

# 모바일 터치 이벤트

지난 번에는 모바일 화면 구성만  
공부했었잖아요?

오늘은 모바일 터치 이벤트에 대해  
알아봅시다!

# 터치 이벤트의 종류

**touchstart**      스크린에 손가락이 닿을 때 발생한다

**touchmove**      스크린에 손가락이 닿은 채로 움직일 때 발생한다

**touchend**      스크린에서 손가락을 뗄 때 발생한다

**touchcancel**      시스템에서 이벤트를 취소시킬 때 발생한다.

터치를 취소한다는 것에 대한 표준이 정의되지 않아  
각 브라우저마다 다르게 발생하기 때문에  
touchend 이벤트로 간주해도 무방하다.

# 터치 이벤트와 마우스 이벤트의 차이

1. 터치 이벤트는 **멀티 터치**가 가능하다.

그래서 2개 이상의 개별 터치에 대한 정보를 모두 얻을 수 있다.

2. 마우스는 포인터의 끝이 있어서 정확한 좌표 값을 얻을 수 있지만

터치 이벤트는 손가락 접촉면적이 넓기 때문에  
**접촉면의 평균 좌표 값을** 이용한다.

3. 터치 이벤트는 **Mouse over** 기능이 없다.

# 터치 인터페이스

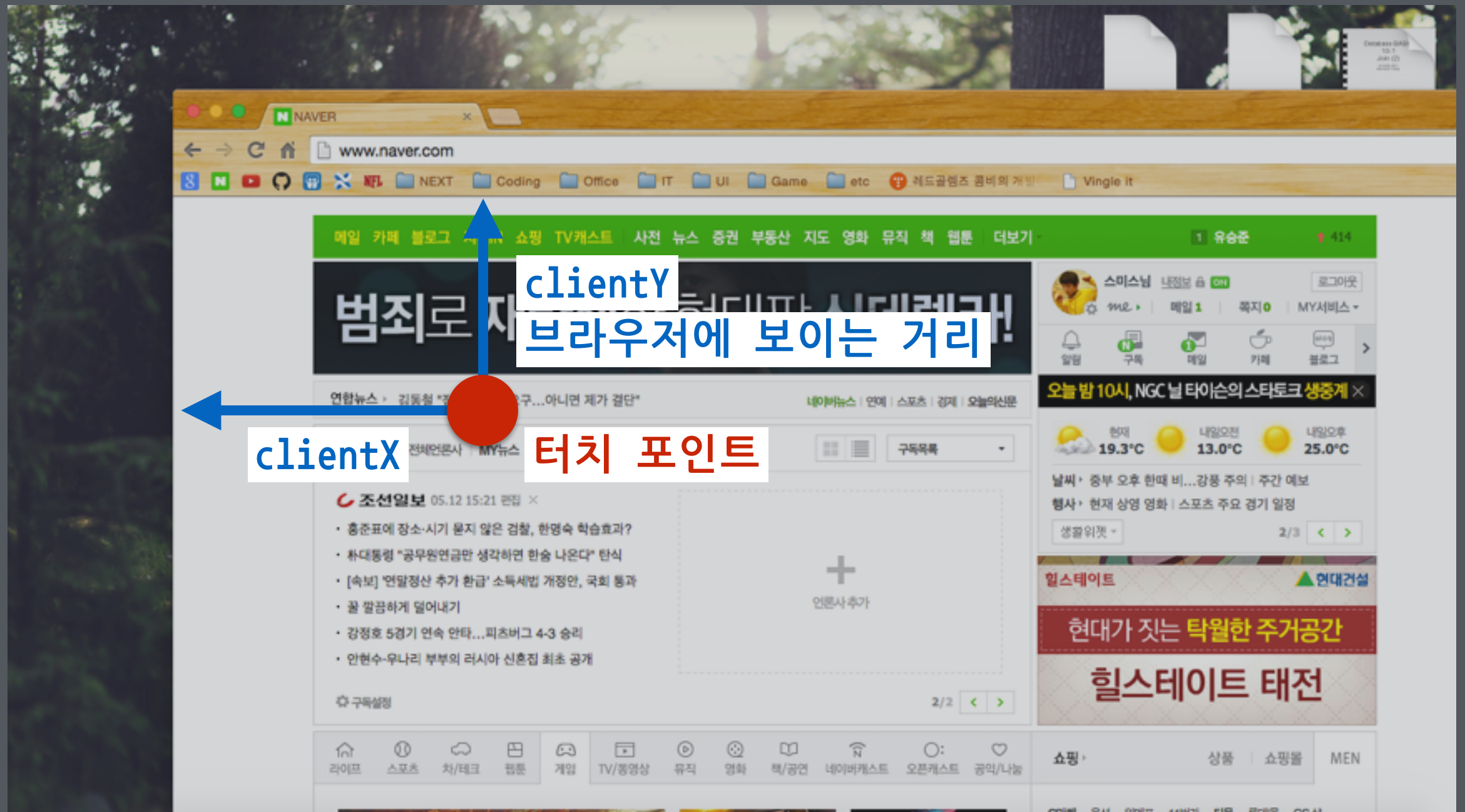
터치 이벤트에 대한 각각의 터치 포인트를 설명합니다.

identifier	손가락을 고유하게 식별하는 번호 터치 포인트가 활성화 될 때 활성화된 다른 터치 포인트와 구별되는 식별자
clientX	스크롤을 제외한 뷰포트의 상대적인 픽셀의 수평 좌표
clientY	스크롤을 제외한 뷰포트의 상대적인 픽셀의 수직 좌표
pageX	스크롤을 포함한 뷰포트의 상대적인 픽셀의 수평 좌표
pageY	스크롤을 포함한 뷰포트의 상대적인 픽셀의 수직 좌표
screenX	픽셀 화면상의 상대적 수평 좌표
screenY	픽셀 화면상의 상대적 수직 좌표
target	화면에 손가락이 처음 닿은 접촉점이 EventTarget입니다. 손가락이 타겟 엘리먼트를 벗어나더라도 EventTarget은 변하지 않습니다.

**터치 포인트 삼형제**

**client, page, screen**

**client:** 스크롤을 제외한 뷰포트의 상대적 좌표



**page:** 스크롤을 포함한 뷰포트의 상대적 좌표

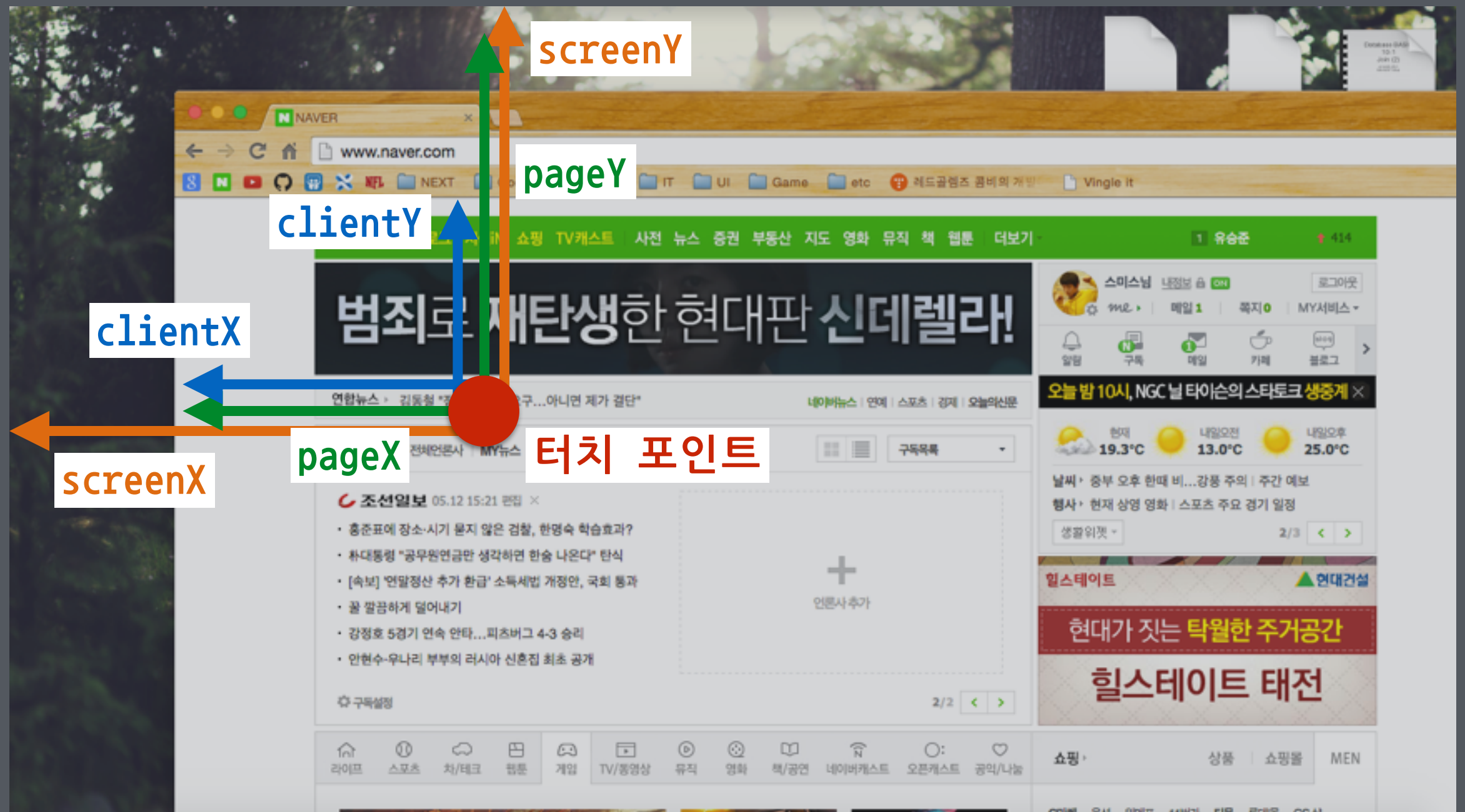




**screen:** 디바이스를 기준으로 한 상대적 좌표



# client, page, screen



# 터치 이벤트 인터페이스

이 인터페이스는 `touchstart`, `touchend`, `touchmove`, `touchcancel` 이벤트를 타입을 정의합니다. `TouchEvent` 객체가 만들어지면 속성은 변하지 않습니다.

## `touches`

현재 화면을 터치하고 있는 모든 접촉점의 터치 리스트입니다. `event.touches.length`를 이용해 멀티 터치 여부를 알 수 있습니다. 특정 터치의 좌표는 `event.touches[i].pageX` 형식으로 가져올 수 있습니다.

## `targetTouches`

현재 이벤트의 타겟인 엘리먼트에서 시작되어 화면을 터치하고 있는 모든 접촉점에 대한 터치 리스트입니다

## `changedTouches`

이벤트에 할당된 모든 접촉점에 대한 터치 리스트 입니다.

이벤트별 `changedTouches`의 의미

- `touchstart`: 현재 이벤트와 함께 바로 활성화된 터치점의 리스트
- `touchmove`: 마지막 이벤트에서 이동한 터치점의 리스트
- `touchend`, `touchcancel`: 화면에서 막 떼어진 터치점의 리스트

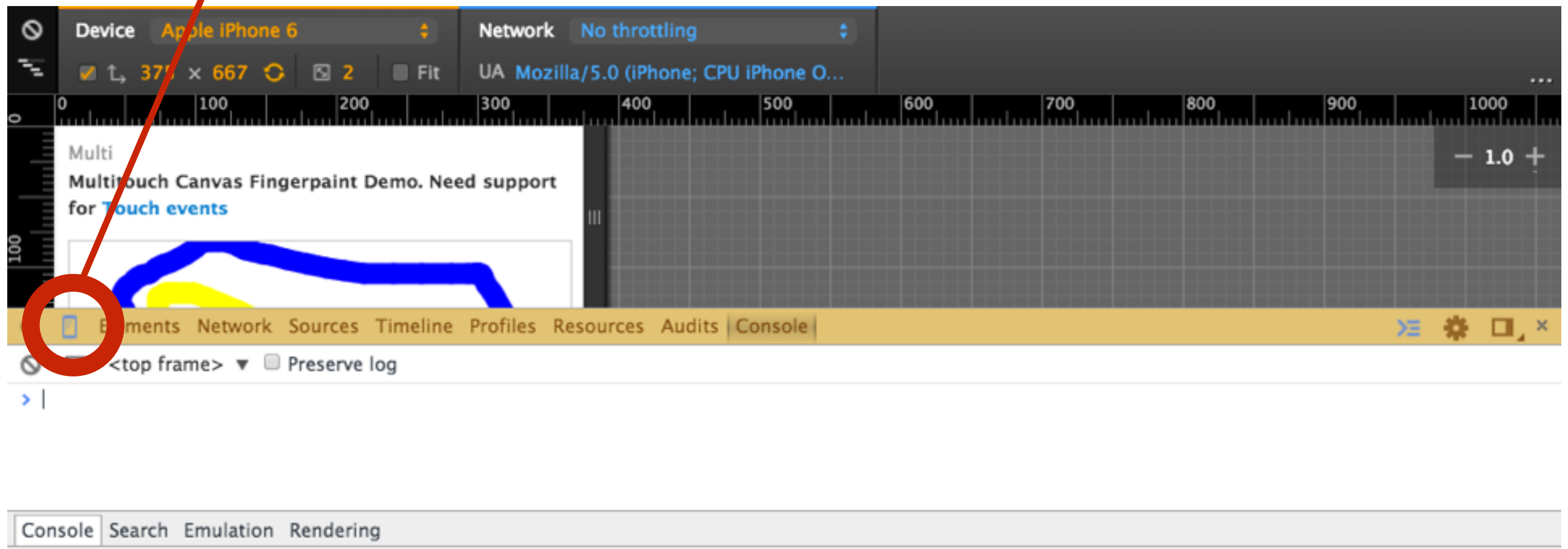
# 아래 코드를 이용해서 공부한 내용을 확인해봅시다

[https://github.com/WooJaeWoo/WEB\\_UI\\_BASIC/tree/master/%5BWeek8%5D%20Mobile\\_Touch\\_Event](https://github.com/WooJaeWoo/WEB_UI_BASIC/tree/master/%5BWeek8%5D%20Mobile_Touch_Event)

에 있는 touch.html, touch.css, touch.js를 참고하세요!

크롬 개발자 도구로  
모바일 화면 확인하기  
대부분 다 알죠???

요거 누르고 refresh하면 됩니당!



이제 이 친구들을 이용해서

모든 제스처를

(예를 들면 tap, swipe 같은...)

구현하시면 됩니다!!!

# 지금 너의 상태





**애써 피곤하게 살 필요가 있나...**

**우리에겐 라이브러리가 있다!!!**

# 제스처 라이브러리

Hammer.js, QuoJS, TouchSwipe, jQueryTouchGestures,  
jQueryMobile, jqTouch, tap.js 등 무지하게 많아요!!

더 많은 정보는 여기를 참고하세요

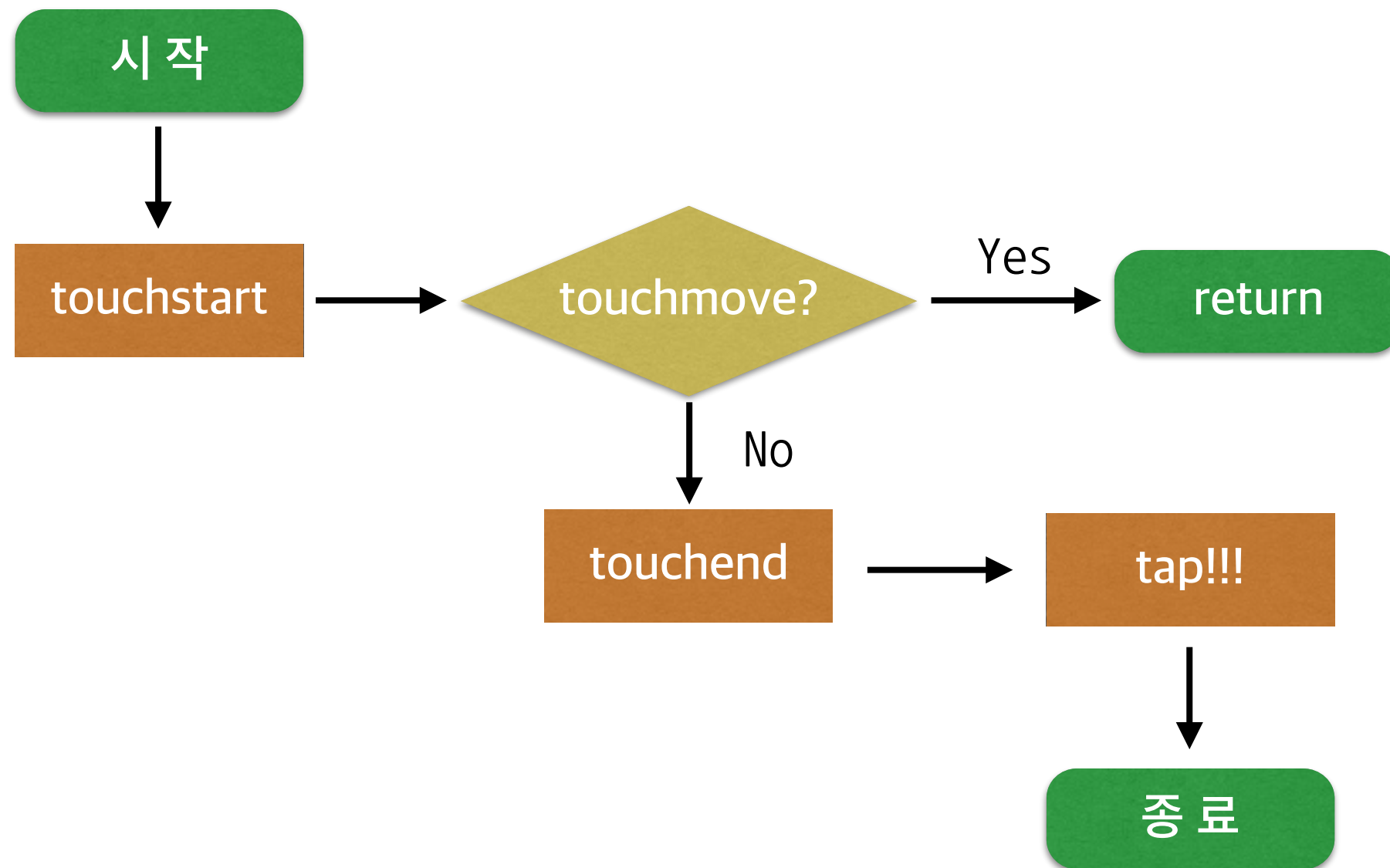
<https://github.com/bebraw/jswiki/wiki/Touch>

다음 jQuery 스터디 때 jQueryMobile 정도  
간단히 보는 것도 나쁘진 않겠군요 ㅎㅎㅎ

## <주의>

라이브러리만 계속 쓰다보면  
게을러져서 소가 될 수도 있습니다  
학습을 위해서 날코딩 해봅시다

# Tap 만들기



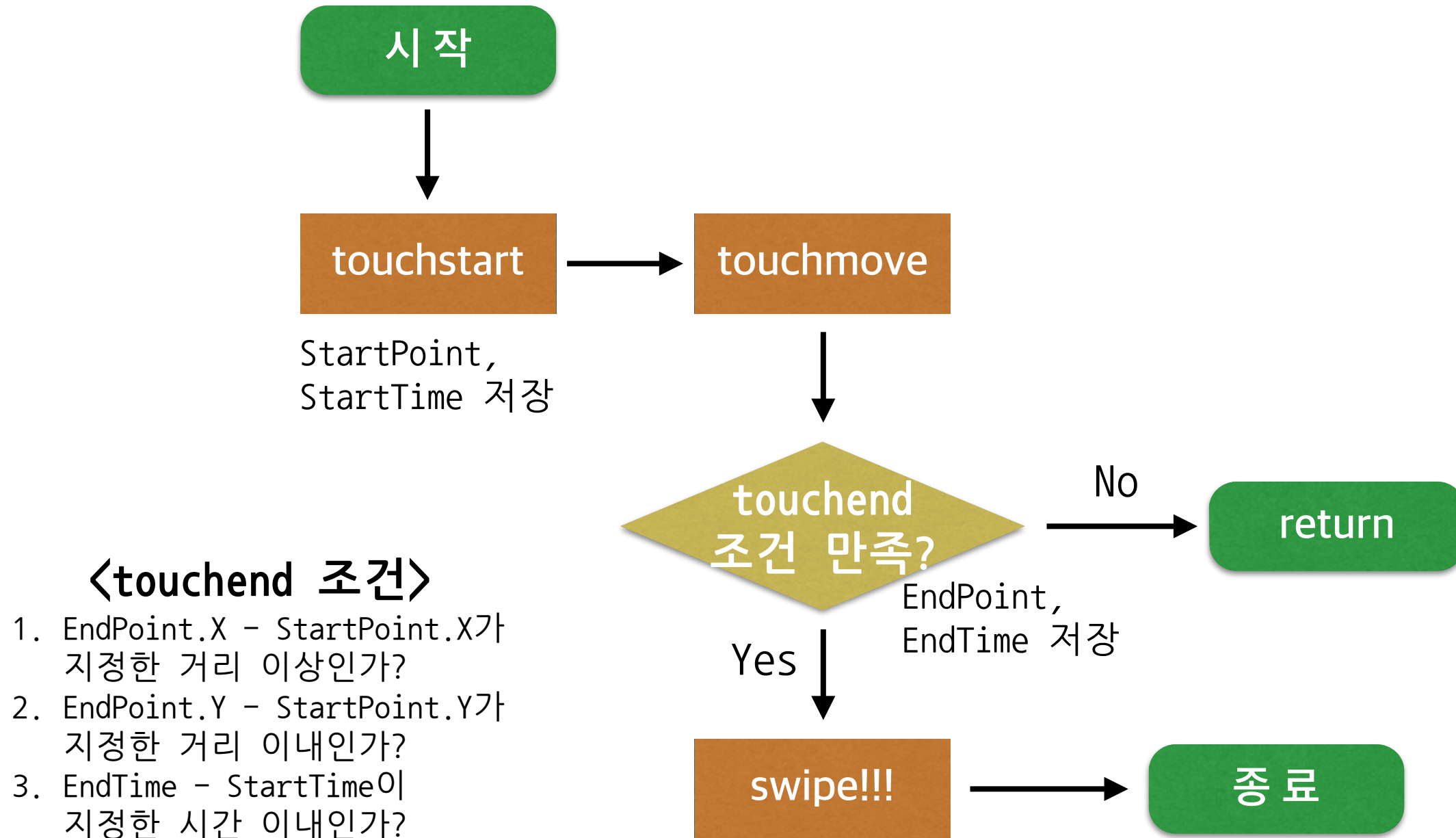
# 예시코드를 확인해봅시다

[https://github.com/WooJaeWoo/WEB\\_UI\\_BASIC/tree/master/%5BWeek8%5D%20Mobile\\_Touch\\_Event](https://github.com/WooJaeWoo/WEB_UI_BASIC/tree/master/%5BWeek8%5D%20Mobile_Touch_Event)

에 있는 tap.js를 참고하세요!

코드 출처: <http://helloworld.naver.com/helloworld/textyle/80243>

# Swipe right 만들기



# 예시코드를 확인해봅시다

[https://github.com/WooJaeWoo/WEB\\_UI\\_BASIC/tree/master/%5BWeek8%5D%20Mobile\\_Touch\\_Event](https://github.com/WooJaeWoo/WEB_UI_BASIC/tree/master/%5BWeek8%5D%20Mobile_Touch_Event)

에 있는 `swipeRight.html`, `swipeRight.js`를 참고하세요!

코드 출처: <http://www.javascriptkit.com/javatutors/touchevents2.shtml>

이런 식으로 원하는 제스처를  
직접 만들어 쓰면 되겠죠?  
당신의 능력을 믿습니다!!!



## 〈참고〉 `e.preventDefault();` ???

iOS overscroll effect의 경우 touchmove 이벤트에 대해서 해당 콘텐츠 경계를 넘어가면 튕기도록 기본 행동이 지정되어 있습니다.

이 경우 외에도 default 이벤트 때문에 터치 이벤트가 제대로 작동하지 않을 수도 있습니다. 그래서 default 이벤트를 막기 위해서 `preventDefault();`를 사용합니다.

```
document.body.addEventListener('touchmove', function(event) {  
    event.preventDefault();  
}, false);
```

**iPhone + Mac 유저에게 팁!**

**모바일 화면을**

**노트북에서 디버그하기**

# Safari Web Inspector

- 1단계: 준비물을 챙깁시다!

Mac OS가 설치된 노트북

Safari Browser

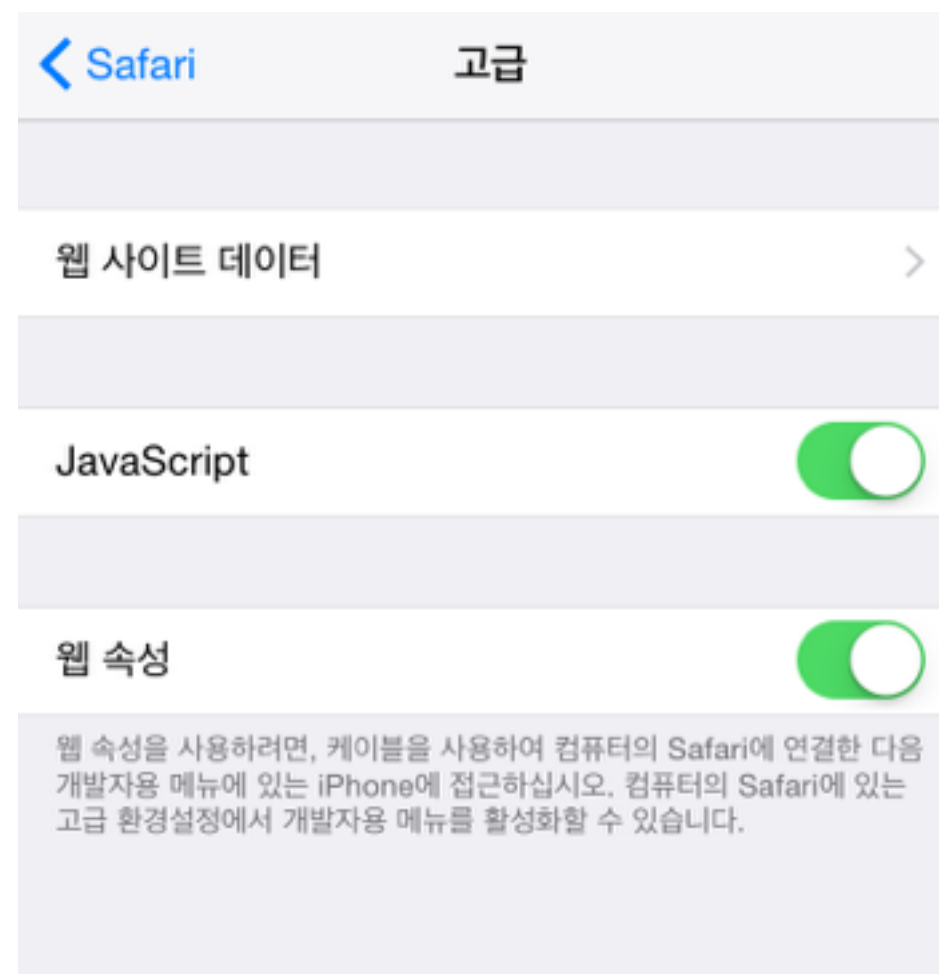
iOS Device

**iPhone USB Cable**

- 2단계: iOS Device와 노트북을 연결합니다.

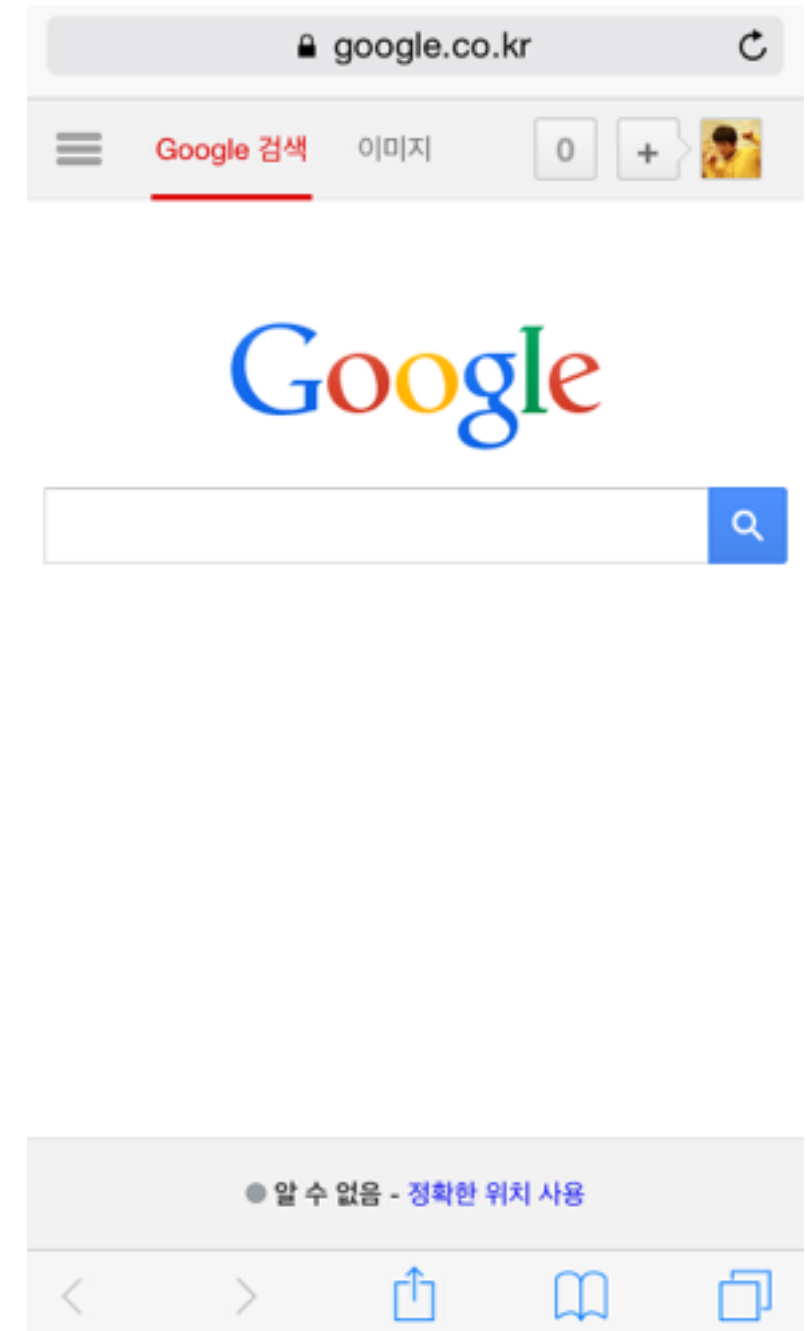
# Safari Web Inspector

- 3단계: iOS 설정-Safari-고급-웹 속성을 활성화 합니다.



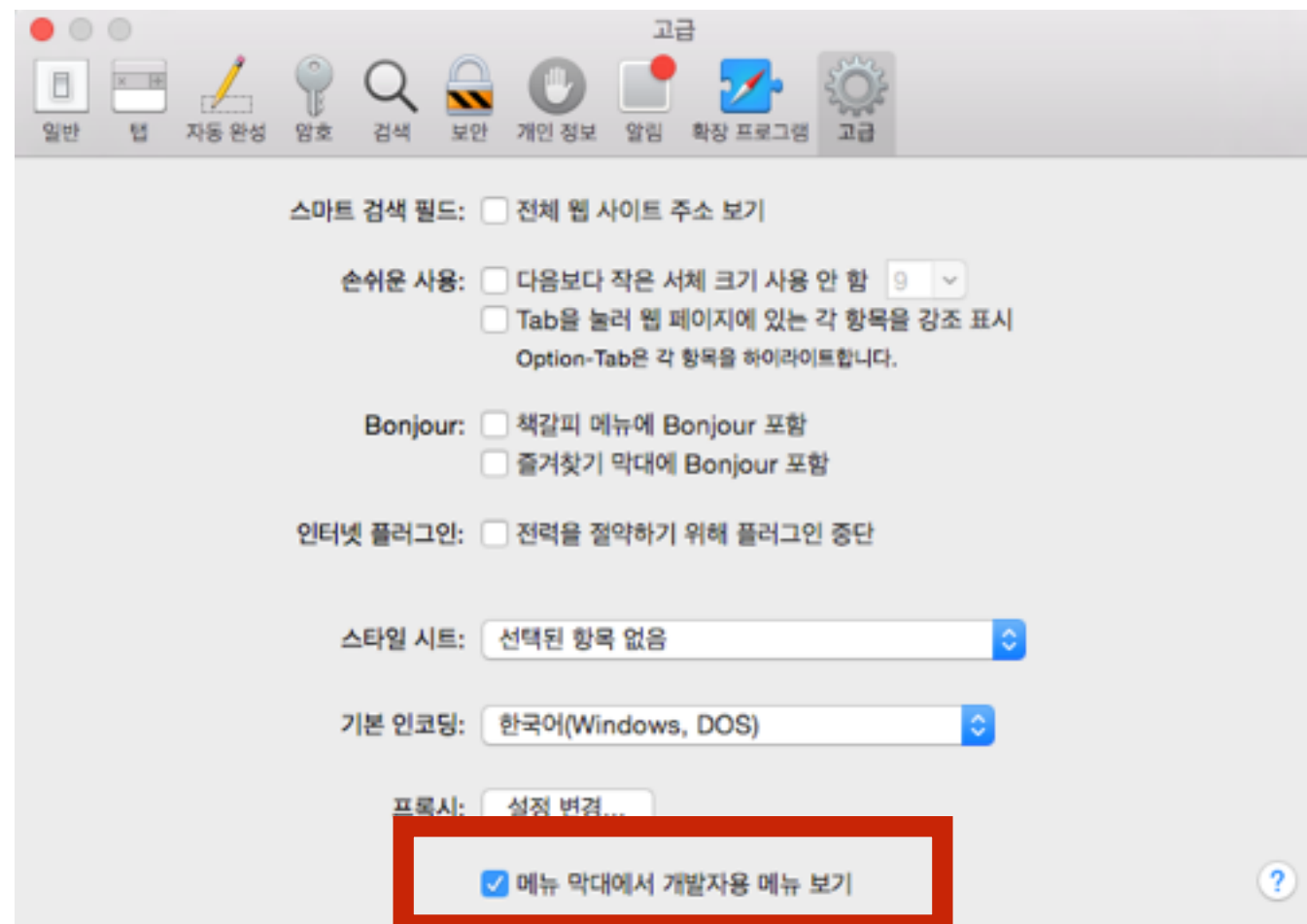
# Safari Web Inspector

- 4단계: iOS Device에서 Safari를 켜고, 디버그할 페이지에 접속합니다.



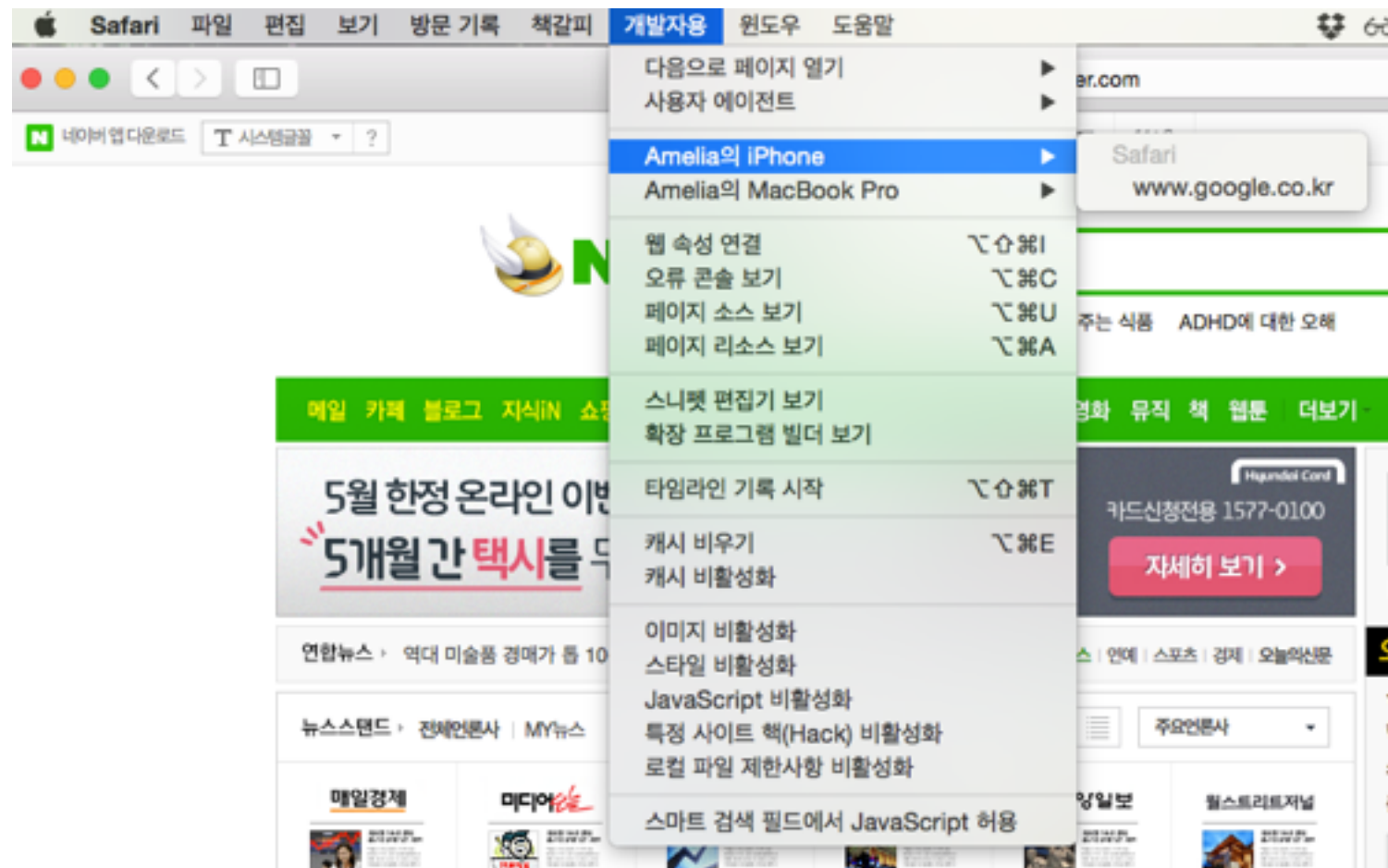
# Safari Web Inspector

- 5단계: 노트북 Safari 환경설정-고급 탭에서 개발자용 메뉴를 활성화 합니다.



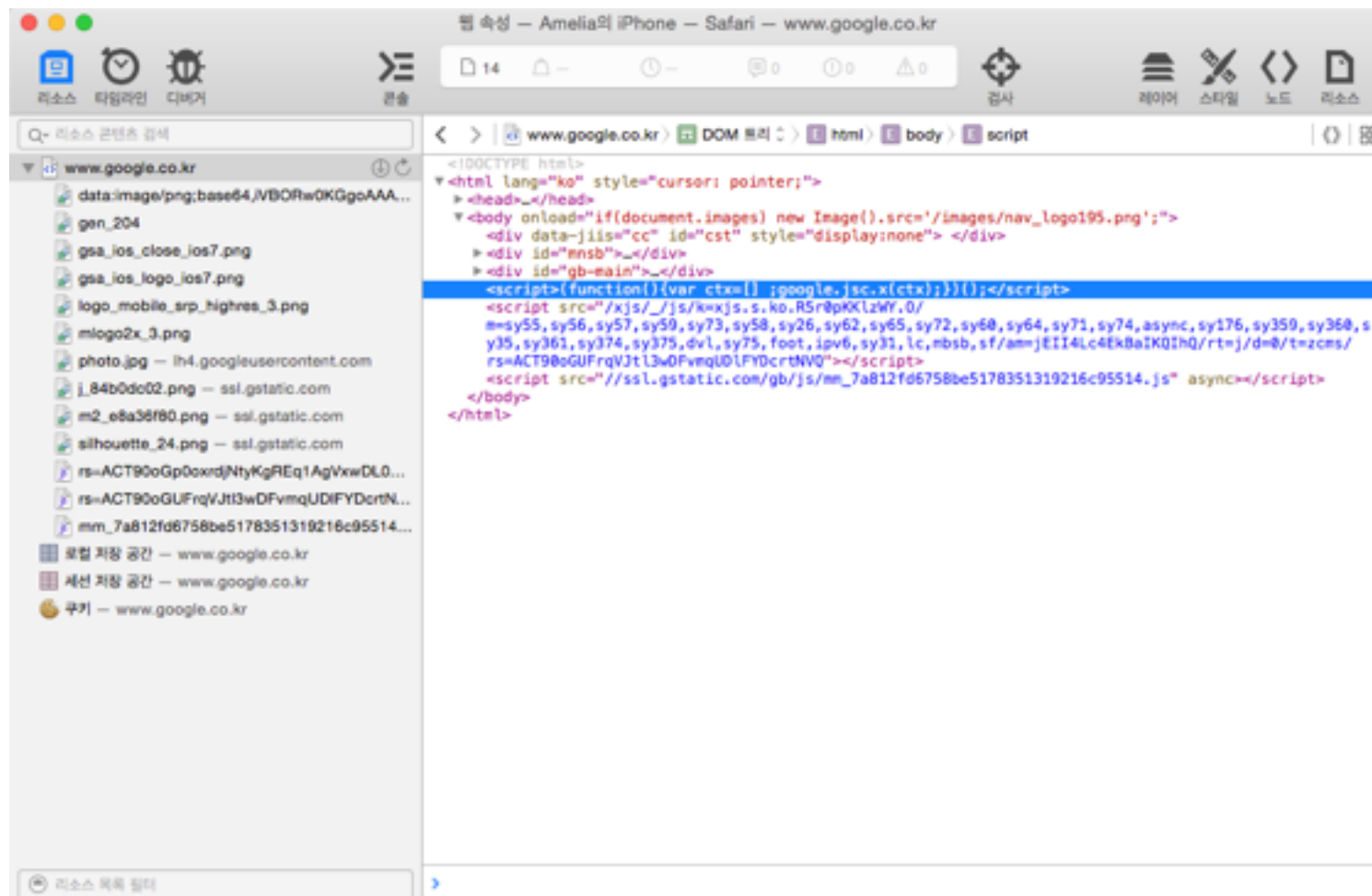
# Safari Web Inspector

- 6단계: 개발자용 메뉴에서 iPhone을 보면  
지금 사파리에서 켜진 페이지들을 확인할 수 있습니다!



# Safari Web Inspector

- 7단계: Inspector를 켜서 디버그 짜라란!





## 참고 사이트

<http://helloworld.naver.com/helloworld/textyle/80243>

<http://wit.nts-corp.com/2013/12/20/583>

<http://www.html5rocks.com/en/mobile/touch/>

<http://www.javascriptkit.com/javatutors/touchevents2.shtml>

끝!