

# 웹기반 앱 프로토타이핑

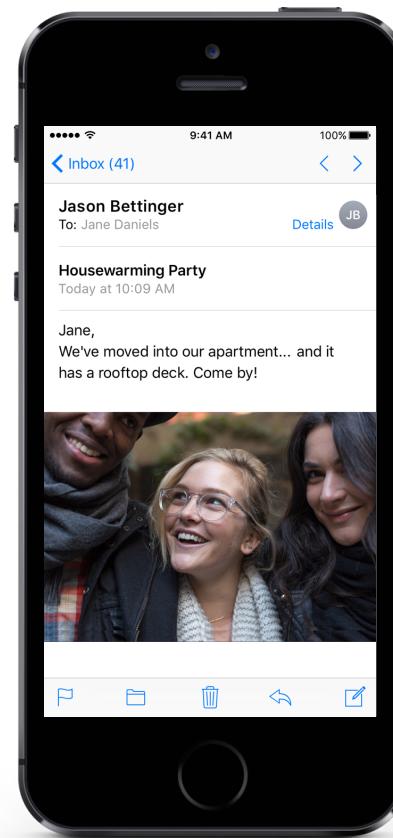
WebView, WebKit, JavaScriptCore



# Web Views

A web view loads and displays rich web content, such as embedded HTML and websites, directly within your app.

Mail uses a web view, for example, to show HTML content in messages.



# 웹 앱 개발 #1

UIWebView





# UIWebView

- \* UIKit.framework 일부
- \* 앱에 웹 콘텐츠를 포함시키는 방법



HTML



IMAGE



CSS



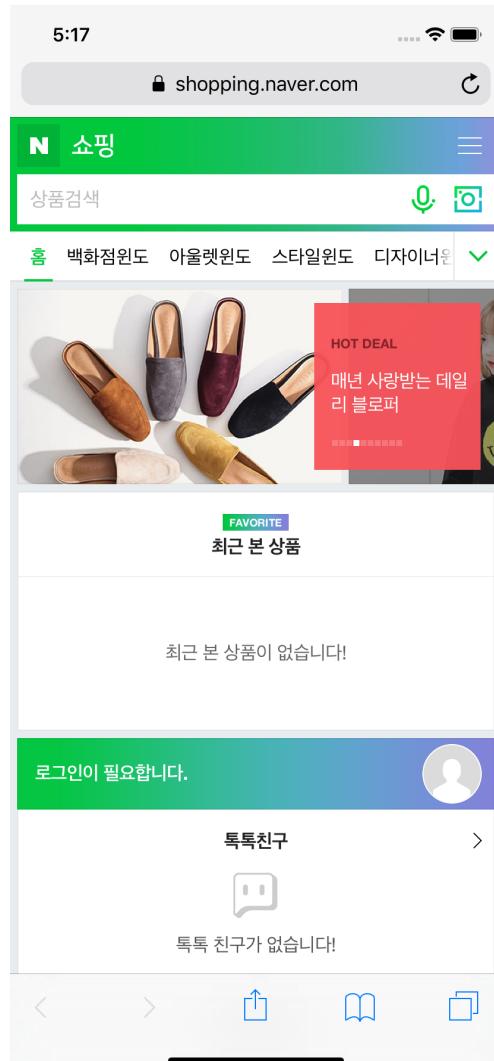
JS

# 웹 앱 만들기

NAVER 쇼핑 앱

# 네이버 쇼핑 웹-앱

<https://m.shopping.naver.com>



# UIWebViewDelegate

```
func webViewDidStartLoad(_ webView: UIWebView) {
    print(#function)
}

func webViewDidFinishLoad(_ webView: UIWebView) {
    print(#function)
}

func webView(_ webView: UIWebView, didFailLoadWithError error: Error) {
    print(#function)
}

func webView(_ webView: UIWebView, shouldStartLoadWith request: URLRequest,
            navigationType: UIWebViewNavigationType) -> Bool {
    print("\(request.url?.absoluteString)")

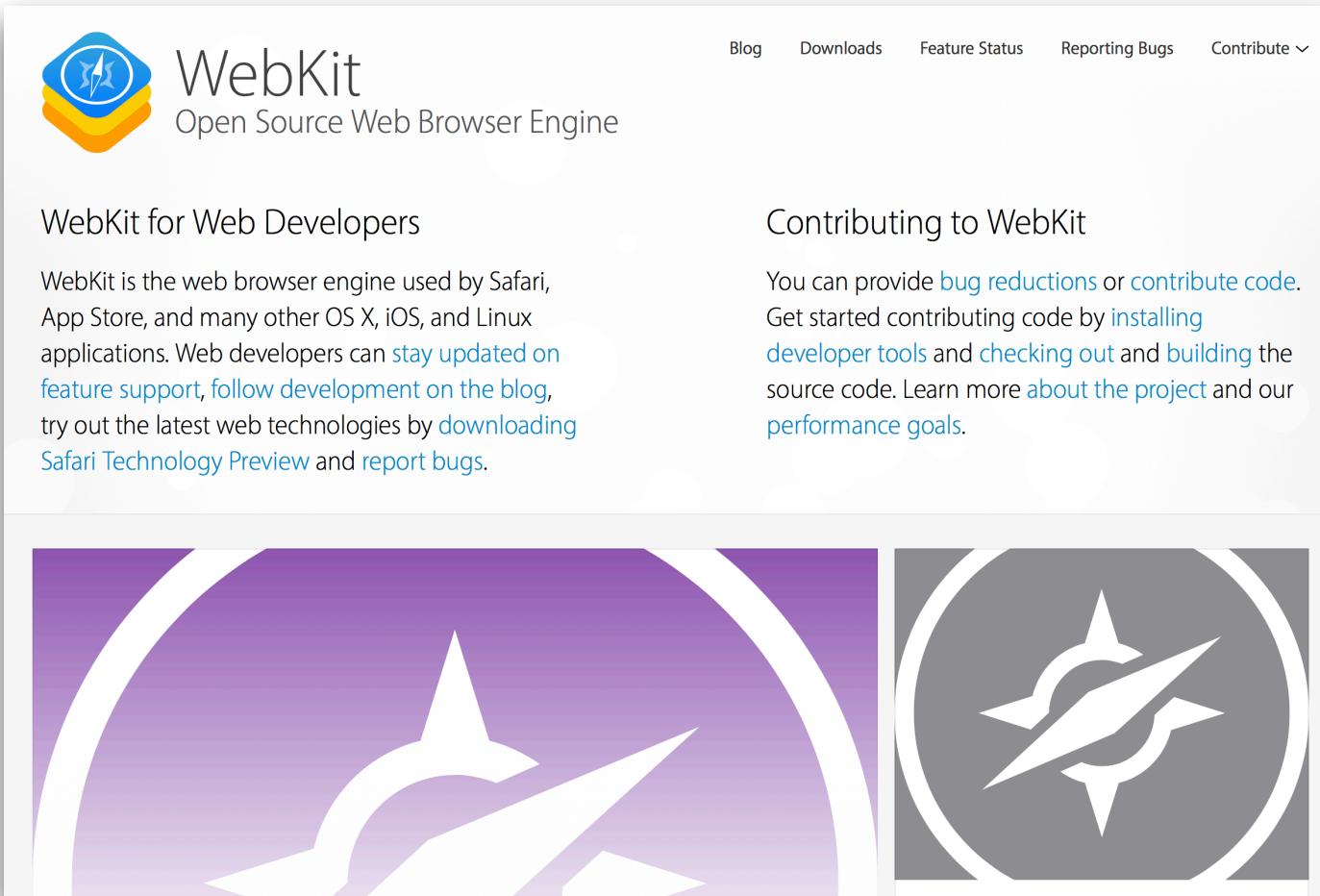
    let popup = request.url?.absoluteString.contains("pop.php") ?? true
    return !popup
}
```

# 웹 앱 개발 #2

WebKit Framework



# WebKit Open source Project



The screenshot shows the homepage of the WebKit Open Source Project. At the top, there's a navigation bar with links for Blog, Downloads, Feature Status, Reporting Bugs, and Contribute. Below the navigation is the WebKit logo, which consists of three overlapping rounded squares in blue, yellow, and orange, with a blue icon containing a white lightning bolt to its right. The text "WebKit" is written in a large serif font, with "Open Source Web Browser Engine" in a smaller sans-serif font below it. The main content area is divided into two columns. The left column is titled "WebKit for Web Developers" and contains text about the engine's use in Safari, App Store, and other OS X, iOS, and Linux applications. It encourages developers to stay updated on feature support, follow development on the blog, try out the latest web technologies by downloading Safari Technology Preview, and report bugs. The right column is titled "Contributing to WebKit" and provides instructions for contributing code, including links for bug reductions, contributing code, installing developer tools, checking out the source code, and building it. It also links to more information about the project and performance goals. At the bottom of the page is a decorative footer section featuring a purple gradient background with a white abstract shape and a circular gray background with a white compass-like icon.

Blog   Downloads   Feature Status   Reporting Bugs   Contribute ▾

 **WebKit**  
Open Source Web Browser Engine

WebKit for Web Developers

WebKit is the web browser engine used by Safari, App Store, and many other OS X, iOS, and Linux applications. Web developers can [stay updated on feature support](#), [follow development](#) on the blog, try out the latest web technologies by [downloading Safari Technology Preview](#) and [report bugs](#).

Contributing to WebKit

You can provide [bug reductions](#) or [contribute code](#). Get started contributing code by [installing developer tools](#) and [checking out](#) and [building](#) the source code. Learn more [about the project](#) and our [performance goals](#).



WKScriptMessageHandler

WKWebView

WKBackForwardList

WKFrameInfo

WKNavigationDelegate

WKUserContentController

WKBackForwardList

WKNavigation

WKScriptMessage

WKPreferences

WKUIDelegate

WKNavigationType

WKProcessPool

WKUserScript

WKNavigationAction

WKNavigationResponse

WKWindowFeatures

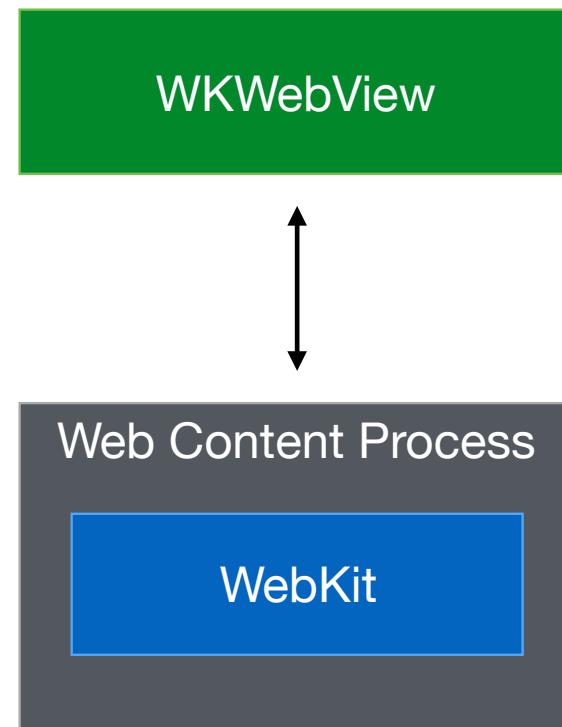
WKWebViewConfiguration



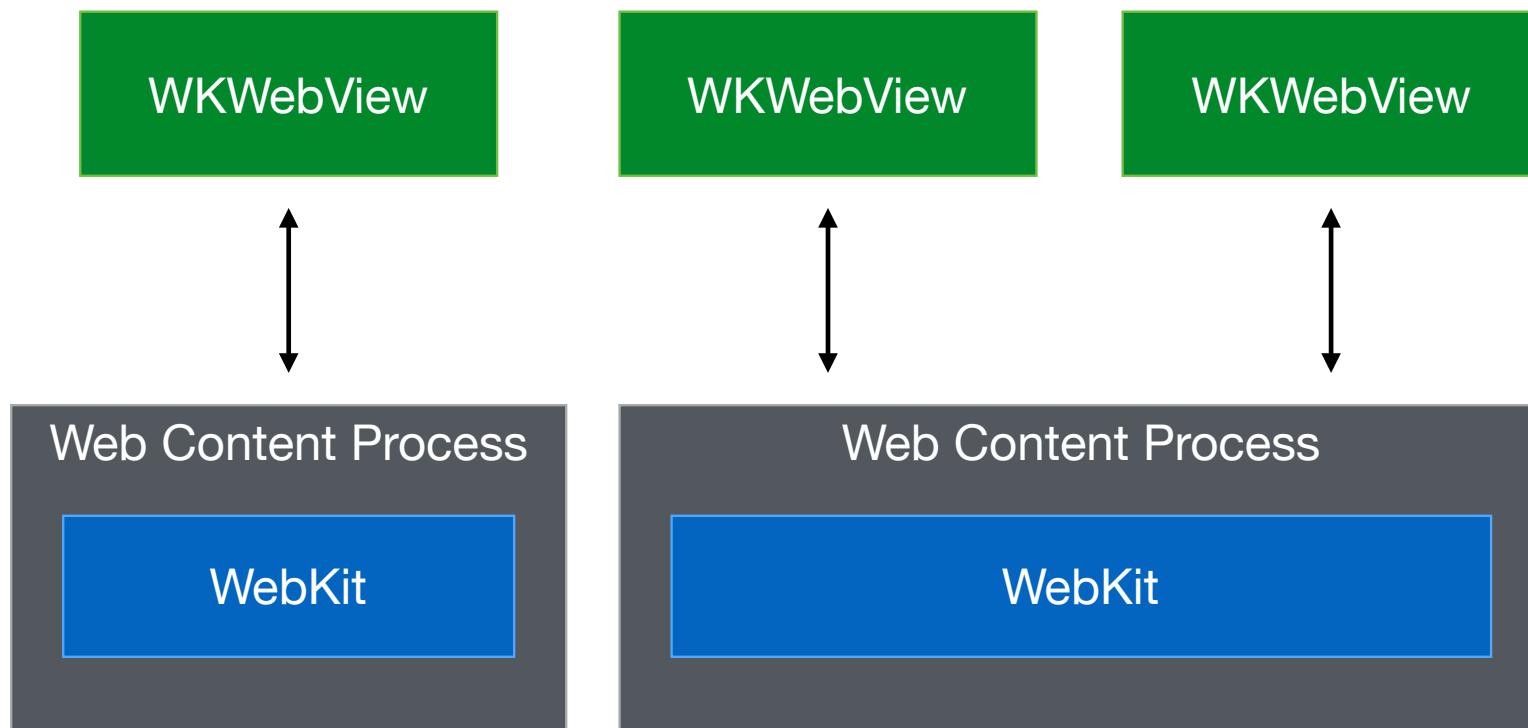
# Modern WebKit API

- \* macOS 와 iOS 동일한 Modern APIs
- \* Multi-process 아키텍처
- \* 60Hz 부드러운 스크롤 지원
- \* Nitro JS 엔진
- \* Swipe 제스처 지원
- \* 간편한 앱-웹페이지 통신 방식

# Multi-process 아키텍처



# Multi-process 아키텍처



# WKWebView로 불러오기

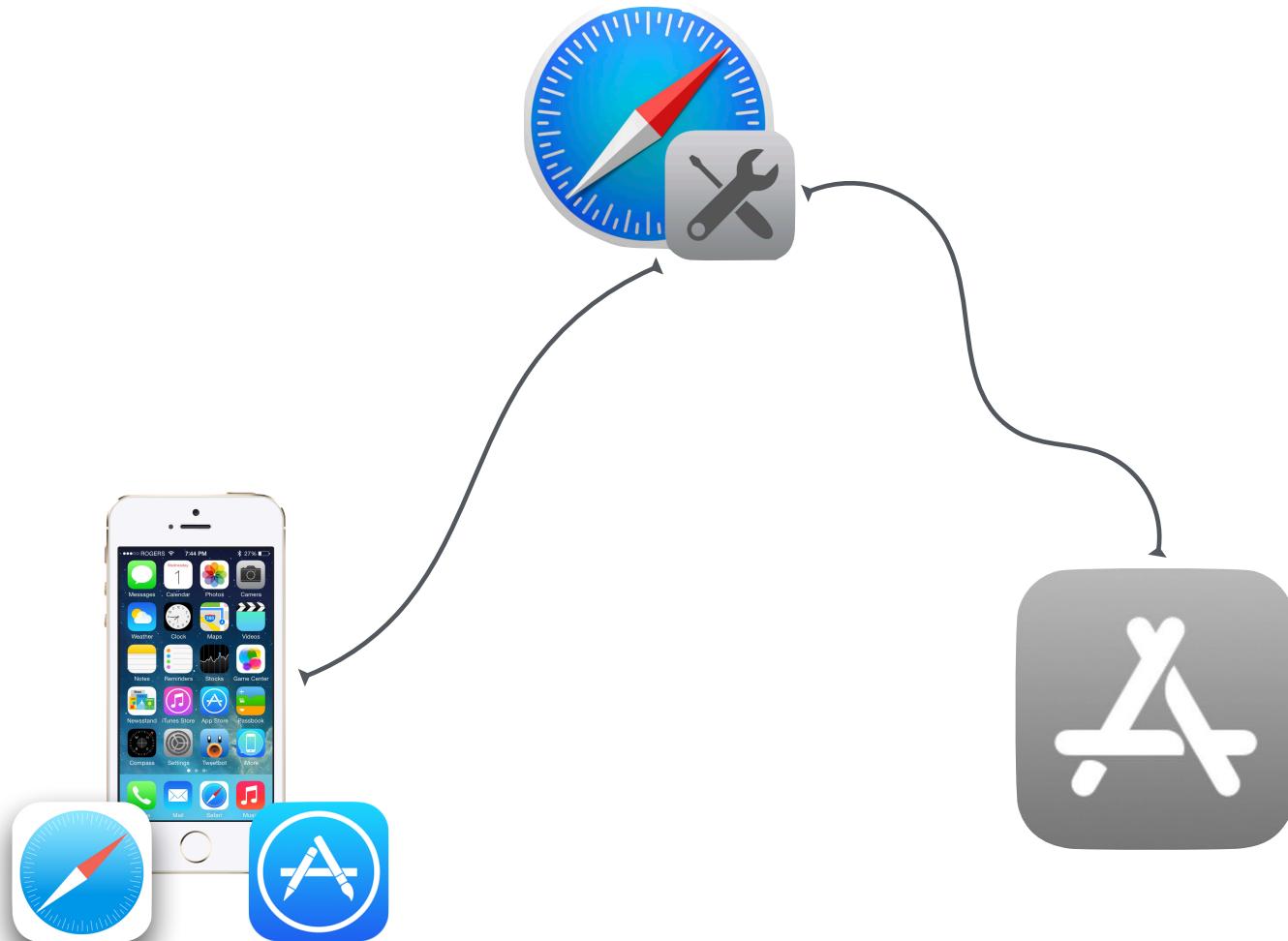
```
import UIKit
import WebKit

class ViewController: UIViewController {
    var webview : WKWebView

    required init?(coder aDecoder: NSCoder) {
        webview = WKWebView.init(frame: UIScreen.main.bounds)
        super.init(coder: aDecoder)
    }

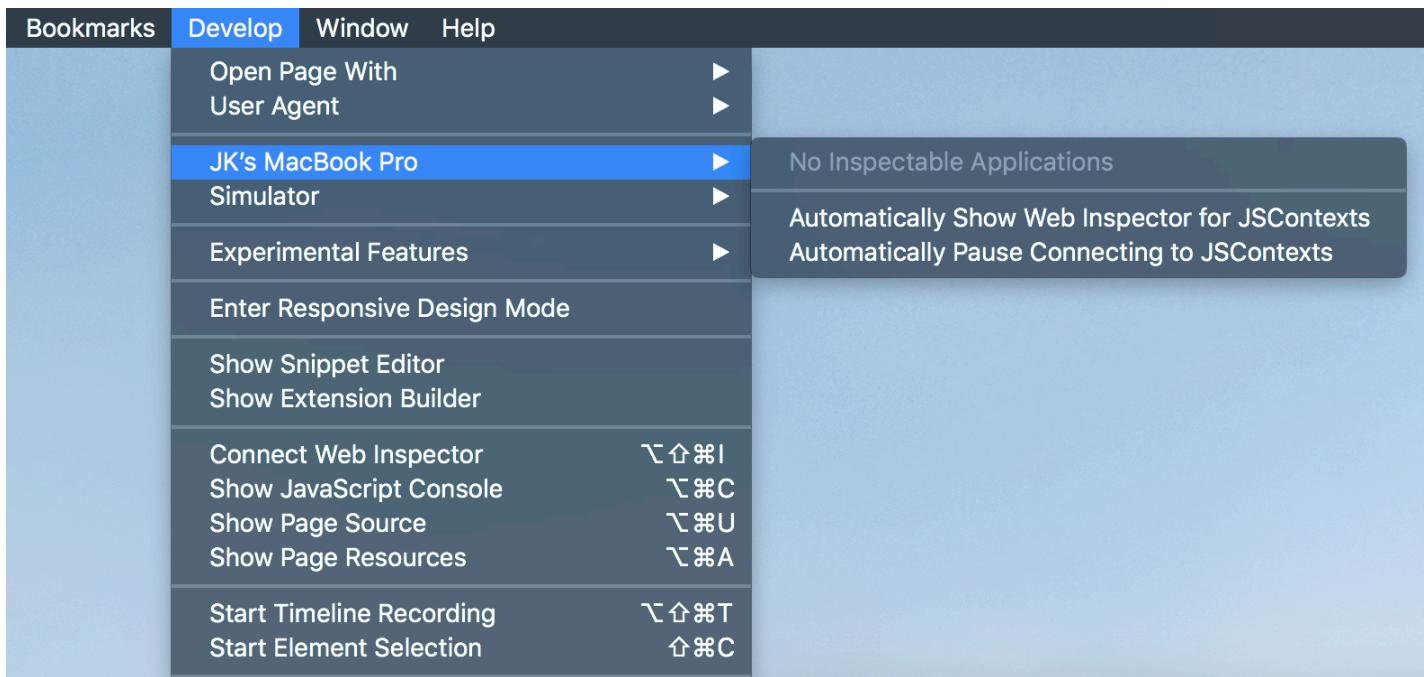
    override func viewDidLoad() {
        super.viewDidLoad()
        self.view.addSubview(webview)

        webview.load(URLRequest(url: URL(string:"https://m.shopping.naver.com")!))
    }
}
```



# Web Inspector

- \* macOS / iOS App Inspection
- \* JavaScriptCore Support
- \* Source code timelines
- \* Breakpoint Probes
- \* Accessibility details
- \* Color and Gradient editing





"com.apple.security.get-task-allow"

Web Inspector — Simulator — HelloWebkit — m.baeminfresh.com — /

Debugger Paused | Global Code — pop.php:14

Elements Network Resources Timelines Debugger Storage Console +

All Resources Documents m.baeminfresh.com

**m.baeminfresh.com**

- Frames
- Images
- Scripts
  - base64.js
  - common-a.js
  - fresh.js
  - polyfill.js
  - trs\_esc.js — analysis.adinsight.co.kr
  - cssua.min.js — cdn.bmf.kr
  - joho-core.0.0.8.js — cdn.bmf.kr
  - jquery-2.2.0.min.js — cdn.bmf.kr
  - jquery-ui.min.js — cdn.bmf.kr
  - m.footer.bundle.js — cdn.bmf.kr
  - m.index.bundle.js — cdn.bmf.kr
  - swiper.jquery.min.js — cdn.bmf.kr
  - URI.min.js — cdn.bmf.kr
  - 1587950671532924 — connect.facebook...
  - fbevents.js — connect.facebook.net
  - wcslog.js — wcs.naver.net
  - analytics.js — www.google-analytics.com
  - ec.js — www.google-analytics.com
  - conversion.js — www.googleadservices.com
- Stylesheets
- XHRs

Filter Resource List

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5     <meta name="keywords" content="반찬배달, 반찬배송, 가정식 반찬배달, 아이반찬, 전국맛집, 새벽배송">
6     <meta name="description" content="반찬배달, 반찬배송, 가정식 반찬배달, 아이반찬, 전국맛집, 새벽배송">
7     <meta name="author" content="베민프레시">
8     <meta name="robots" content="INDEX,FOLLOW">
9     <meta name="robots" content="ALL">
10    <meta name="revisit-after" content="1 days">
11    <meta http-equiv="X-UA-Compatible" content="IE=Edge"/><script type="text/javascript">(window.NREUM||(NREUM={})).item=window.item||[]</script>
12    <meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,user-scalable=no" />
13    <meta name="naver-site-verification" content="e543b30ae873355f2c556f58b48ae7c95c1cea05"/>
14    <title>베민프레시 - 모바일 넘버별 반찬가게</title>
15    <link rel="apple-touch-icon" href="/template/image/bg/bookmark.png" />
16    <link rel="shortcut icon" href="https://cdn.bmf.kr/favicon.ico" />
17    <link rel="stylesheet" type="text/css" href="https://cdn.bmf.kr/static/m.css/vendor.min.v-5a377d17.css" />
18    <link rel="stylesheet" type="text/css" href="https://cdn.bmf.kr/static/m.css/app.min.v-5a377d17.css" />
19
20    <script type="text/javascript">
21      (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){(i[r].q=i[r].q||[])
22        ga('create', 'UA-65873686-1', 'auto');
23        ga('set', 'dataSource', 'web (mobile)');
24        ga('require', 'ec');
25        ga('send', 'pageview');
26      })();
27      <script type="text/javascript">
28        var hid_frame = "hidden1503045511";
29        var mv = "10";
30        var alv = "";
31        var root_url = "https://m.baeminfresh.com";
32        var engine_url = "https://m.baeminfresh.com/_engine";
33        var this_url = "https://m.baeminfresh.com/";
34        var upload_base="https://cdn.bmf.kr";
35        var ssl_url = "https://m.baeminfresh.com";
36        var ssl_tunnel_url = "https://m.baeminfresh.com";
37        var ssl_tunnel_use = false;
38        var soldout_name = "품절";
39        var cy_sid = "";
40
41      <script type="text/javascript" src="https://cdn.bmf.kr/web/js/URI/1.17.0/URI.min.js"></script>
42      <script type="text/javascript" src="https://cdn.bmf.kr/web/js/jquery/jquery-2.2.0.min.js"></script>
43      <script type="text/javascript" src="https://cdn.bmf.kr/web/js/bootstrap/bootstrap.min.js"></script>
44

```

Resource

Type

MIME Type text/html  
Resource Type Document

Location

Full URL https://m.baeminfresh.com/  
Scheme https  
Host m.baeminfresh.com  
Path /

Initiated

- vendor.min.v-5a377d17.css
- app.min.v-5a377d17.css
- URI.min.js
- jquery-2.2.0.min.js
- jquery-ui.min.js
- cssua.min.js
- joho-core.0.0.8.js
- swiper.jquery.min.js
- base64.js
- common-a.js
- polyfill.js
- fresh.js
- bf-logo.png
- 1502872580989\_9e50ae1c2332.jpg
- 1502949132719\_bab0f4adb628.jpg
- 150164589964\_76a025b2060.jpg
- 1501825614154\_cede71bc0126.jpg
- 1501733789751\_65322296b88e.jpg
- 1502683989408\_23f214236bd.jpg
- 1501658986070\_686dfce4f299.jpg
- 1502944738930\_iae83775254.jpg
- 1502426079243\_591dca169863.jpg
- 1502266606273\_87e7af3f292b.jpg
- 1503005008448\_253fc319153.jpg
- 1502945257626\_973099681614.jpg
- 1502343825853\_3a9a031cc89.jpg
- 1502946680002\_104dd4186cf5.jpg
- 1501243755604\_54efe28304b1.jpg
- m.index.bundle.js
- wcslog.js

Web Inspector — Simulator — HelloWebKit — m.baeminfresh.com — /

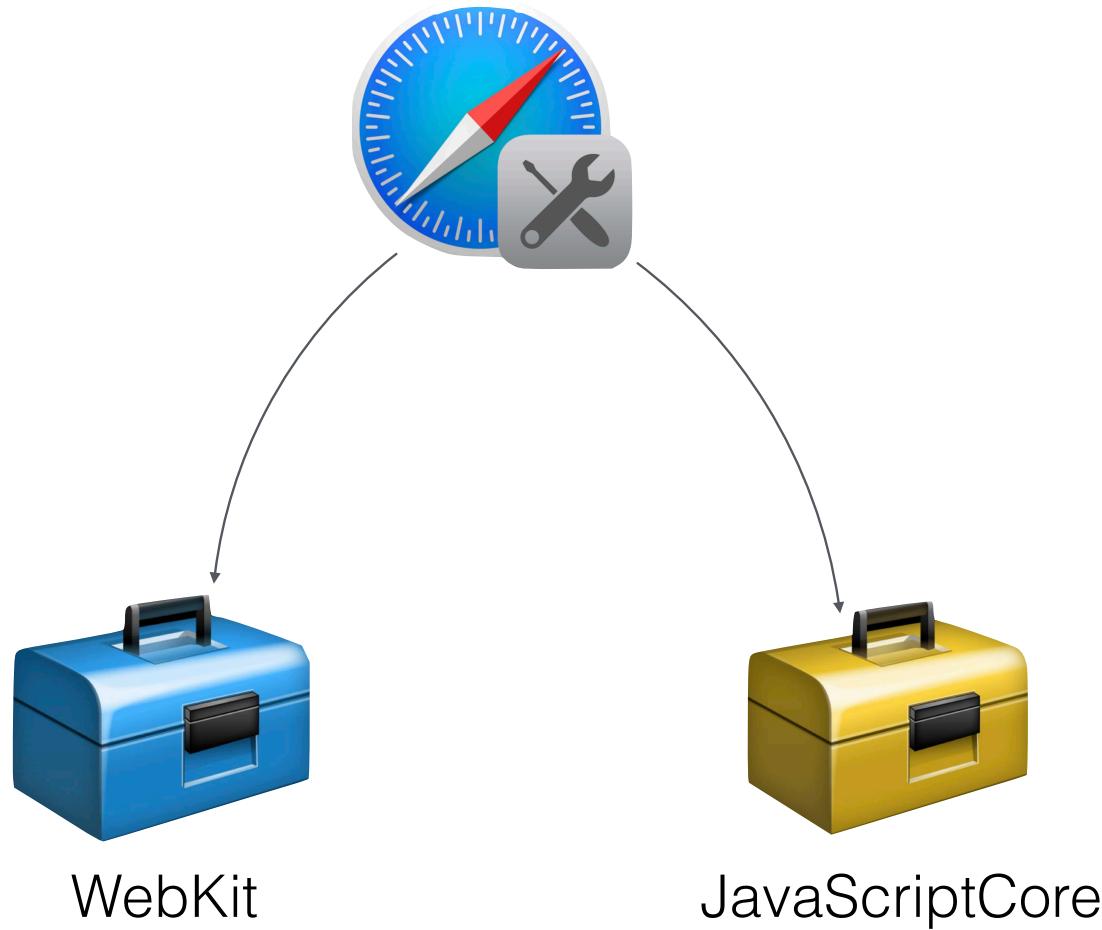
Debugger Paused | Global Code — pop.php:14

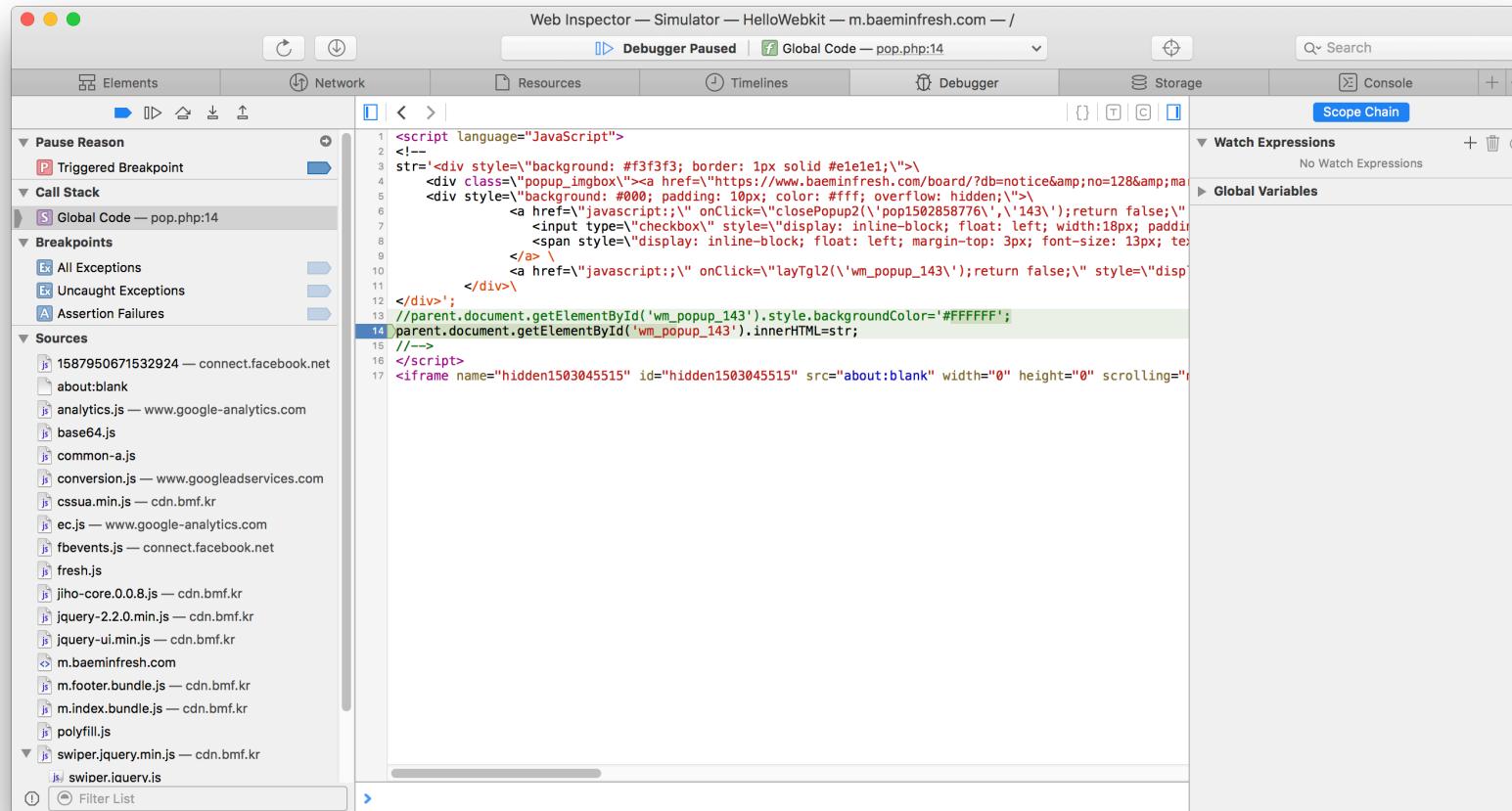
All Resources Documents ▾

Name Domain Type Method Status Cache Size Transferred Start Time Latency Duration

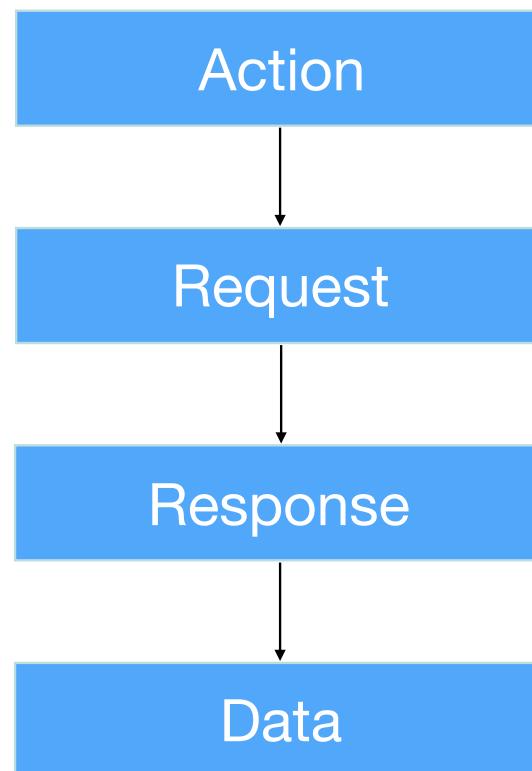
Name	Domain	Type	Method	Status	Cached	Size	Transferred	Start Time	Latency	Duration
									10.00s	20.00s
pixel	bid.g.doubleclick.net	Document	GET	HTTPS	200	No	—	624 B	4.32s	56.80ms -3.25s
877049423	www.google.co.kr	Document	GET	HTTPS	200	No	63 B	475 B	4.32s	220.1ms -220.5ms
pop.php	m.baeminfresh.com	Document	GET	HTTPS	200	No	1.58 KB	1.22 KB	4.30s	45.01ms 114.5ms
m.baeminfresh.com	m.baeminfresh.com	Document	GET	HTTPS	200	No	91.13 KB	20.32 KB	0.796ms	373.9ms 48.70ms
m	wcs.naver.com	Image	GET	HTTPS	—	—	—	—	4.31s	—
log.php	analysis.adnsig...	Image	GET	HTTPS	200	No	93 B	403 B	4.30s	11.34ms 34.57ms
ga-audiences	www.google.co.kr	Image	GET	HTTPS	200	No	42 B	522 B	3.39s	193.0ms 446.6ms
tr	www.facebook....	Image	GET	HTTPS	—	—	—	—	3.26s	—
7c7b2c57cbfefaeda5c7a7cd08475b1.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	189.92...	369 B	3.20s	6.512ms 28.24ms
e0ced8769e6938b58b15e91cb1dfe31b.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	109.24...	369 B	3.20s	4.825ms 29.61ms
bf-icon-more.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	1.74 KB	332 B	3.15s	5.579ms 45.66ms
bf-main-category-tab.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	13.70 KB	332 B	3.15s	5.871ms 49.37ms
bf-btn-basket.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	1.35 KB	332 B	3.15s	6.303ms 49.80ms
bf-btn-search.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	1.91 KB	332 B	3.15s	7.335ms 49.94ms
bf-btn-menu.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	1.05 KB	332 B	3.15s	6.319ms 49.42ms
1501243755604_54efe28304b1.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	23.69 KB	369 B	590.9ms	6.864ms 57.00ms
1502946680002_104dd4186cf5.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	108.65...	368 B	590.6ms	6.198ms 57.68ms
1502343825853_3a19a031cc89.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	133.94...	369 B	590.3ms	6.071ms 60.49ms
1502945257626_973099681614.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	135.20...	369 B	590.0ms	5.295ms 62.22ms
1503005008448_253ff3f9153.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	121.28...	369 B	589.7ms	4.588ms 63.10ms
1502266606273_87e7af3f292b.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	159.40...	368 B	589.4ms	4.945ms 63.15ms
1502426079243_591dcfa169863.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	146.99...	369 B	589.1ms	5.973ms 65.12ms
1502944738930_1ae8377525f4.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	125.22...	369 B	588.8ms	5.971ms 67.07ms
1501658986076_686dfce4f259.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	98.01 KB	368 B	588.5ms	6.325ms 66.53ms
1502683989408_23f2f4236bd.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	192.63...	369 B	588.2ms	7.426ms 66.36ms
1501733789751_65322296b88e.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	202.94...	369 B	587.9ms	5.138ms 69.83ms
1501825614154_cede71bc0126.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	114.60...	369 B	587.7ms	5.250ms 68.66ms
1501564589964_76a0258b2060.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	169.49...	369 B	587.4ms	5.139ms 66.39ms
1502949132719_bab0f4adb628.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	154.44...	368 B	587.1ms	9.479ms 66.21ms
1502872580989_9e50ae1c2332.jpg	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	81.89 KB	369 B	586.8ms	6.264ms 67.78ms
bf-logo.png	cdn.bmf.kr	Image	GET	HTTPS	304	Yes	5.14 KB	331 B	586.5ms	6.171ms 67.28ms

Filter Resource List >

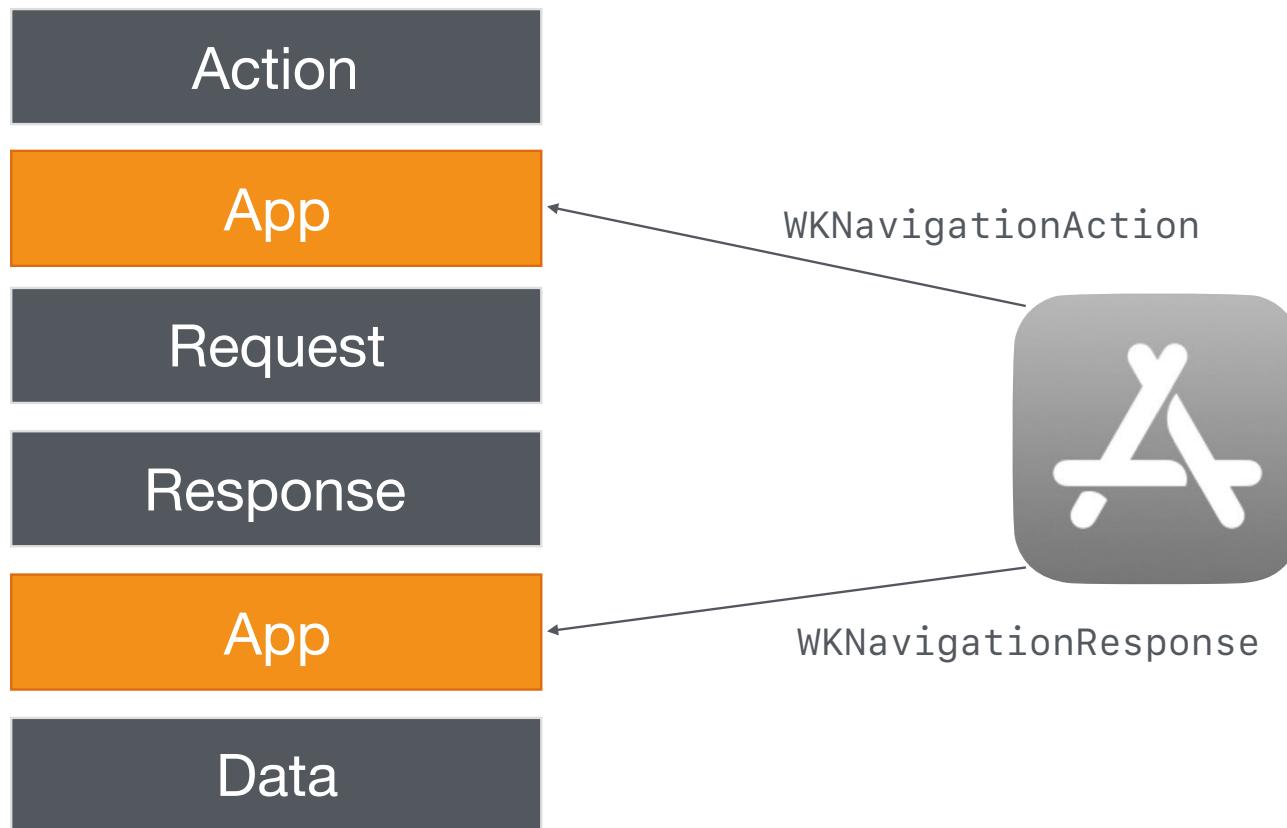




# Page Loading



# Page Loading



# Custom Page Loading

## WKNavigationDelegate

```
weak var navigationDelegate: WKNavigationDelegate?

optional func webView(_ webView: WKWebView,
decidePolicyFor navigationAction: WKNavigationAction,
decisionHandler: @escaping (WKNavigationActionPolicy) -> Swift.Void)

optional func webView(_ webView: WKWebView,
decidePolicyFor navigationResponse: WKNavigationResponse,
decisionHandler: @escaping (WKNavigationResponsePolicy) -> Swift.Void)
```

# Custom Page Loading

WKNavigationAction

Name	Type
navigationType	WKNavigationType
sourceFrame	WKFrameInfo
destinationFrame	WKFrameInfo
request	NSURLRequest
modifierFlags	NSEventModifierFlags

# Custom Page Loading

## WKNavigationResponse

Name	Type
response	NSURLResponse
forMainFrame	Bool
canShowMIMEType	Bool

# Custom Page Loading

## Decision Handler

- \* Cancel or Allow 결정
- \* Navigation action
  - \* [WKNavigationActionPolicyCancel](#)
  - \* [WKNavigationActionPolicyAllow](#)
- \* Navigation response
  - \* [WKNavigationResponsePolicyCancel](#)
  - \* [WKNavigationResponsePolicyAllow](#)
- \* Asynchronous

# Custom Page Loading

## WKNavigationDelegate

```
extension ViewController: WKNavigationDelegate {
    func webView(_ webView: WKWebView,
                 decidePolicyFor navigationAction: WKNavigationAction,
                 decisionHandler: @escaping (WKNavigationActionPolicy) -> Void) {
        let baemin = navigationAction.request.url?.host?.hasSuffix("naver.com") ?? false
        if baemin {
            decisionHandler(.allow)
        } else {
            decisionHandler(.cancel)
        }
    }
}
```

# Custom Webpage Content

- \* User Scripts
- \* Script Messages
- \* WKUserContentController 활용하기
- \* WKWebViewConfiguration 일부

# User Scripts

- \* 언제 실행할지
  - \* Document Start
  - \* Document End
- \* 어디서 실행할지
  - \* All Frames
  - \* Main frame only

# User Scripts

## WKUserScript

```
var configuration = WKWebViewConfiguration()

required init?(coder aDecoder: NSCoder) {
    let scriptString = "alert('hello world')"
    let popupScript = WKUserScript(source: scriptString!,
                                   injectionTime: WKUserScriptInjectionTime.atDocumentEnd,
                                   forMainFrameOnly: false)
    configuration.userContentController.addUserScript(popupScript)
    webview = WKWebView.init(frame: UIScreen.main.bounds, configuration:configuration)
    super.init(coder: aDecoder)
}
```

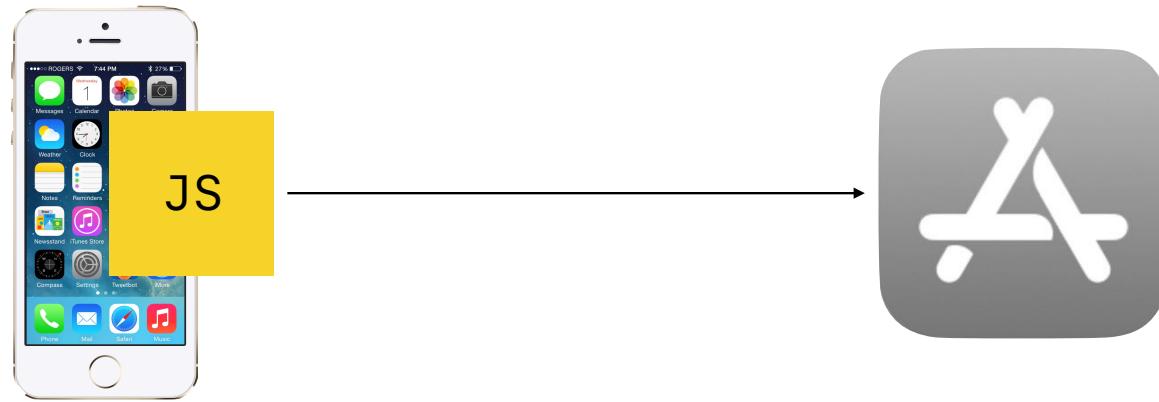
>popup.js

```
document.querySelector('#popup_143').style.display = 'none';
document.querySelector('.app-download-popup').style.display = 'none';
```

# User Scripts

- \* 어떤 용도가 좋은가
  - \* Document 변경하기
  - \* 이벤트 처리
  - \* 리소스 불러오기
  - \* 앱과 통신하기

# Script Messages



- \* sent as JSON
- \* Converted to ObjC/Swift types

# Script Messages

## Script Message Handler 등록

```
configuration.userContentController.add(self, name: "myName")
```

# WKScriptMessageHandler 프로토콜 구현

# Script Messages

message 전달하기

```
webkit.messageHandlers.<name>.postMessage();  
  
function postMyMessage()  
{  
    var message = { 'message': 'order completed', 'price': '13000' };  
    webkit.messageHandlers.myName.postMessage(message);  
}
```

# Script Messages

message 받아서 처리하기

```
var myHandler = MyHandler()

configuration.userContentController.add(myHandler, name: "myName")

class MyHandler: NSObject, WKScriptMessageHandler {
    func userContentController(_ userContentController: WKUserContentController,
                               didReceive message: WKScriptMessage) {
        print("Message from JS: \(message.body)")
    }
}
```

# Script Messages

WKScriptMessage

Name	Type
body	ANY
webView	WKWebView
name	String

# Script Messages

## Main Menu Fetch

```
var myHandler = MyHandler()

configuration.userContentController.add(myHandler, name: "didFetchMainMenu")

//... 핸들러 클래스 예제
class MyHandler: NSObject, WKScriptMessageHandler {
    var mainMenu : Array<Dictionary<String, String>> = []
    func userContentController(_ userContentController: WKUserContentController,
                               didReceive message: WKScriptMessage) {
        print("Message from JS: \(message.body)")
        mainMenu = message.body as! Array<Dictionary<String, String>>
    }
}
```

>fetch.js

```
var menuArray = [];
var menuLinks = document.querySelectorAll('.slide-navi li a')
for (item of menuLinks) {
    var menuItem = {'title':item.textContent, 'urlString':item.href};
    menuArray.push(menuItem);
}
webkit.messageHandlers.didFetchMainMenu.postMessage(menuArray);
```

# Custom UI

## WKUIDelegate

```
weak var uiDelegate: WKUIDelegate?

optional func webView(_ webView: WKWebView, createWebViewWith configuration: WKWebViewConfiguration, for navigationAction: WKNavigationAction, windowFeatures: WKWindowFeatures) -> WKWebView?

optional func webView(_ webView: WKWebView, runJavaScriptAlertPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping () -> Swift.Void)

optional func webView(_ webView: WKWebView, runJavaScriptConfirmPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping (Bool) -> Swift.Void)

optional func webView(_ webView: WKWebView, runTextInputPanelWithPrompt prompt: String, defaultText: String?, initiatedByFrame frame: WKFrameInfo, completionHandler: @escaping (String?) -> Swift.Void)

optional func webViewDidClose(_ webView: WKWebView)
```

# Configurations

WKWebViewConfiguration

Name	Type
applicationNameForUserAgent	String
preferences	WKPreferences
processPool	WKProcessPool
userContentController	WKUserContentController
websiteDataStore	WKWebsiteDataStore

# 웹 앱 개발 #3

Safari View Controller



# 주요 기능

- \* 익숙한 사용자 인터페이스
- \* 사파리와 쿠키 공유
- \* 패스워드 자동완성
- \* 신용카드 자동완성
- \* 주소록 자동완성
- \* 사파리 읽기모드
- \* 콘텐츠 블로킹

# Safari ViewController

SFSafariViewController

```
import SafariServices

let sfVC = SFSafariViewController.init(url: URL(string:"http://apple.com"))
self.present(sfVC, animated: true, completion: nil)
```

# Safari ViewController

iOS 11+

- \* Drag and drop
- \* StatusBar
- \* Privacy
- \* Support Window.open()
- \* Customizing UI Elements
  - \* Dismiss Button, Share Sheet, Bar Collapsing ...

참고자료 <https://developer.apple.com/videos/play/wwdc2017/225/>

# SFAuthenticationSession

iOS 11+

- \* 웹 서비스 로그인 정보 관리
  - \* 쿠키, 웹사이트 데이터
  - \* 사파리 ↔ 앱
- \* 3<sup>rd</sup>-party 인증 프로토콜 (OAuth) 로그인
- \* 기업용 SSO 지원

