

MobileApp 과정 #1

2019.4

박재성

목차

1. 과정 소개

2. Viewport

3. 엘리먼트

4. 이벤트

5. 모바일 환경 이해

6. 디바이스 & OS

7. 모바일 디버깅 환경

1. 과정 소개

과정 소개

-	월	화	수	목	금
강사	박재성	박재성	박재성	손찬욱	손찬욱
8:30	모바일 환경의 이해 <ul style="list-style-type: none">과정 소개기본환경 설정Viewport엘리먼트모바일 이벤트	HTML5 <ul style="list-style-type: none">GeolocationOnline/OfflineNetwork InformationBattery Status API	<ul style="list-style-type: none">React BasicJSXVirtualDOM / SSRComponent & propsPropTypes: 타입 체크	<ul style="list-style-type: none">Redux함수형 프로그래밍 맛보기Redux 사용하기Redux devtool	React 웹앱 구현실습
9:30					영화 후기 관리 시스템 <ul style="list-style-type: none">UI 설계하기개발하기
10:30					
11:30					
	점심				
1:30	<ul style="list-style-type: none">환경 이해기기별 이해모바일 디버깅 환경디버깅 실습	<ul style="list-style-type: none">Media QueryRequest Animation FrameHistory APIDevice Sensors	<ul style="list-style-type: none">StateContext APIComponent LifecycleEvent handlingList & key propsForm handlingReact Router	<ul style="list-style-type: none">Redux 구조 설계하기Redux MiddlewareRedux + React	<ul style="list-style-type: none">React 웹앱 구현실습
2:30					
3:30					
4:30					과정 정리

기본 환경 설정

1) node.js 설치 (v8.x)

→ <https://nodejs.org/dist/latest-v8.x/>

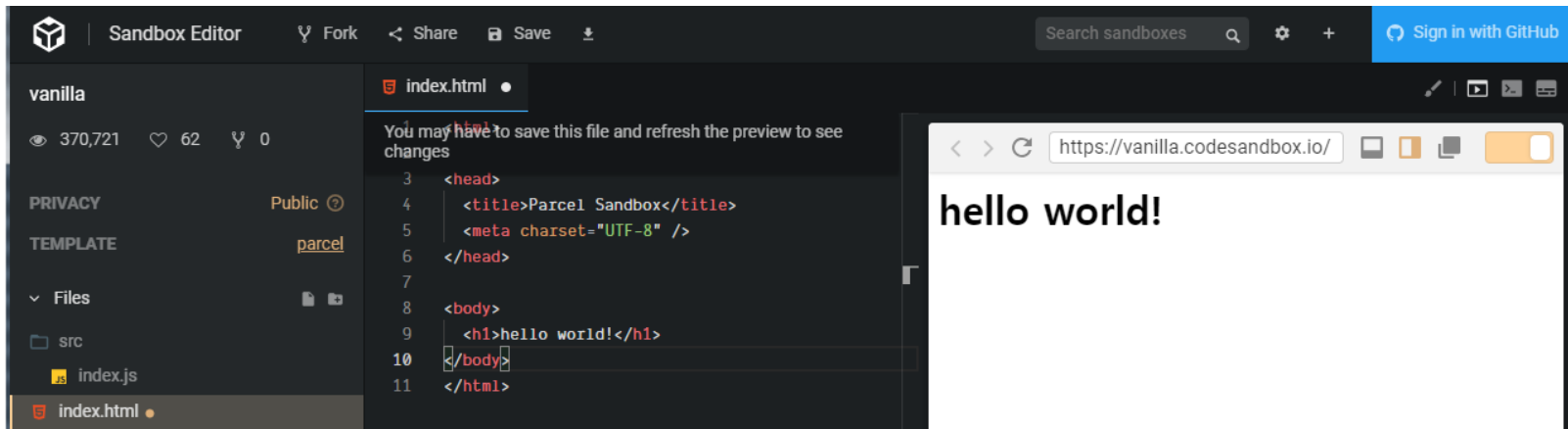
→ 윈도우 환경은 .msi 확장자를 갖는 인스톨러를 통해 설치

2) npm proxy 설정

```
$ npm config set proxy http://10.241.3.7:8080  
$ npm config set https-proxy http://10.241.3.7:8080  
$ npm config set strict-ssl false
```

기본 환경 설정 (작성 코드를 모바일에서 테스트)

1) codesandbox → <https://codesandbox.io/s/vanilla>



2) 코드 수정 후, 상단 메뉴 중 'Save'(또는 ctrl+s)로 저장하면 unique한 URL이 생성

→ 생성된 URL을 모바일 브라우저에서 접근하면 현재 작성한 코드를 확인할 수 있다.

ex) <http://n7xv5nwv6m.codesandbox.io>



실습 파일 다운로드

→ 다운로드 주소는 수업에서 공지

2. Viewport

Viewport

화면에 보이는 가상 크기
브라우저에 따라 기본 값이 다름

<http://viewportsizes.com/>
<http://screensiz.es>

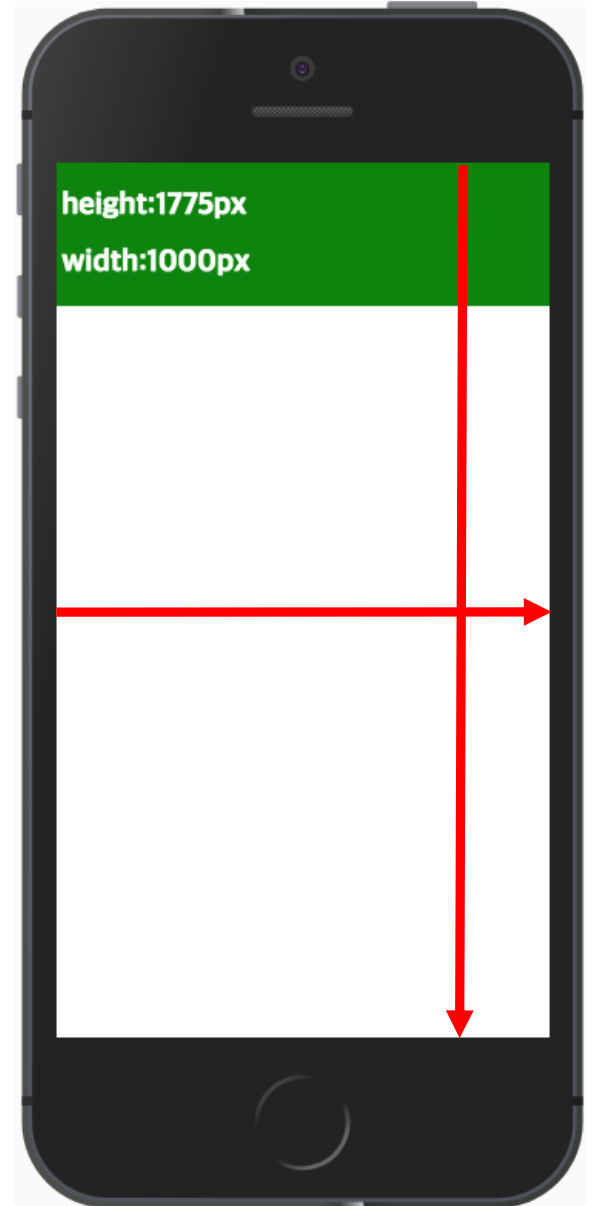
VIEWPORT SIZES ?					
iphone					
Device ▲	Platform	Version	Portrait	Landscape	Release
iPhone	iOS	3.1.3	320	480	2007-06
iPhone 3G	iOS	4.2.1	320	480	2008-07
iPhone 3GS	iOS	6.0a2	320	480	2009-06
iPhone 4	iOS	5.1.1	320	480	2010-06
iPhone 4S	iOS	4.3.5	320	480	2011-10
iPhone 5	iOS	6.0	320	568	2012-09
iPhone 5c	iOS	7.0	320	568	2013-09
iPhone 5s	iOS	7.0	320	568	2013-09
iPhone 6	iOS	8.0	375	667	2014-09
iPhone 6 Plus	iOS	8.0	414	736	2014-09
Unique Counts	1 / 30	8 / 85	3 / 38	4 / 51	8 / 11

1.viewport/1.html

meta tag

메타태그를 이용하여 viewport를 조절할 수 있음.
조건들은 ,(쉼표)로 구분하며 속성 = 값으로 설정.

```
<meta name="viewport" content="....">
```



width

넓이는 지정할 수 있음
width = 300px

기기에 맞는 넓이를 지정하고 싶다면
device-width속성을 사용

```
<meta name="viewport"  
      content="width=device-width">
```



initial-scale

숫자가 들어가며 초기 배율 지정
1을 넣으면 정 배율로 보이게 한다.

```
<meta name="viewport"  
      content="initial-scale=1">
```



xxxx-scale

최대 스케일과 최소 스케일을 지정.

```
<meta name="viewport"  
  content="width=device-width,  
    initial-scale=1,maximum-scale=1,  
    minimum-scale=1">
```



user-scalable

사용자가 스케일을 변경할 수 있는지는 결정.

```
<meta name="viewport"
      content="width=device-width,initial-scale=1,
              maximum-scale=1,minimum-scale=1,
              user-scalable=no">
```

iOS에서 다른점:

- iOS 10+ Safari에서는 'user-scalable=no' 설정하더라도 pinch-to-zoom시 접근성 향상을 위해 스케일 된다.
- iOS Chrome에서는 'maximum-scale=1' 설정 시, 사용자가 스케일링하지 못하도록 만들 수 있다.

1.viewport/5.html



Reference

- 반응형 웹 디자인 기본 사항

<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>

- Using the viewport meta tag to control layout on mobile browsers

https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag

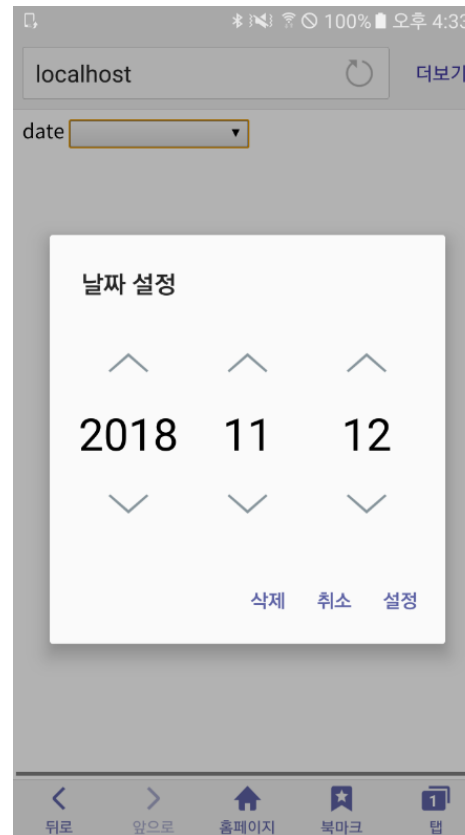
3. 엘리먼트

엘리먼트

- 모바일에서 input 엘리먼트를 효과적으로 활용해야 함.
- input의 타입에 따라 가상 키보드의 모습이 다름.
- 사용자에게 적절한 input의 타입은 좋은 UX을 보여줌.

input[type=date]

날짜를 나타내는 input 엘리먼트



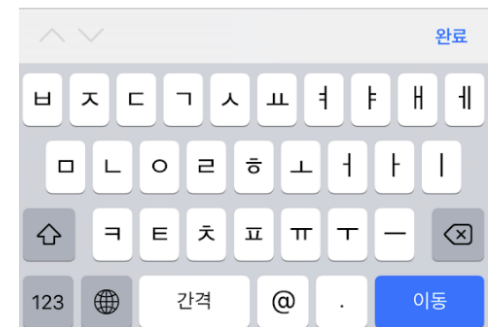
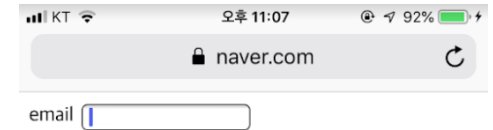
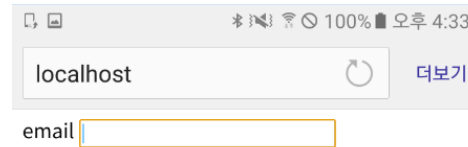
Android 6
(Galaxy S6, Samsung Internet)



iOS 12
iPhone 7

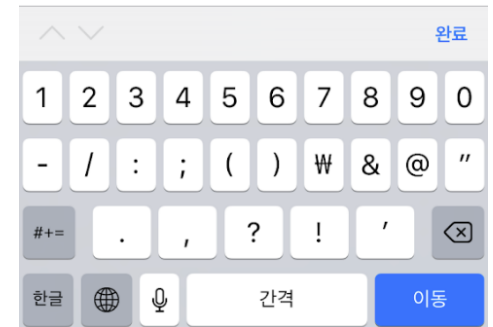
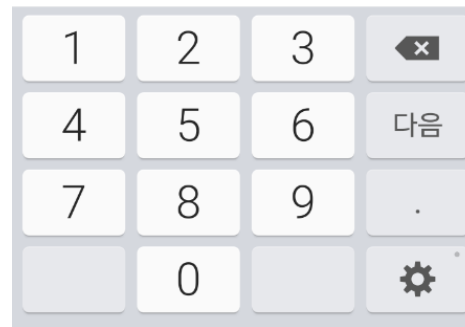
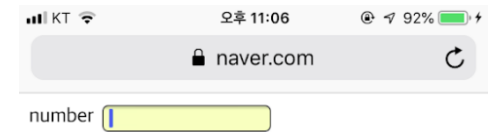
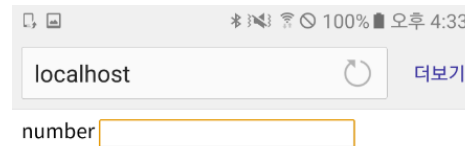
input[type=email]

이메일 필드를 넣을 때
사용하는 input 엘리먼트



input[type=number]

숫자를 넣을 때 사용하는
input 엘리먼트



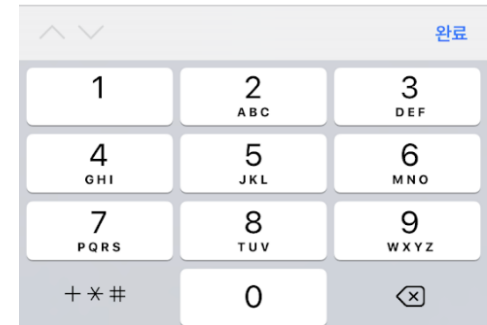
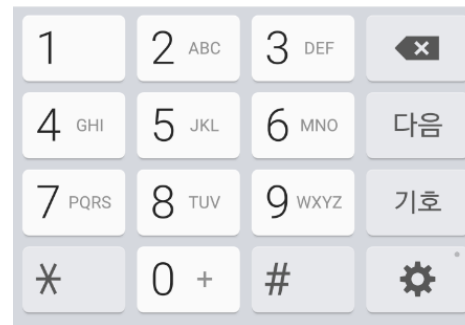
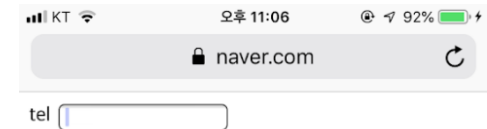
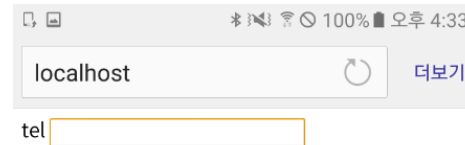
input[type=tel]

전화번호를 넣을 때 사용하는

input 엘리먼트

숫자 키가 팝업되므로, 숫자

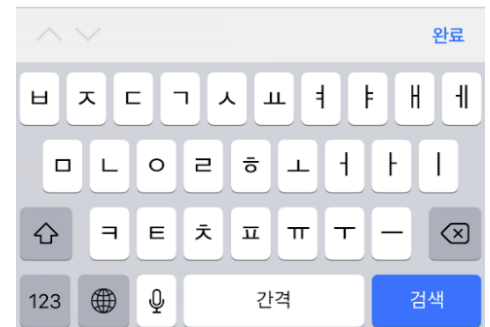
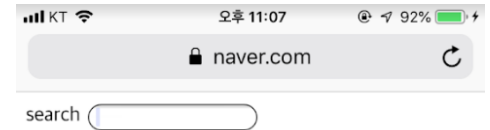
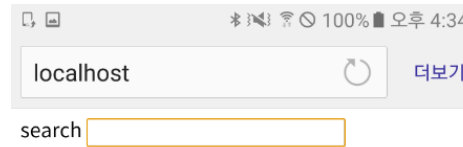
입력만 받을 경우에도 사용가능



input[type=search]

검색할 때 사용하는 엘리먼트

input 요소에 포커싱 되면,
좌측 하단에 '검색' 버튼 노출



실습

앞서 알아본 엘리먼트를 사용하기

4. 이벤트

모바일 이벤트

mouse 관련한 이벤트를 사용하지 않음.
touch 이벤트를 사용.



터치 이벤트

touchstart

touchend

touchmove



멀티 터치



다수의 터치를 알 수 있음.

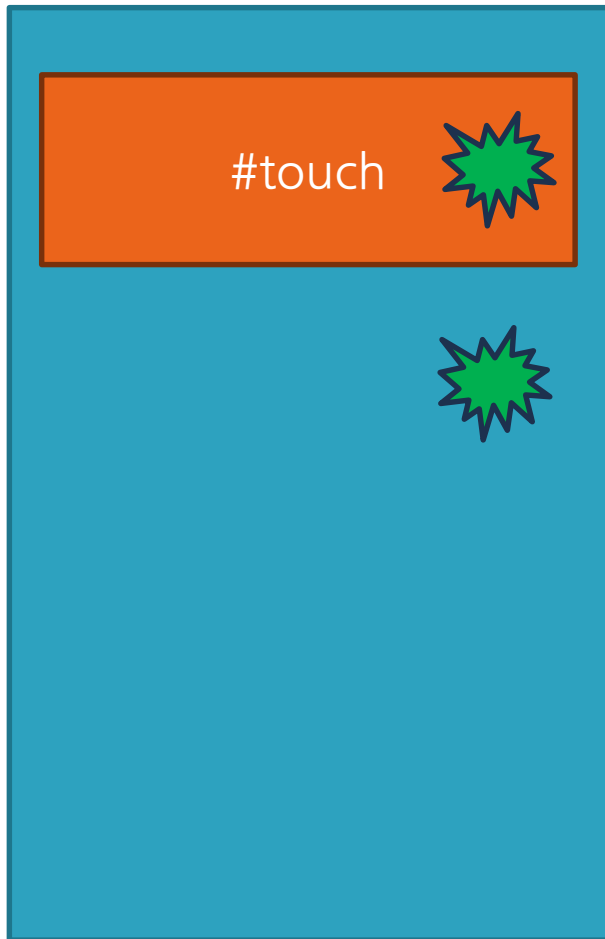
싱글 DOM에서만 가능.

이벤트 객체

```
ele.addEventListener("touchstart",function(event) {  
    // 터치 객체 [touchEvent,touchEvent]  
    event.touches;  
    event.changedTouches;  
    event.targetTouches;  
});
```

- **touches** : 화면에 터치가 발생한 모든 touch 터치 이벤트 객체들
어떤 요소로 부터 터치가 시작/변경 되었는지와는 상관없다.
- **changedTouches** : 터치 목록 객체로, 이벤트에 따라 다른 값을 반환한다.
touchstart: 현재 활성화된 터치 포인트 목록
touchmove: 마지막 이벤트 부터 변경이 발생한 터치 포인트 목록
touchend: 표면(surface)로 부터 제거된 터치 포인트 목록 (즉, 표면을 더이상 터치하지 않는 손가락의 터치)
- **targetTouches** : 대상 요소 표면을 터치 중인 터치 목록
touchstart가 발생한 동일 요소인 경우에 한한다.
end 시점에 없어진다.

이벤트 객체



touchstart

```
event.touches; //2  
event.changedTouches; //1  
event.targetTouches; //1
```

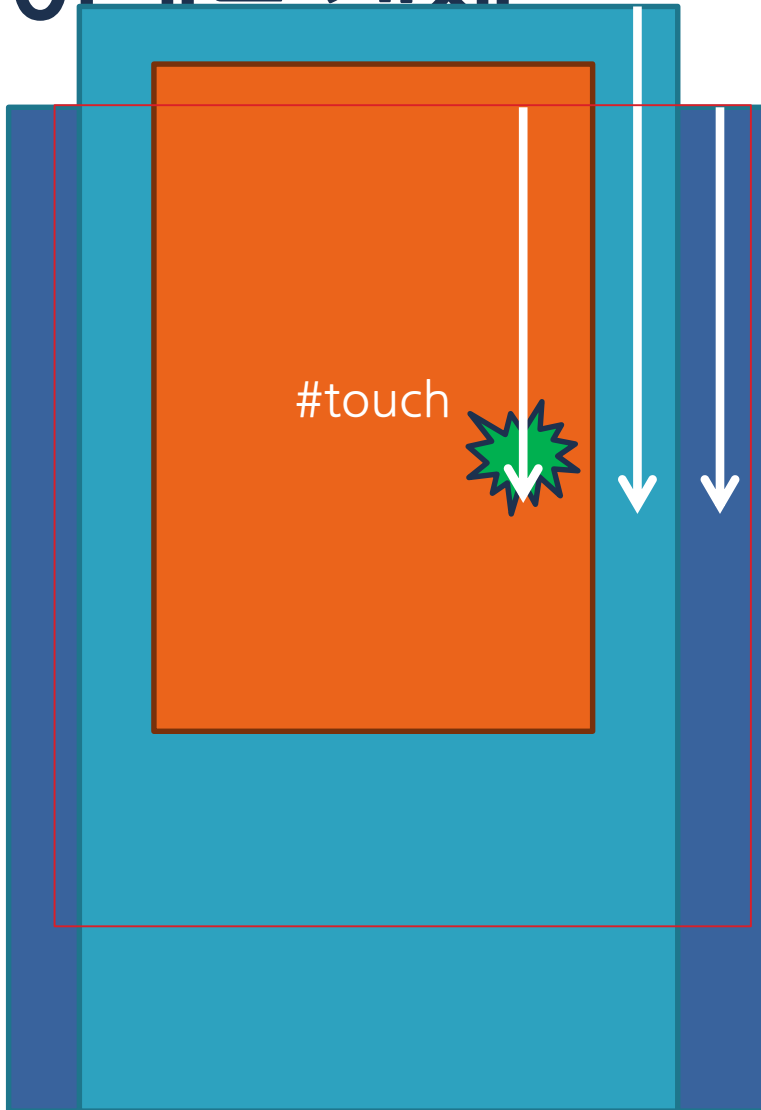
touchmove

```
event.touches; //2  
event.changedTouches; //1  
event.targetTouches; //1
```

touchend

```
event.touches; //0  
event.changedTouches; //1  
event.targetTouches; //0
```

이벤트 개체



속성	설명
identifier	이벤트의 유일한 값
target	터치한 엘리먼트
screenX	디바이스 화면에서 X 좌표
screenY	디바이스 화면에서 Y 좌표
clientX	브라우저 화면에서 X 좌표
clientY	브라우저 화면에서 Y 좌표
pageX	전체 페이지에서 X 좌표
pageY	전체 페이지에서 Y 좌표

실습

터치 이벤트를 이용한 간단한 실습

5. 모바일 환경 이해

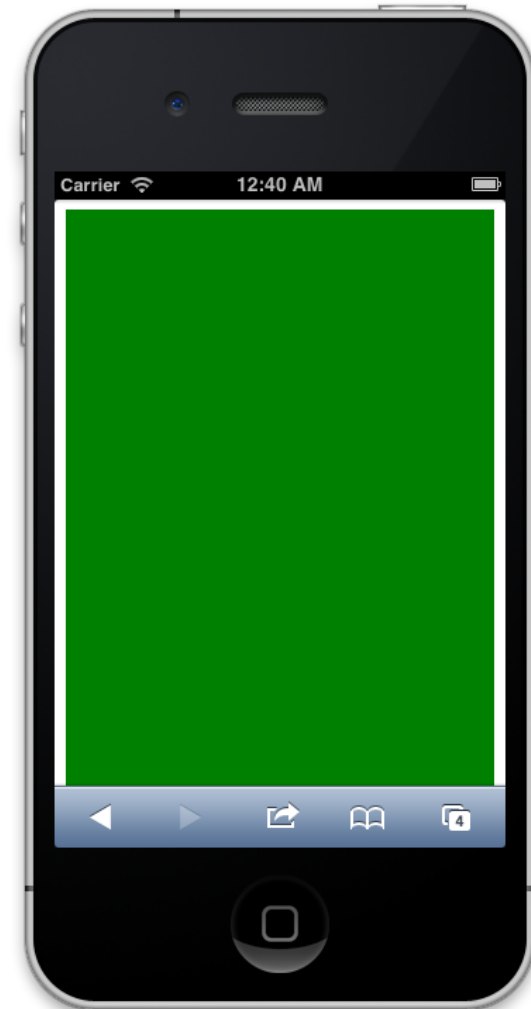
App Style#1

주소창 없애기

사용자 동작에 응답하여 브라우저에 전체 화면 요청

<https://developers.google.com/web/fundamentals/native-hardware/fullscreen/>

4.environment/1.html



Fullscreen API

- 1) 사용자 동작에 응답하여 브라우저에 전체 화면 요청
→ iOS 제외(12+ iPad only) 모든 브라우저에서 사용 가능

	Type	설명
el.requestFullscreen()	메서드	전체 화면 모드 설정
document.exitFullscreen() / FF는 cancelFullscreen()	메서드	전체 화면 모드 취소
document.fullscreenElement	속성	전체 화면 상태의 요소

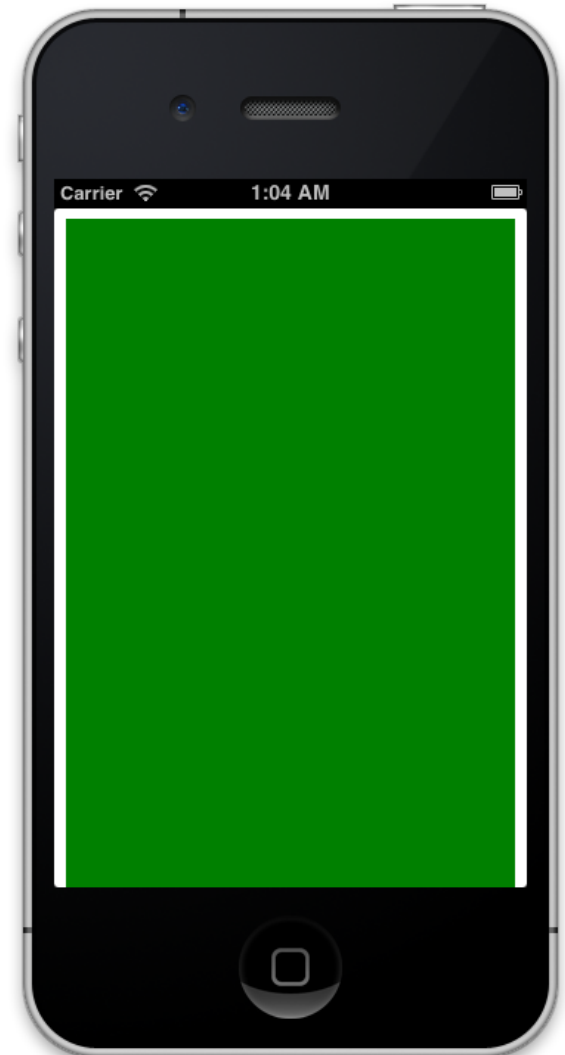
```
document.querySelector("#some").addEventListener("click", function() {  
    if (document.fullscreenElement) {  
        document.cancelFullscreen();  
    } else { document.documentElement.requestFullscreen(); }  
}, false);
```

→ https://developer.mozilla.org/en-US/docs/Web/API/Fullscreen_API

App Style#2

- 홈 화면에 추가한 경우 사용 가능한 메타 태그

```
<!-- iOS -->  
<meta name="apple-mobile-web-app-capable" content="yes">  
  
<!-- Android -->  
<meta name="mobile-web-app-capable" content="yes">
```

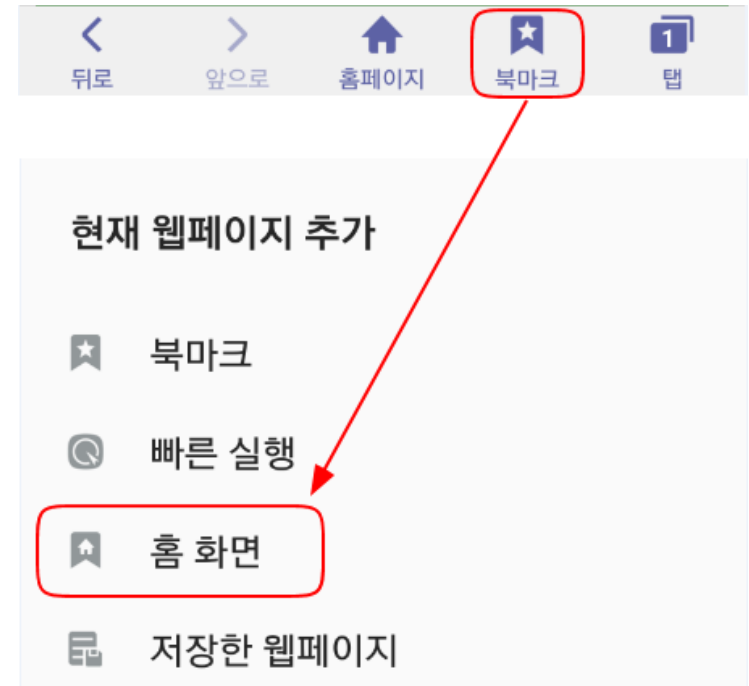


홈 화면에 앱 설치: Android

meta 태그를 설정한 후, 애플리케이션 바로가기 아이콘을 기기 홈 화면에 추가하고 호출 시 전체 화면으로 호출된다.

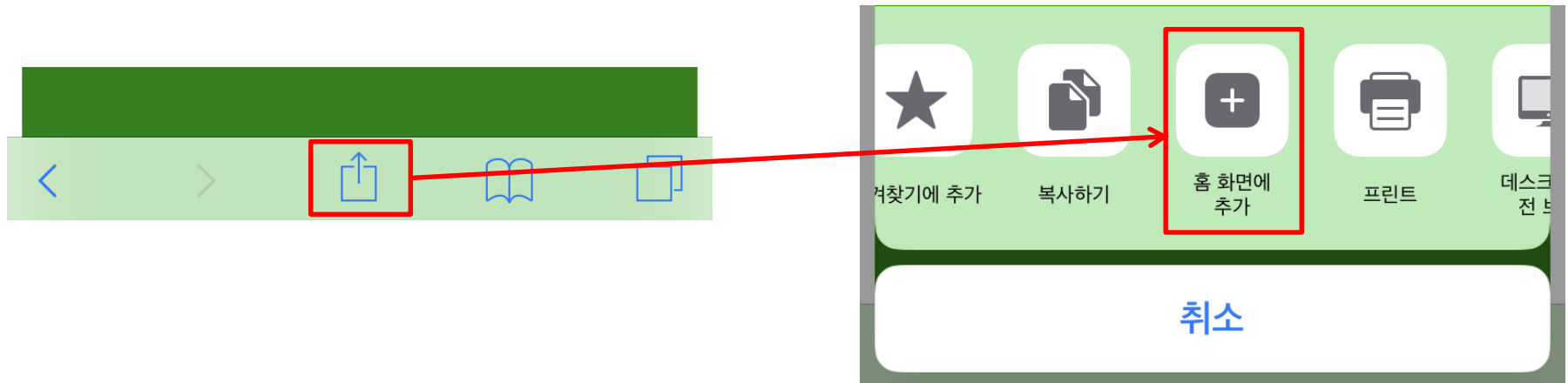
Android의 경우:

- 브라우저 하단, 북마크를 누른채 기다린다.
- '홈 화면' 추가를 선택



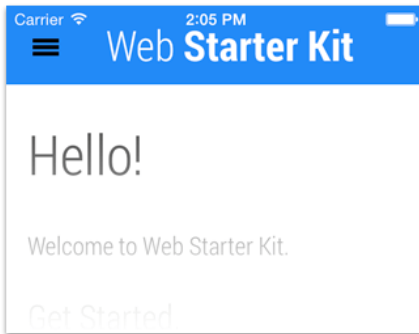
홈 화면에 앱 설치: iOS Safari

- 1) Safari 하단의 '추가하기' 버튼 선택
- 2) '홈 화면에 추가' 선택

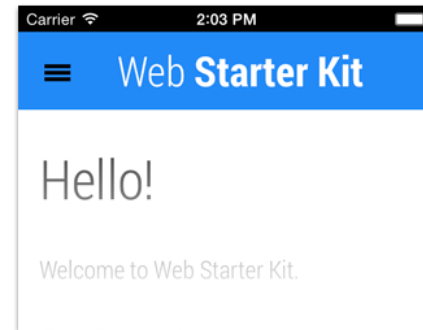


상태바 스타일 & 테마 색상

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

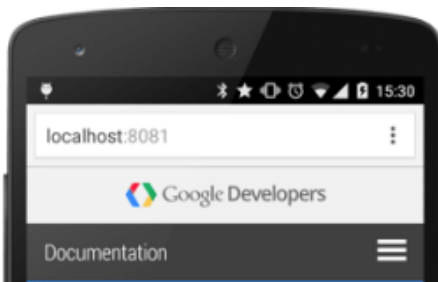


black-translucent



black

```
<!-- 메타 테마 색상: Chrome, Firefox OS, Opera -->  
<meta name="theme-color" content="#4285f4">
```

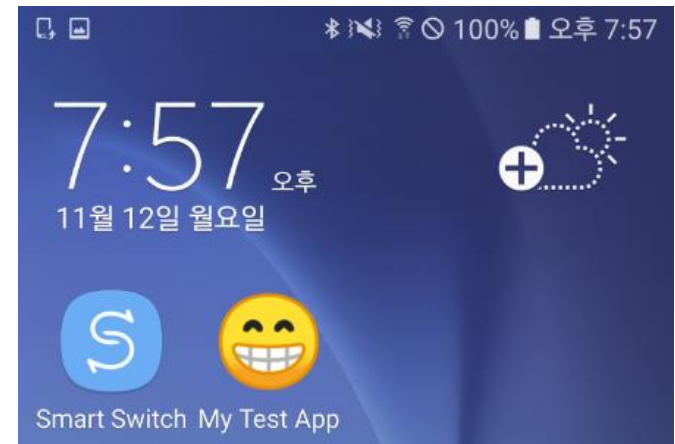


App Style#3 - Web App Manifest

웹앱의 시작 정보와 아이콘 등을 담고 있는
JSON 형태의 설정파일

```
<link rel="manifest" href="/manifest.json" />
```

```
{
  "short_name": "My Test App",
  "name": "Mobile Web Application Test App",
  "icons": [
    {
      "src": "grin.png",
      "sizes": "128x128",
      "type": "image/png"
    }
  ],
  "start_url": "2.html",
  "display": "standalone",
  "orientation": "landscape"
}
```

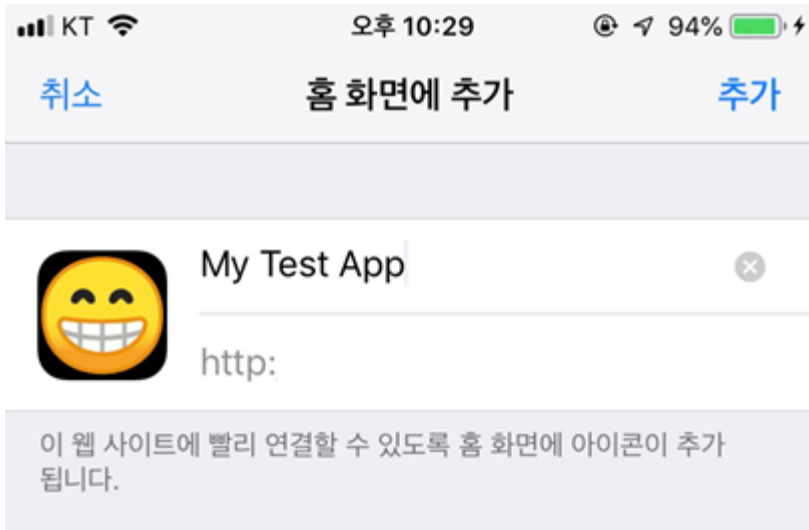


→ <https://developers.google.com/web/fundamentals/web-app-manifest/>

iOS에서의 아이콘

iOS의 경우, 아이콘을 적용하기 위해선 meta 태그를 통해 지정해야 한다.

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



Event Delegation

엘리먼트에 이벤트를 직접 할당하는 것이 아니라 **상위 엘리먼트에 위임**.

Event Delegation

```
<ul id="parent">  
  <li class="odd">1</li>  
  <li>2</li>  
  <li class="odd">3</li>  
  <li>4</li>  
</ul>
```

```
const aOdd = document.querySelectorAll(".odd");  
  
for(const i = 0, l = aOdd.length; i < l ;i++){  
  aOdd[i].addEventListener("click", function() {  
    alert("click");  
  })  
}
```

Event Delegation

```
<ul id="parent">  
  <li class="odd">1</li>  
  <li>2</li>  
  <li class="odd">3</li>  
  <li>4</li>  
</ul>
```

```
const ul = document.getElementById("parent");  
  
ul.addEventListener("click",function(e) {  
  if(e.target.classList.contains("odd")) {  
    alert("click");  
  }  
});
```

모바일에서 문제점

ul#parent

li

`nhn`



li

`naver`



li

`hangame`

해결 방법

```
// Safari 또는 chrome에서만 사용가능
.delegate {
    -webkit-tap-highlight-color: rgba(0,0,0,0);
}

.delegate .highlight{
    -webkit-tap-highlight-color: rgba(0,0,0,0.25);
}
```

- 링크 터치시 하이라이팅 색을 변경
- 마지막 값은 alpha로 투명도를 설정하며, 0으로 설정시 투명하게 처리되므로 링크를 터치하더라도 하이라이팅 되지 않는것처럼 처리할 수 있음

→ <https://developers.google.com/web/fundamentals/design-and-ux/input/touch/?hl=ko>

실습

delegate 사용해보기.

Page Cache (Back-Forward Cache)

페이지가 이동되고 다시 돌아왔을 때 캐시 되는 현상

페이지 캐시가 안되면 리소스를 다시 가져오거나 복잡하게 구현해야 한다.

지원 브라우저:

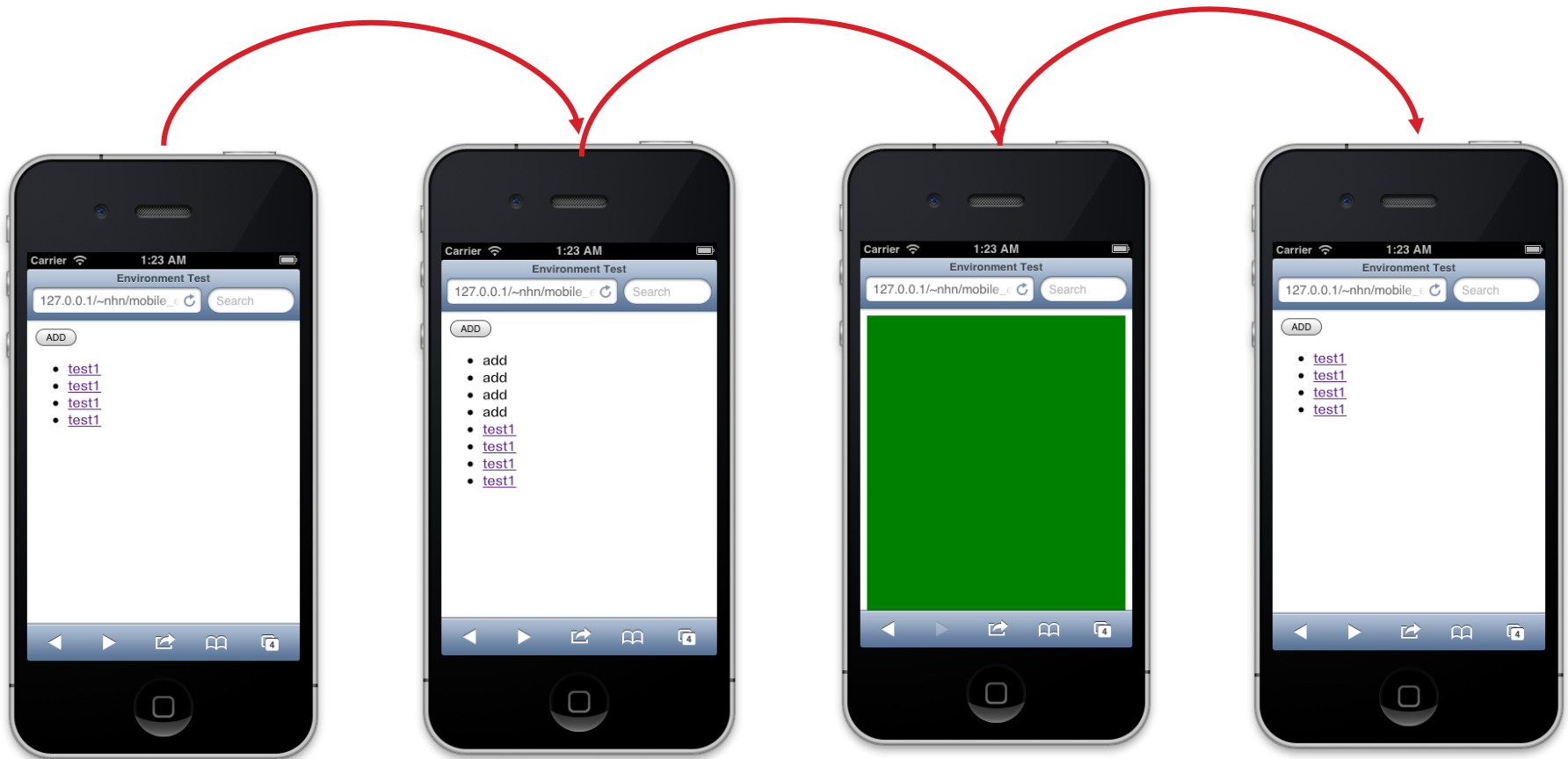
- iOS 기본브라우저(4.x+)
- Firefox 최신버전

미지원 브라우저:

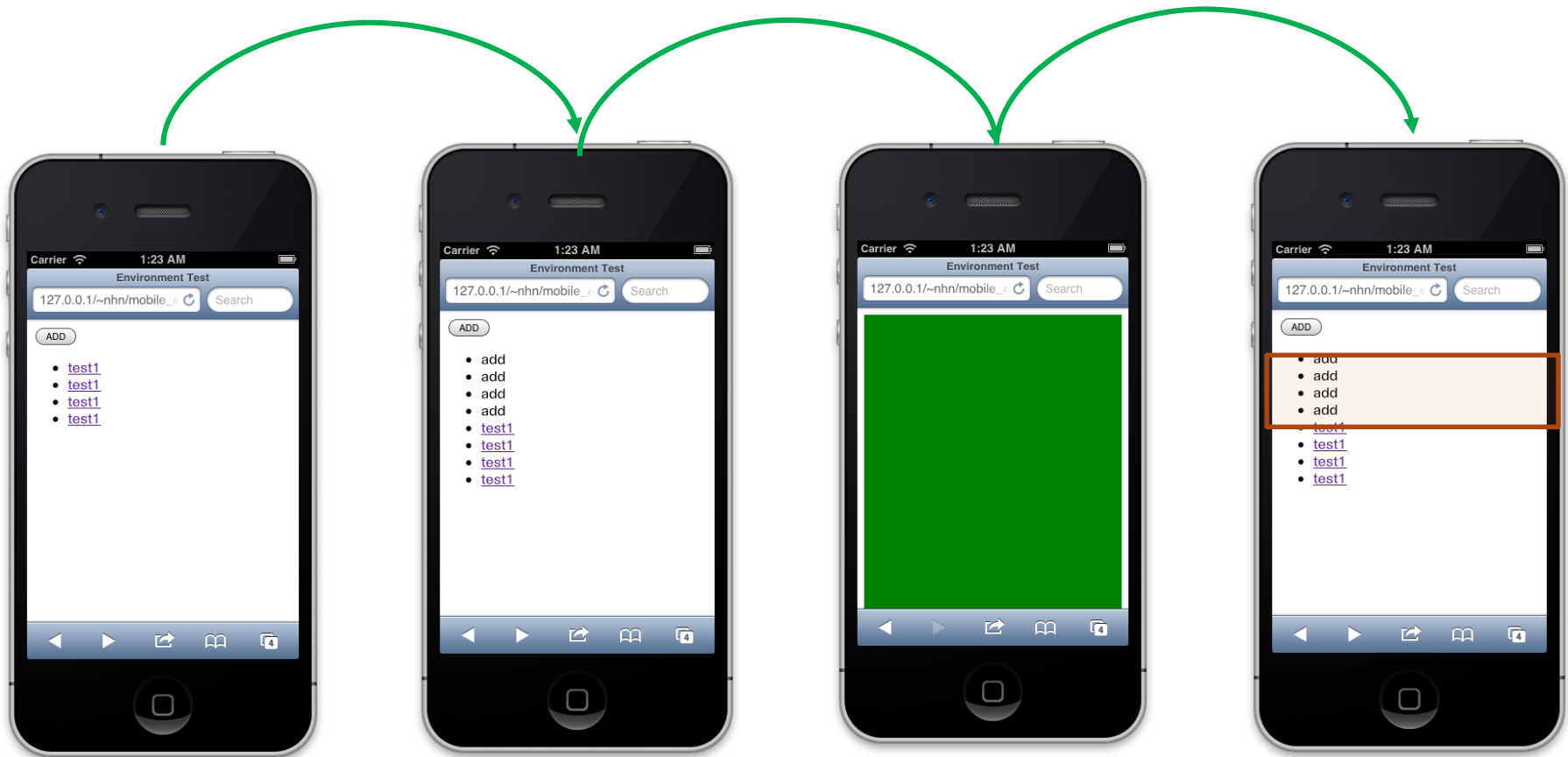
- Android 기본브라우저(4.4+)

→ https://developer.mozilla.org/en-US/docs/Archive/Misc_top_level/Working_with_BFCache

Page Cache



Page Cache



캐싱 방지 방법

아래의 방법을 사용하면 page cache 되지 않도록 할수 있다.

- onunload, beforeunload 이벤트를 사용한 경우
- "cache-control: no-store" 설정한 경우
- HTTPS 이면서 아래 사항 중 하나 이상인 경우:
 - "Cache-Control: no-cache"
 - "Pragma: no-cache"
 - "Expires: 0" 또는 "Expires" 헤더 만료 경우
 - "Cache-Control: max-age=" 또한 사용되지 않은 경우
- 페이지가 완전히 로드 되기 전에 페이지 이동을 한 경우
- iframe을 사용한 경우
- navigator.geolocation을 사용할 경우

실습

Page Cache 경험해보기.

6. 디바이스 & OS

디바이스 & OS

기기와 OS에 따라 차이를 가지고 있다.

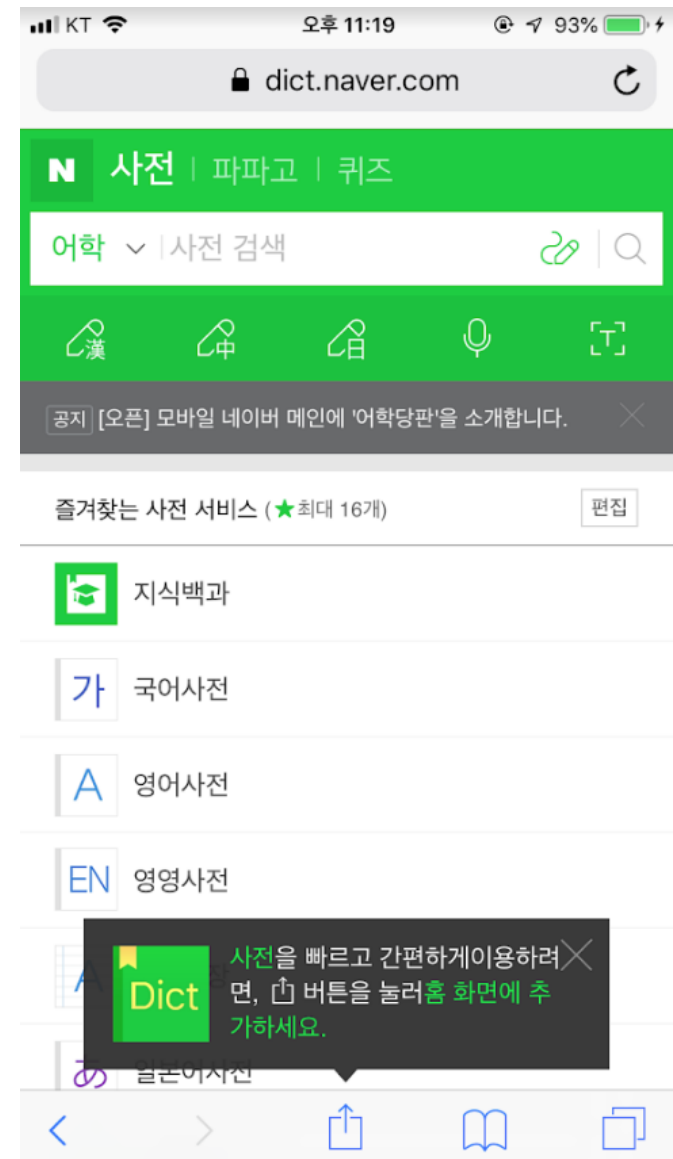
- iOS는 OS의 파편화와 최신 OS로의 업데이트가 비교적 빠르게 이뤄지기 때문에 iOS 제품군에 대한 차이는 다소 적다.
- Android는 제조사와 모델, 그리고 OS 버전에 따라 다르게 동작될 수 있는 상황이 많이 발생된다.

고정 레이아웃

- 고정 레이아웃은 페이지의 스크롤과는 상관없이 현재 위치(상대적인)에 고정되게 한다.
- 현재 모든 모던 브라우저에서 지원된다.

iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet
		2.1-2.3							
3.2-4.3		3							
5-7.1		4-4.3							
8-11.4		4.4-4.4.4	7	12-12.1			10		4-6.2
12	all	67	10	46	69	62	11	11.8	7.2

5.device /1.html



이벤트 흐름

버전/기기 별 이벤트 흐름이 다름.

이벤트의 흐름을 이해해야 프로그래밍하는데 유용함.

touch



터치하고 바로 떼었을 때 이벤트 발생 순서

Android 7 (Galaxy S8)

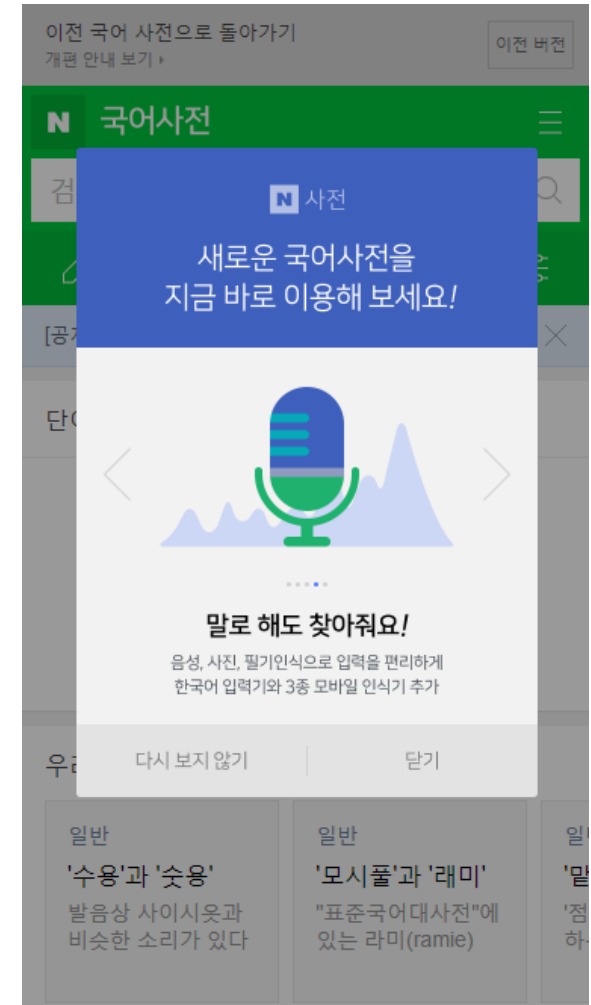
touchstart → touchend → mouseover → mousemove → mousedown
→ mouseup → click → mouseout (포커스 이동되면 발생)

iOS 11

touchstart → touchend → mouseover → mousemove
→ mouseout (포커스 이동되면 발생)

Dimmed 레이아웃

- 정보 레이어를 출력하면서, 배경은 흐릿하게 처리해 정보를 집중해 소비할 수 있도록 하는 UI 구성
- 사용자에게 필수적으로 전달하고자 하는 내용을 표시한다.
- 딤드 레이어는 뷰포트 전체를 덮는 레이어로 구성된다.



Dimmed 레이아웃

문서의 스크롤을 막기 위해선 이벤트의 기본 동작을 막도록 처리한다.

이후, 스크롤을 다시 허용하기 위해선 이벤트를 꼭 제거해야 한다.

```
// 문서의 스크롤을 막는다.
```

```
document.body.addEventListener("touchstart", function(e) {  
    e.preventDefault();  
}, false);
```

```
// passive 이벤트를 지원하는 경우
```

```
document.body.addEventListener("touchstart", handler, { passive: false });
```

Passive 이벤트?

→ <https://developers.google.com/web/tools/lighthouse/audits/passive-event-listeners?hl=ko>

5.device /3.html

7. 모바일 디버깅 환경

모바일 디버깅

모든 상황에 적합한 디버깅은 없음.

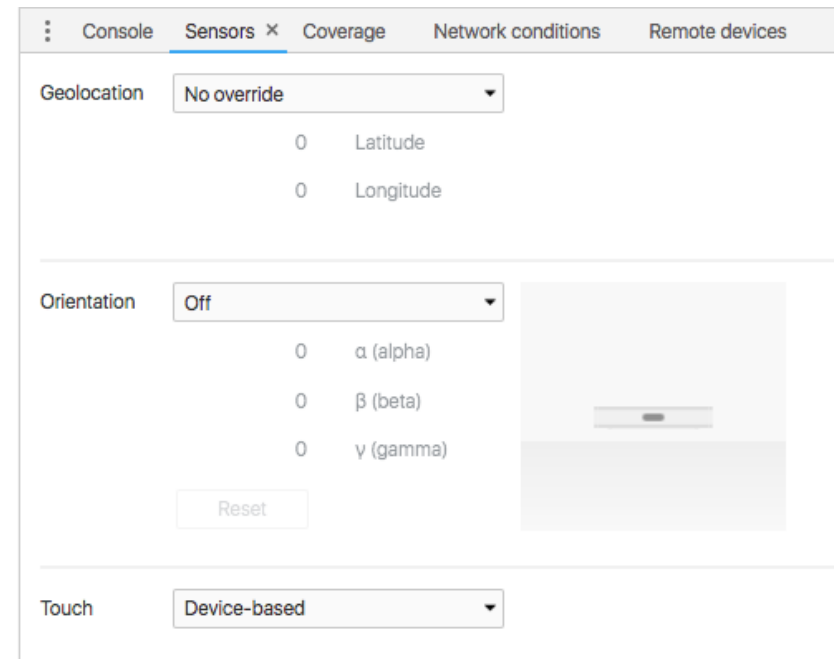
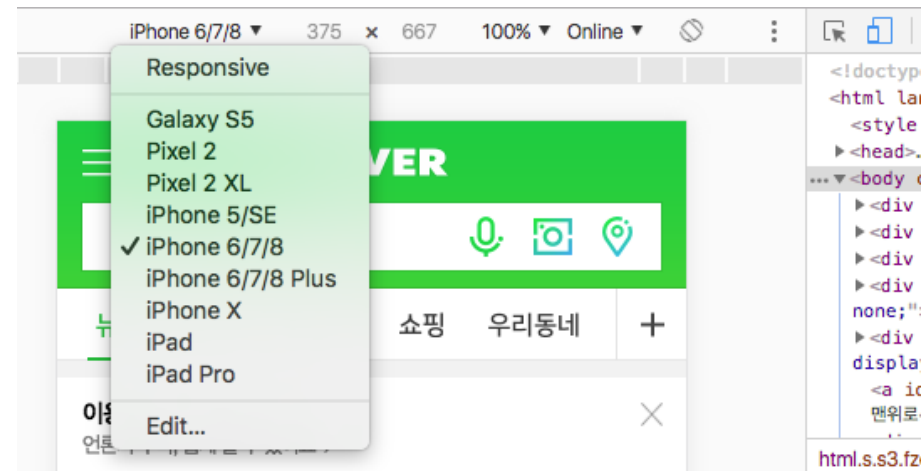
상황에 맞는 디버깅 도구를 사용하는 것이 핵심.

크롬 개발자 도구

일반적으로 모바일 디버깅 도구는 좋지 않다.

크롬 개발자 도구를 통해 기본적인 모바일 스크린과 일부 sensors를 에뮬레이팅 할수 있다.

그러나, 실제 모바일 기기와는 다르므로 기본적인 테스트 용도로 확인 후, 실제 기기에서 테스트가 필요하다.



크롬 Remote Debugging

Android 기기의 경우, 모바일 화면을 직접 확인하면서 디버깅할 수 있다.

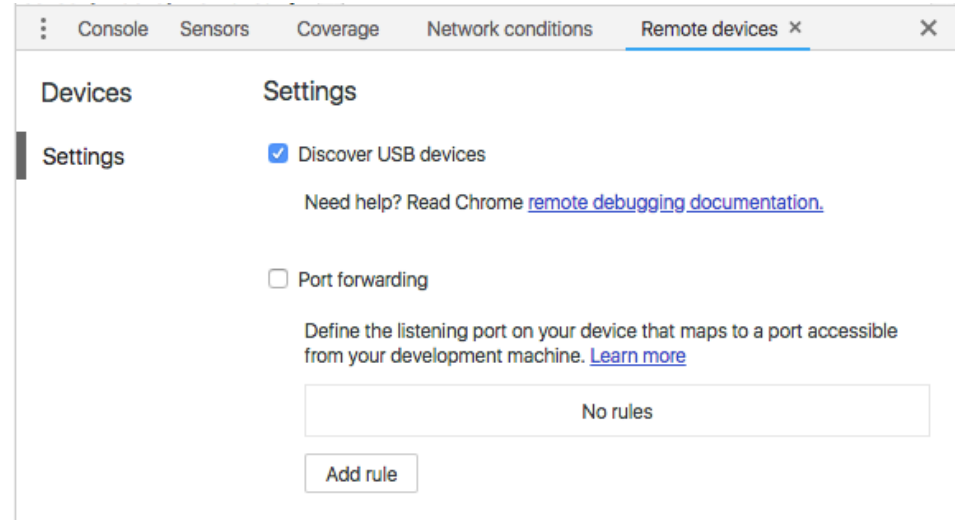
MacOS의 경우, 별도 드라이버 설치 없이 인식되지만, Windows 환경의 경우 제조사의 드라이버를 설치해야 한다.

Universal ADB Driver

→ <http://adbdriver.com/downloads/>

삼성 USB 드라이버

→ <https://developer.samsung.com/galaxy/others/android-usb-driver-for-windows>



<https://developers.google.com/web/tools/chrome-devtools/remote-debugging/>

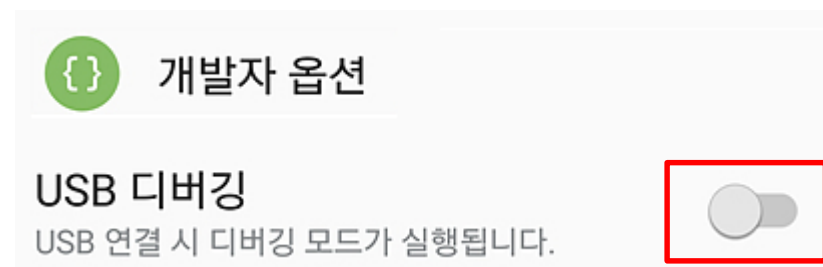
디버깅 모드 활성화#1

Android 4.2 이상에서의 개발자 메뉴 활성화

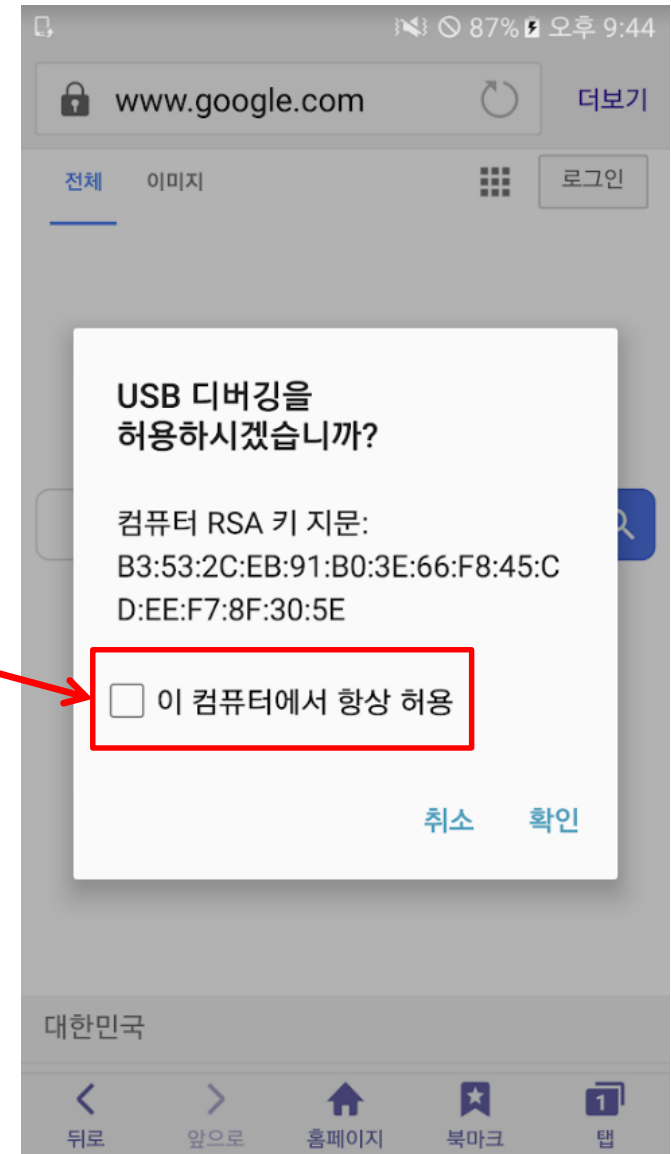
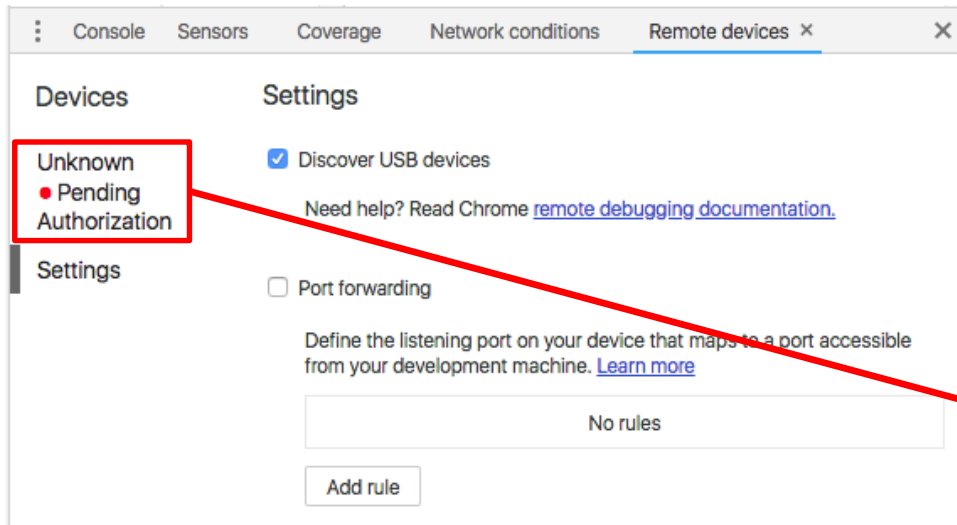
- 1) '개발자 옵션' 메뉴 활성화 방법
설정 > 디바이스 정보 > 소프트웨어 정보 >
빌드번호 영역 **7회 연속터치**



- 2) 설정 > 개발자 옵션 > USB 디버깅 활성화
→ 보안 설정상의 이슈로 인해 "USB 디버깅"이 노출되지만 활성화 되지 않게
막혀 있을 수도 있다.



디버깅 모드 활성화#2



데스크탑 크롬 개발도구 활용

The image shows the Chrome DevTools interface with the Naver mobile site loaded. The left pane displays the mobile site's layout, including the Naver logo, navigation tabs (News, Entertainment, Sports, Shopping, My Community), and a section for popular news articles. The middle pane shows the 'Elements' tab with the DOM tree, highlighting the 'body' element. The right pane shows the 'Styles' tab with the computed styles for the 'body' element, including font-size, line-height, position, background-color, and color. A red arrow points from the 'Inspect' button in the Remote devices panel to the 'body' element in the Elements tab.

DevTools - m.naver.com/?ref=sm

m.naver.com/?ref=sm

Elements Network Sources > > >

```
<div class="section_panelauthor" id="MM_PANEL_EDITOR_AREA" style="display:none;">...</div>
<div id="contents_footer" class="footer_wrap" style="position: relative; display: block;">...</div>
  <a id="btn_top" href="#TOP" class="imh im_top" style="opacity: 0; display: none; zoom: 1;">맨위로</a>
  <div id="set_start_layer" class="ly_favortab" style="display: none;">...</div>
  <div id="ly_beauty_event" class="ly_beauty_event" style="display: none;">...</div>
```

Styles Computed >>

Filter +, -, <, >

element.style { }

.a_w_g181108a.css:1 ip6 body, .fzoom body { font-size: 16px; line-height: 18px; }

bo_w_g181108a.css:1 dy { position: relative; background-color: #f2f2f2; color: #000; }

html.a.and4.fzoom body

Console Search Emulation Rendering

<top frame> < > Preserve log

Devices

SM-G920S #02157df238d64a2c

com.sec.android.app.sbrowser (44.0.2403.133)

NAVER

<https://m.naver.com/?ref=sm>

Inspect

Port Forwarding

USB로 연결된 데스크톱 컴퓨터의 로컬 서버를 모바일 디바이스에서 접근해 브라우징 및 디버깅 할수 있는 방법

☒ Port forwarding

Define the listening port on your device that maps to a port accessible from your development machine. [Learn more](#)

Device port (3333) Local address (dev.example.corp:3333)

Add Cancel

Add rule

Android 기기에서 액세스
할 localhost 포트 번호

개발용 컴퓨터 웹 서버에
서 실행 중인 사이트의 IP
주소 또는 호스트 이름과
포트 번호

Devices

Nexus 5

● Connected

Settings

Nexus 5 #08ae8c2700f43a61

Port Forwarding: ● :5000

Chrome (49.0.2623.105)

New tab: Enter URL

Open

→ <https://developers.google.com/web/tools/chrome-devtools/remote-debugging/local-server>

iOS6 이상 + Mountain Lion

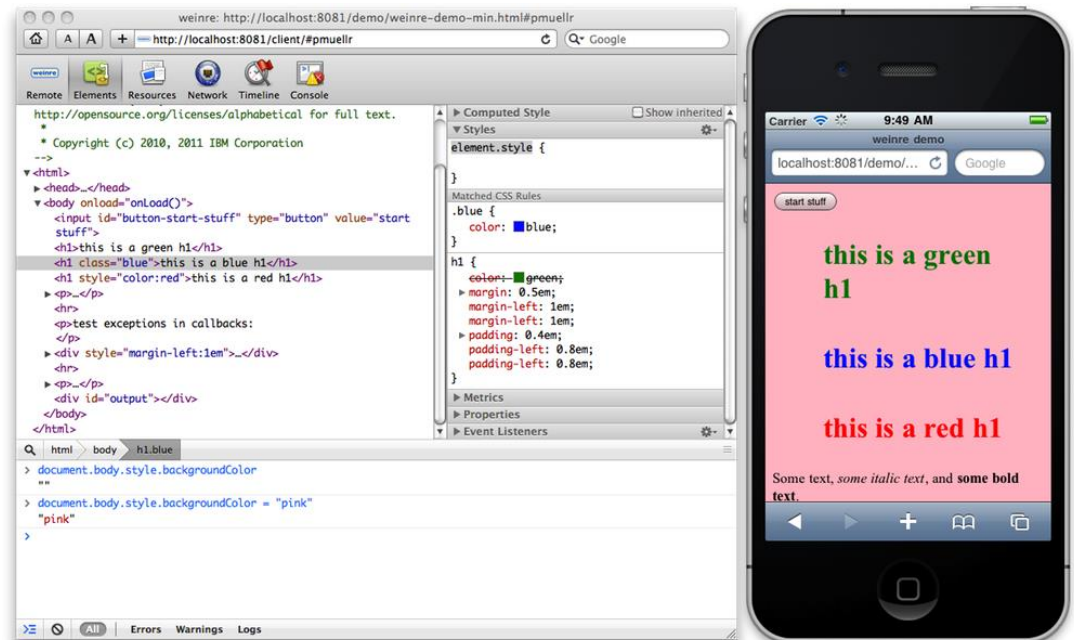
The diagram illustrates the process of enabling WebKit in Safari on an iPhone and connecting it to a computer for development. It consists of three main parts:

- iPhone Settings:** A screenshot of the iPhone's Settings app, showing the 'Safari' settings. The 'Web Content' (웹 속성) toggle is highlighted with a red box and is turned on. Below it, a text box explains that to use WebKit, users need to connect their iPhone's Safari to a computer's Safari via a USB cable.
- Mac Developer Menu:** A screenshot of the Mac's 'Developer' menu. The 'Enable WebKit in Safari' (웹 속성 활성화) option is highlighted, which is the step that triggers the connection to the iPhone.
- Mac Safari Browser:** A screenshot of the Mac's Safari browser window. The address bar shows 'm.naver.com'. The 'Developer' menu is open, and the 'Web Content' (웹 속성) option is selected, which is the step that enables WebKit in the browser.

설정 > Safari > 고급 > 웹속성 활성화

weinre

- **Wenire** <http://people.apache.org/~pmuellr/weinre/docs/latest/Home.html>
- 데스크탑의 일부 기능을 제공하는 디버거
- 엘리먼트 디버깅 / 콘솔 기능 사용
- 리소스/네트워크/타임라인 탭의 일부 기능



weinre 설치#1

- node.js 기반의 프로그램.
- npm으로 설치
 - `sudo npm -g install weinre`
- 서버 실행
 - `weinre --boundHost 1.1.1.1 --httpPort 8080`
 - `weinre --boundHost -all- --httpPort 8080`

wenire 설치#2

1.1.1.1:8080

weinre - web inspector remote

Access Points

debug client user interface: <http://1.1.1.1:8080/client/#anonymous>
documentation: <http://1.1.1.1:8080/doc/>

Target Demos

The following links point to an already instrumented sample application, run in a couple of different environmental conditions.

First open a new browser window for the debug client user interface, as specified above. Then open another new browser window for one of

weinre 설치#3

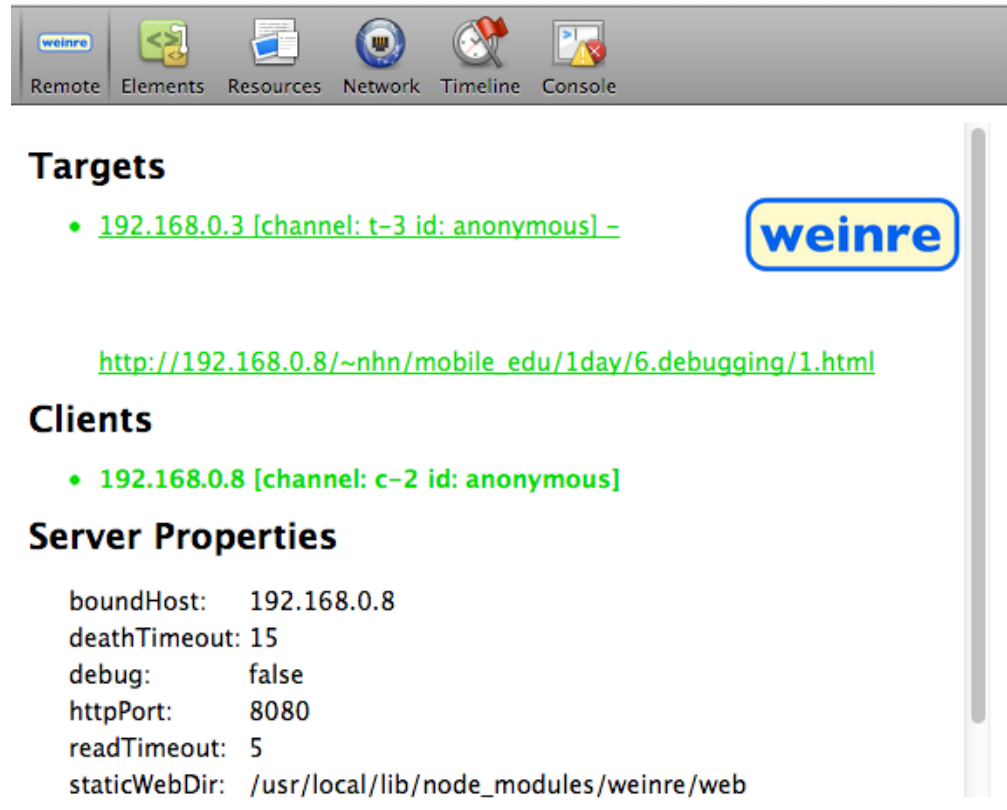
테스트할 페이지에 스크립트 삽입

```
<script src="http://1.1.1.1:8080/target/target-script-min.js#anonymous"></script>
```

weinre 설치#4

UI에 접근하여 디버깅

→ http://weinre_서버_주소:포트/client/



The screenshot shows the Weinre web interface. At the top is a toolbar with icons for Remote, Elements, Resources, Network, Timeline, and Console. Below the toolbar is a 'Targets' section with a single entry: [192.168.0.3 \[channel: t-3 id: anonymous\]](http://192.168.0.3/[channel: t-3 id: anonymous]). To the right of this entry is a 'weinre' logo. Below the 'Targets' section is a 'Clients' section with a single entry: [192.168.0.8 \[channel: c-2 id: anonymous\]](http://192.168.0.8/[channel: c-2 id: anonymous]). At the bottom is a 'Server Properties' section with the following details:

- boundHost: 192.168.0.8
- deathTimeout: 15
- debug: false
- httpPort: 8080
- readTimeout: 5
- staticWebDir: /usr/local/lib/node_modules/weinre/web

wenire

– 장점

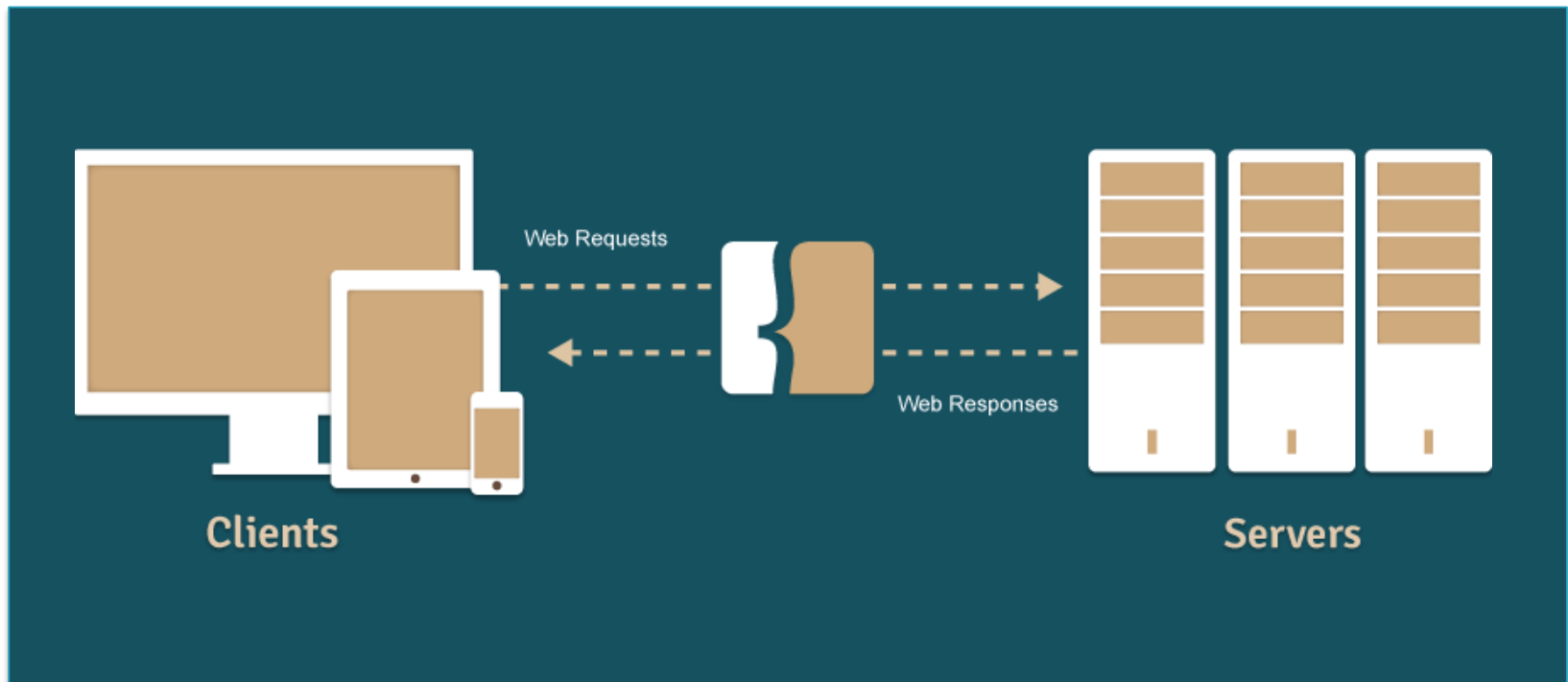
- 데스크탑과 유사한 기능을 제공
- 환경(Android/iOS 등) 구분없이 사용 가능

– 단점

- 서버를 실행해야 하는 불편
- 아주 단순한 스크립트 디버깅만 가능

Fiddler?

웹 디버깅 프록시 <http://fiddler2.com>



Fiddler

AutoResponse기능

- 모바일에서 데스크탑을 proxy서버로 설정하여 디버깅하는 기능

Fiddler 4.6.1.5+에서 https 트래픽이 캡처 되지 않을 때:

- Click Tools > Fiddler Options > HTTPS 탭 선택
- 'Certificates generated by CertEnroll engine'으로 표시되는지 확인
- 'Actions' 버튼 클릭 > Reset Certificates 선택
- 이후 출력되는 모든 대화 창에서 Accept를 선택

고맙습니다.
