoForward](https://developer.apple.com/documentation/uikit/uiwebview/1617951-cangoforward?language=objc) properties to disable the buttons when the user can’t move in a direction.

By default, a web view automatically converts telephone numbers that appear in web content to Phone links. When a Phone link is tapped, the Phone app launches and dials the number. To turn off this default behavior, set the [dataDetectorTypes](https://developer.apple.com/documentation/uikit/uiwebview/1617965-datadetectortypes?language=objc) property with a [UIDataDetectorTypes](https://developer.apple.com/documentation/uikit/uidatadetectortypes?language=objc) bitfield that does not contain the [UIDataDetectorTypePhoneNumber](https://developer.apple.com/documentation/uikit/uidatadetectortypes/uidatadetectortypephonenumber?language=objc) flag.

You can also use the [scalesPageToFit](https://developer.apple.com/documentation/uikit/uiwebview/1617950-scalespagetofit?language=objc) property to programmatically set the scale of web content the first time it is displayed in a web view. Thereafter, the user can change the scale using gestures.

Set the [delegate](https://developer.apple.com/documentation/uikit/uiwebview/1617937-delegate?language=objc) property to an object conforming to the [UIWebViewDelegate](https://developer.apple.com/documentation/uikit/uiwebviewdelegate?language=objc) protocol if you want to track the loading of web content.

* **Important**

You should not embed UIWebView or [UITableView](https://developer.apple.com/documentation/uikit/uitableview?language=objc) objects in [UIScrollView](https://developer.apple.com/documentation/uikit/uiscrollview?language=objc) objects. If you do so, unexpected behavior can result because touch events for the two objects can be mixed up and wrongly handled.

* You can easily debug the HTML, CSS, and JavaScript contained inside a UIWebView with Web Inspector. Read Debugging Web Content on iOS to learn how to configure Web Inspector for iOS. Read the rest of [Safari Web Content Guide](https://developer.apple.com/library/content/documentation/AppleApplications/Reference/SafariWebContent/Introduction/Introduction.html#//apple_ref/doc/uid/TP40002051) to learn how to create web content that is optimized for Safari on iPhone and iPad.

For information about basic view behaviors, see [View Programming Guide for iOS](https://developer.apple.com/library/content/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/Introduction/Introduction.html#//apple_ref/doc/uid/TP40009503).

* Supported File Formas
* In addition to HTML content, UIWebView objects can be used to display other content types, such as Keynote, PDF, and Pages documents. For the best rendering of plain and rich text in your app, however, you should use [UITextView](https://developer.apple.com/documentation/uikit/uitextview?language=objc) instead.
* State Preservation
* In iOS 6 and later, if you assign a value to this view’s [restorationIdentifier](https://developer.apple.com/documentation/uikit/uiviewcontroller/1621499-restorationidentifier?language=objc) property, it attempts to preserve its URL history, the scaling and scrolling positions for each page, and information about which page is currently being viewed. During restoration, the view restores these values so that the web content appears just as it did before. For more information about how state preservation and restoration works, see [App Programming Guide for iOS](https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007072).

For more information about appearance and behavior configuration, see [Web Views](https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/UIKitUICatalog/UIWebView.html#//apple_ref/doc/uid/TP40012857-UIWebView).

* Subclassing Notes
* The UIWebView class should not be subclassed.
* Topics
* Responding to Web View Changes
* Delegate
* Instance Property
* The receiver’s delegate
* Declaration
* @property(nonatomic, assign) id<UIWebViewDelegate> delegate;
* Discussion
* The delegate is sent messages when content is loading. See [UIWebViewDelegate](https://developer.apple.com/documentation/uikit/uiwebviewdelegate?language=objc) for the optional methods this delegate may implement.
* Importance
* Before releasing an instance of UIWebView for which you have set a delegate, you must first set its delegate property to nil. This can be done, for example, in your dealloc method.
* See Also
* Responding to Web View Changes
* UIWebViewDelegate
* Protocol
* The UIWebViewDelegate protocol defines methods that a delegate of a UIWebView object can optionally implement to intervene when web content is loaded.
* OverView
* **Important**
* Before releasing an instance of UIWebView for which you have set a delegate, you must first set the UIWebView delegate property to nil before disposing of the UIWebViewinstance. This can be done, for example, in the dealloc method where you dispose of the UIWebView.
* Topics
* Loading Content
* webView:shouldStarLoadWithRequest:navigationType;
* Instance Method
* Sent before a web view begins loading a frame.
* Declaration
* - (BOOL)webView:(UIWebView \*)webView

shouldStartLoadWithRequest:(NSURLRequest \*)request

navigationType:(UIWebViewNavigationType)navigationType;

* Parameters
* webView:The web view that is about to load a new frame.
* request:The content location.
* navigationType:The type of user action that stared the load request.
* Return Value
* YES if the web view should begin loading content;otherwise,NO.
* webViewDidStarLoad:
* Iinstance Method
* Sent after a web view starts loading a frame.
* Declaration
* - (void)webViewDidStartLoad:(UIWebView \*)webView;
* Parameters
* webView:The web view that has begun loading a new frame.
* webViewDidFinishLoad:
* Instance Method
* Sent after a web view finishes loading a frame.
* Declaration
* - (void)webViewDidFinishLoad:(UIWebView \*)webView;
* Parameters
* webView:The web view has finished loading.
* WebView:didFailLoadWithError;
* Instance Method
* Sent if a web view failed to load a frame.
* Declaration
* -(void)webView:(UIWebView \*)webView

didFailLoadWithError:(NSError \*)error;

* Parameters
* webView:The error that failed to load a frame.
* Error:The error that occurred during loading.
* Relationships
* Inherits From
* NSObject
* Loading Content
* loadData:MIMEType：textEncodingName:baseURL:
* Instance Method
* Set the main page contens,MIME type,content encoding,and base URL,
* Declaration
* -(void)loadData:(NSData \*)data

MIMEType:(NSString \*)MIMEType

textEncodingName:(NSString \*)textEncodingName

baseURL:(NSURL \*)baseURL;

* Parameters
* Data:The content for the main page.
* MIMEType:The MIME TYPE of the content.
* encodingName:The IANA encoding name as in utf-8 or utf-16.
* baseURL:The base URL for the content.
* loadHTMLString:baseURL:
* Instance Metho