

## 1강 PC까지 고려한 모바일 웹

이 강좌에서는 데스크톱 PC는 물론 스마트 기기인 스마트폰과 태블릿 패드까지 고려한 웹 페이지를 만들어 보겠습니다.

웹을 이용할 수 있는 기기가 많아지면서 각 기기에 맞는 디자인을 만드는 것이 점점 어려워지고 있습니다. 데스크톱 PC만 하더라도 플랫폼이나 웹 브라우저별로 구분하기 힘든데, 거의 모든 기기에 대응한다는 것은 불가능에 가까운 일입니다. 하지만 이런 문제를 말끔히 해결할 수 있는 기술이 차차 등장하고 있습니다. 여기서는 한 가지 문서로 세가지 기기에 모두 대응하는 디자인을 해보겠습니다.

### 1 HTML5 웹 문서 구조 작성

아래 내용은 IE 브라우저에 따라 다른 클래스가 적용됩니다.

```
<!--[if IE 6]><html class="no-js old ie6"><![endif]-->
<!--[if IE 7]><html class="no-js old ie7"><![endif]-->
<!--[if IE 8]><html class="no-js old ie8"><![endif]-->
<!--[if IE 9]><html class="no-js modern ie9"><![endif]-->
<!--[if (gt IE 9)!!(IE)]><!--><html class="no-js modern"><!--<![endif]-->
```

IE 브라우저의 경우, 크롬 브라우저가 설치되어 있다면 크롬 방식으로 렌더링됩니다.

```
<!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1" /><![endif]-->
```

viewport를 visual viewport로 지정합니다.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

문서에 관련된 스타일을 적용합니다.

```
<link rel="stylesheet" href="css/style.css" />
```

IE 6 ~ 8에서는 미디어 쿼리를 지원하지 않지만 'respond.js'을 이용하면 부분적으로나마 지원하게 할 수 있습니다.

```
<script src="include/js/libs/respond.min.js"></script>
```

Modernizr는 사용자의 브라우저에서 HTML5 및 CSS3 기능을 감지하는 자바스크립트 라이브러리입니다.(<http://modernizr.com/>)

```
<script src="js/libs/modernizr.min.js"></script>
```

jQuery 라이브러리 외부 CDN입니다.

```
<script src="http://code.jquery.com/jquery.min.js"></script>
```

jQuery 스크롤 고정하는 플러그인입니다.

```
<script src="js/jquery.scrolledFix.js"></script>
```

문서에 관련된 스크립트를 적용합니다.

```
<script src="js/script.js"></script>
```

HTML5를 개발하는데 있어서 가장 큰 걸림돌이라하면 바로 인터넷 익스플로러(IE)일 것입니다. 이러한 문제를 해결하기 위한 한 가지 방법은 바로 구글의 크롬 프레임(Chrome Frame)입니다.

#### \* 구글 크롬 프레임(Google Chrome Frame(GCF))

: 구글 크롬 프레임은 IE에서 구글 크롬의 환경을 구현할 수 있도록 도와주는 오픈소스 플러그인입니다. 구글의 V8 자바스크립트 엔진과 다양한 웹기술들을 최신 경향의 적용이 느린 IE에 적용시켜줄 수 있는 유용한 도구입니다. 이것을 통해 무엇보다도 canvas나 WebRTC와 같은 다양한 HTML5 태그들을 이용할 수 있고 자바스크립트의 성능이 월등히 좋은 구글의 자바스크립트 엔진 V8을 이용할 수 있게 됩니다.

: HTML5와 자바스크립트 개발자라면 이것이 얼마나 유용할 것인지 알 수 있을 것입니다.

#### \* 구글 크롬 프레임 웹 페이지에 적용하기

: 일단 적용하고자하는 html 웹 페이지를 열어서 페이지의 위 <head> 근처 다른 <meta> 태그들과 함께 다음을 넣어줍니다.

```
<!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1" /><![endif]-->
```

: 정말로 끝입니다. 이러면 적용은 끝난 것입니다. 이것을 보자마자 도대체 구글은 어떻게 되먹은 놈들인가 생각이 듭니다. 하지만 문제는 사용자가 구글 크롬 프레임이 깔려있다면 끝난 것이지만, 안깔려있다면 아무 현상도 일어나지 않습니다. 위의 경우 chrome=IE9는 IE9 버전 이하에서 작동을 하겠다는 것이고, chrome=1이면 어느 버전에서든지 무조건 작동을 하게 됩니다.

: 만약 사용자가 구글 크롬 프레임이 설치되지 않은 경우 설치여부를 확인하고 설치를 권유해야 할 것입니다. 그것 또한 방법을 제공해주고 있는데, 아래와 같이 작성합니다.

```
<!--[if lt IE 9]><script src="include/js/libs/CFInstall.min.js"></script>
<script>window.attachEvent("onload",function(){CFInstall.check({mode:"overlay"}});</script><![endif]-->
```

이렇게 작성하면 만약 사용자가 구글 크롬 프레임이 깔려있지 않다면 구글 크롬 프레임을 설치하라고 하는 다이얼로그가 나올 것이다. 이러한 설치 방식은 다양한데, overlay는 페이지를 먹통으로 만들고 반드시 설치를 요구할 때에 이용하는 것이고, inline으로 설정한다면 페이지 내에 iframe을 추가하게 되고 그 안에서 설치를 하는 url을 띄워줄 것입니다.

다음에 설정된 코드는 PNG 그래픽과 관련된 것입니다. 이번 예제에서는 png 그래픽을 사용할 것입니다. 하지만 IE 6 구형 웹 브라우저에서는 PNG 형식을 정상적으로 처리하지 못합니다. 즉, 조건부 주석문을 이용하여 IE 6에서만 처리되도록 합니다.

```
<!--[if IE 6]><script src="include/js/libs/DD_belatedPNG.min.js"></script>
<script>DD_belatedPNG.fix('.png_bg, img');</script><![endif]-->
```

## responsive\_web.html

```
<!DOCTYPE html>
<!--[if IE 6]><html class="no-js old ie6"><![endif]-->
<!--[if IE 7]><html class="no-js old ie7"><![endif]-->
<!--[if IE 8]><html class="no-js old ie8"><![endif]-->
<!--[if IE 9]><html class="no-js modern ie9"><![endif]-->
<!--[if (gt IE 9)!!(IE)]><!--><html class="no-js modern"><!--<![endif]-->
<head>
<!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1" /><![endif]-->
<meta charset="utf-8" />
<title>반응형 웹 디자인</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<script src="js/libs/respond.min.js"></script>
<link rel="stylesheet" href="css/style.css" />
<script src="js/libs/modernizr.min.js"></script>
<script src="js/libs/jquery.min.js"></script>
<script src="js/jquery.scrolledFix.js"></script>
<script src="js/script.js"></script>
<script src="js/libs/respond.min.js"></script>
<!--[if lt IE 9]><script src="js/libs/CFInstall.min.js"></script>
<script>window.attachEvent("onload",
function(){CFInstall.check({mode:"overlay"}});</script><![endif]-->
<!--[if IE 6]><script src="js/libs/DD_belatedPNG.min.js"></script>
<script>DD_belatedPNG.fix('.png_bg, img');</script><![endif]-->
</head>
```