

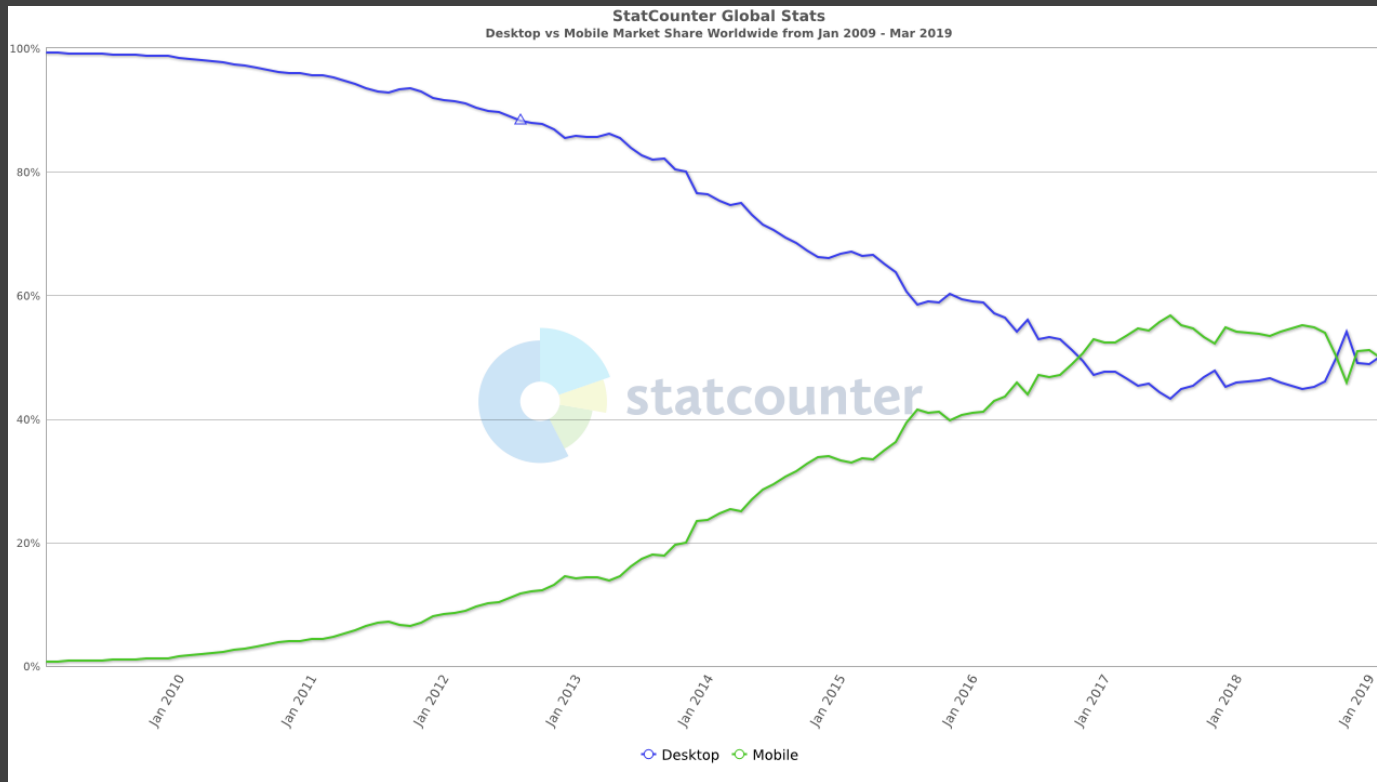
기초웹개발론

<http://bit.ly/2Wzkt0L>

kakao

Mobile & Responsive Web

Desktop VS Mobile



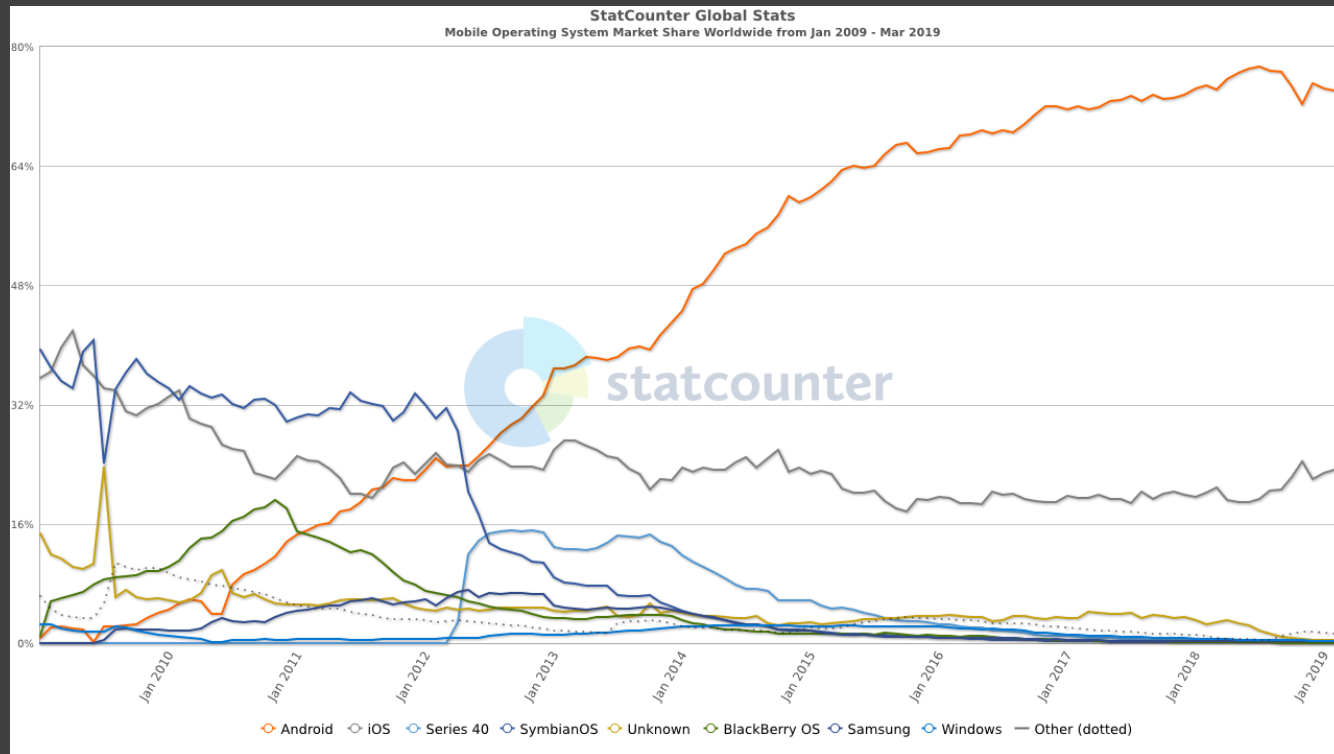
kakao

Mobile OS

- 안드로이드 - 구글
- ios - 애플
- 블랙베리 - RIM
- 바다, 타이젠 - 삼성
- 심비안 - 노키아
- 윈도우모바일 - MS

kakao

Mobile OS Market Share



kakao

Android

- 2005년 구글이 안드로이드사 인수
- 2008년 오픈소스로 공개
- 2010년 레퍼런스폰 HTC NEXUS ONE 발표
- 2014년 안드로이드 에브리웨어(안드로이드 웨어, 안드로이드 오토, 안드로이드 TV) 발표
- 제조사에서 커스텀이 가능하여 다양한 제조사에서 다양한 기기를 제작함.
- 기기 파편화가 심하며, 보안에 취약함.

Android

- 알파벳 첫글자를 오름차순으로 정렬한 디저트 이름을 코드명으로 사용.
 - C. 컵케이크(1.5)
 - D. 도넛 (1.6)
 - E. 에클레어(2.0)
 - F. 프로요(2.2)
 - G. 진저브레드(2.3)
 - H. 허니콤(3.0)
 - I. 아이스크림 샌드위치(4.0)
 - J. 젤리빈(4.3)
 - K. 킷캣(4.4)
 - L. 롤리팝(5.x)
 - M. 마시멜로(6.x)
 - N. 누가(7.x)
 - O. 오레오(8.x)
 - P. 파이(9.x)

IOS

- 2007년 Iphone OS 로 공개(아이폰, 아이패드터치)
- 2008년 6월 이후 SDK 공개, 2.0 부터 앱스토어 운영.
- 2010년 IOS 로 이름 변경
- 멀티 터치 최적화
- Apple 기기 전용 OS.
- 폐쇄적인 정책으로 타 OS 보다 보안성이 뛰어남.
- 2019년 현재 버전 12까지 발표됨

Mobile Browser

- Safari
 - webkit 기반
 - IOS 기본 브라우저
 - OS X 의 사파리를 통해 디버깅 이 가능함.
- Android 브라우저
 - webkit 기반
 - Android 4.4 이전의 기본 브라우저
 - 기기 제조사별로 커스텀을 통해 다른 사용성을 가짐

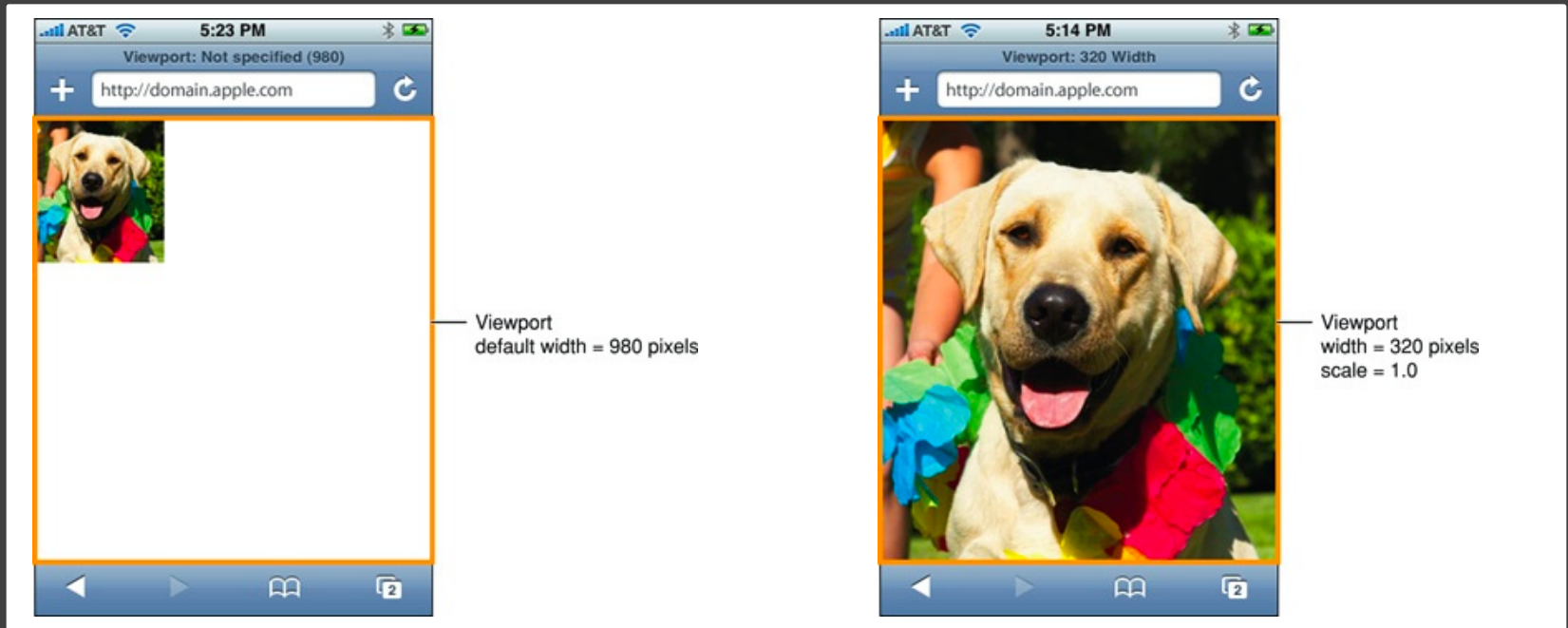
Mobile Browser

- Chrome For Android
 - webkit 기반
 - Android 4.4 이후 기본브라우저로 채택
- Samsung Internet
 - chrome 과 같은 blink 엔진을 사용했으나, 랜더링의 차이가 있다.
 - 삼성 로그인과 같은 독자적인 기능.
 - 선택제에서 플레이스토어나 APK 를 이용한 설치가 가능해짐.

viewport

- 모바일 브라우저에서 상단 주소바 및 기타 영역을 제외한 contents 가 그려지는 부분
- 세로모드는 portrait, 가로 모드는 landscape 라고 함
- 애플에서 자사 브라우저를 위해 만든 비표준.
- PC 화면에서 보이는 내용을 모바일 기기에서 그대로 볼수 없는 이유는 PC 화면과 모바일 화면의 픽셀 표현법이 다르기 때문.
- viewport 를 지정하면 접속한 기기 화면에 맞추어 확대하거나 축소해 표시 할수 있음.
- webkit 기반의 브라우저는 기본 width가 980px 임.
- 모바일 기기의 종류가 많아 다양한 환경이 존재함으로 필수적인 설정임

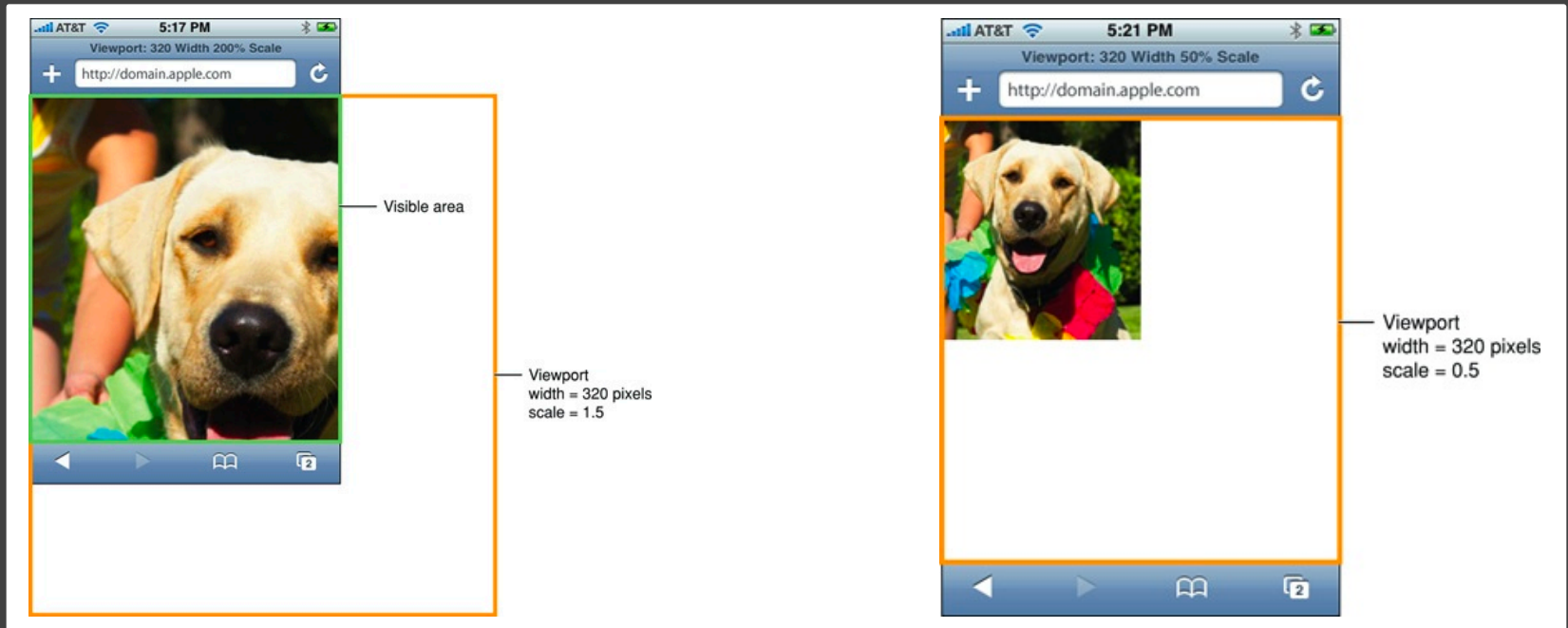
viewport 이해하기



320x356픽셀의 이미지가 있는 웹페이지이다.
아이폰OS의 default viewport(980px)로 렌더링.

viewport 를 320px 로 설정

viewport 이해하기



viewport는 보여지는 영역보다 크거나 작을수 있으며,
보여지는 영역보다 클경우 패닝이 가능하다.
위의 이미지는 scale = 1.5

위의 이미지는 scale = 0.5,

viewport 지정하기

- head 태그 안에 meta 태그 를 이용하여 적용함.

속성값	설명	기본값
width	픽셀단위로 설정하는 뷰포트의 가로 넓이	980px
height	픽셀단위로 설정하는 뷰포트의 세로 높이	width 와 기기의 비율로 계산
initiali-scale	배율로 설정가능한 뷰포트 초기값	1
minimum-scale	최소 배율	0.25
maximun-scale	최대 배율	1.6
user-scalable	사용자의 축소/확대 여부 설정	yes

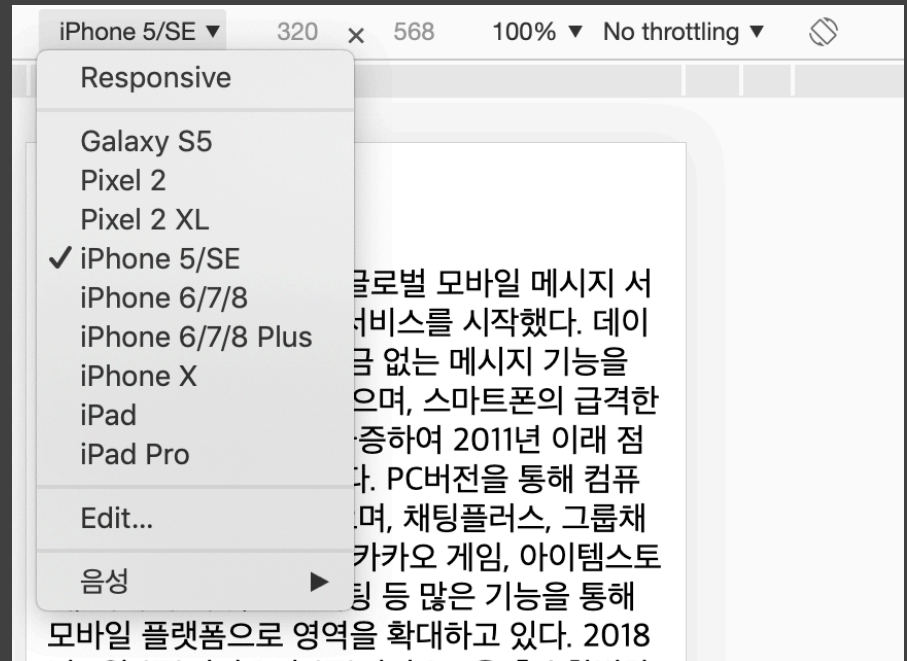
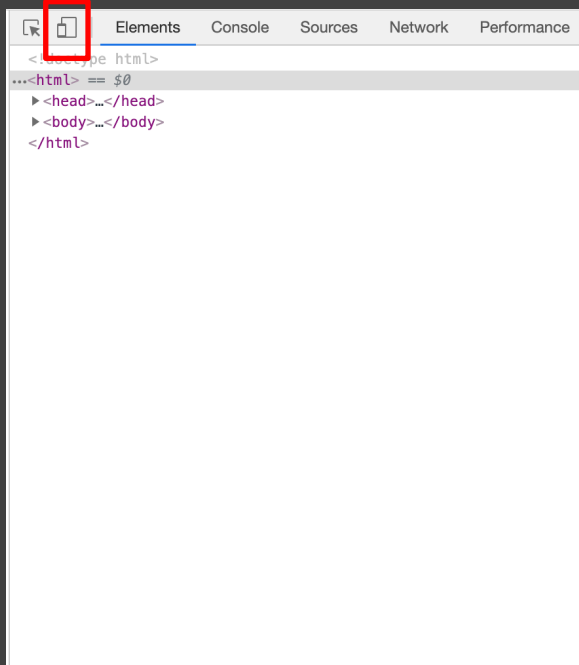
viewport 태그

- 가로세로 픽셀을 고정하지 않고, 디바이스에 따라 유연하게 대처하기 위해 device-width 를 사용가능하다.
 - `<meta name="viewport" content="width=device-width">`
- scale 도 지정가능
 - `<meta name="viewport" content="initial-scale=1.0">`
- 사용자의 scale 변경을 막을수 있음
 - `<meta name="viewport" content="initial-scale=1.0, user-scalable=no">`
- `<meta name="viewport" content="user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, width=device-width">`

viewport 실습

<http://bit.ly/2Wzkt0L>

Mobile 디버깅



kakao

해상도

- 고해상도의 디바이스 들이 출시되며, 폰의 해상도와 브라우저의 해상도가 다른 현상이 나타남.
- iPhone 3GS
 - 기기 : 320x240 / 브라우저 : 320x240
- 아이폰 4
 - 기기 : 640x960 / 브라우저 : 320x480
- 아이폰 5
 - 기기 : 640x1136 / 브라우저 : 320x568
- 아이폰6, 아이폰6 P
 - 기기 : 750x1334 / 브라우저 : 375x667 , 기기 : 1242x2208 / 브라우저 : 414x736

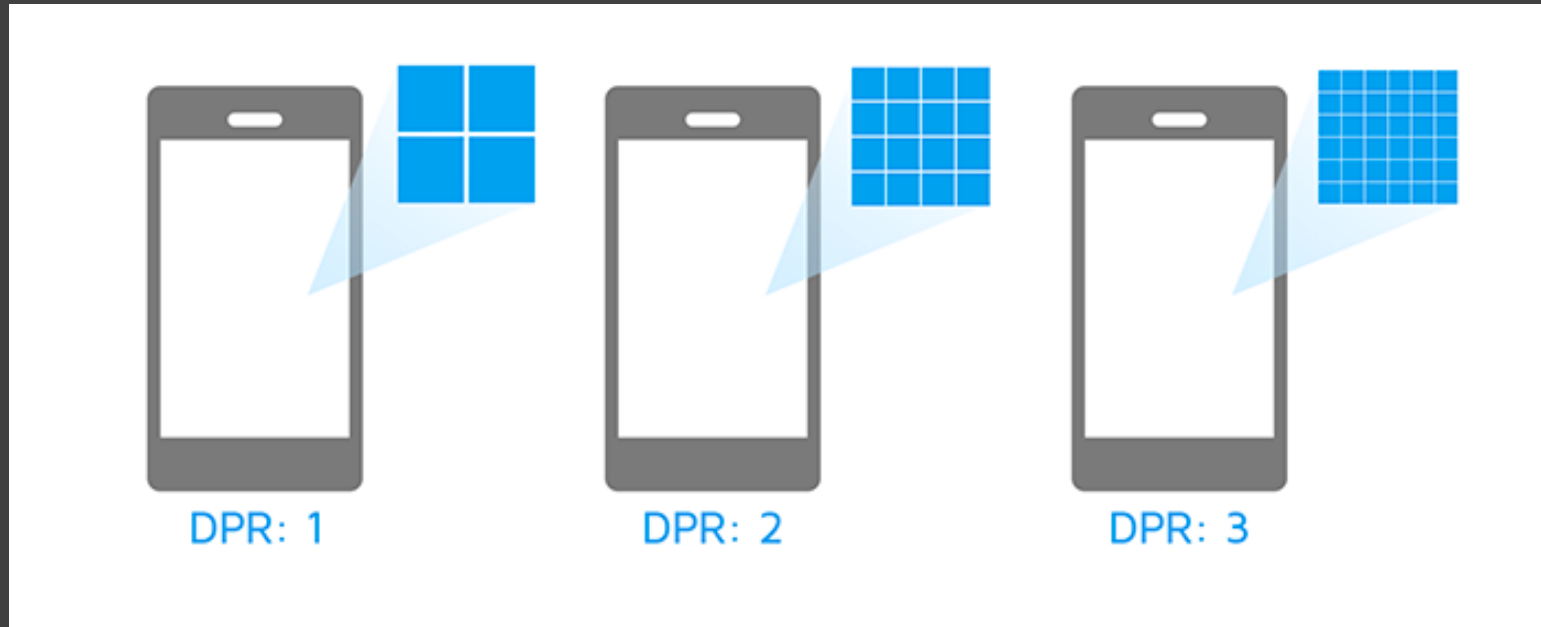
Device Pixel Ratio

- 디바이스의 해상도는 밀도(Density)로 구분됨
- 밀도 : 물리적 공간에 얼마만큼의 픽셀이 포함되어 있는가.
- 단위 : PPI(Pixel Per Inch), 1인치 안에 포함된 픽셀의 갯수
- 72ppi : 1인치 안에 72개의 픽셀, 300ppi : 1인치 안에 300 개의 픽셀
- ppi 가 높을 수록 고해상도의 디바이스.

Device Pixel Ratio

- iPhone 3GS
 - 기기 : 320x240 / 브라우저 : 320x240
 - 1 dpi
- 아이폰 4
 - 기기 : 640x960 / 브라우저 : 320x480
 - 2 dpi
- 아이폰 5
 - 기기 : 640x1136 / 브라우저 : 320x568
 - 2 dpi
- 아이폰6, 아이폰6 P
 - 기기 : 750x1334 / 브라우저 : 375x667 , 기기 : 1242x2208 / 브라우저 : 414x736
 - 2 dpi, 3 dpi

Device Pixel Ratio

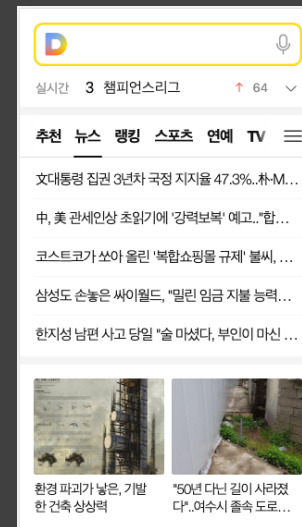
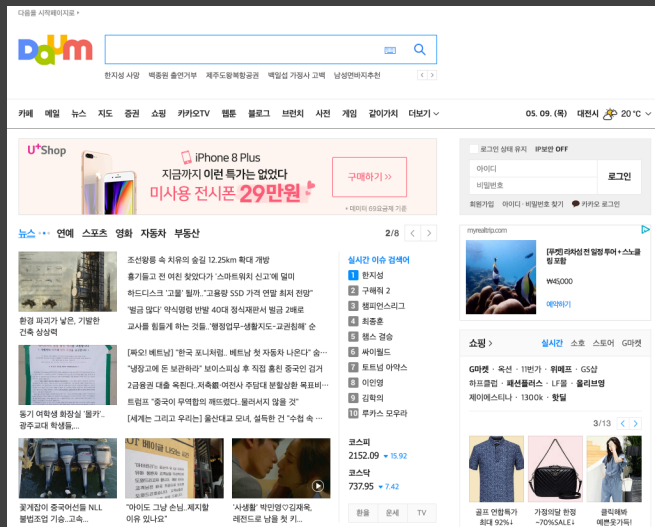


국내 모바일 웹 히스토리

- WAP -> Wireless Application Protocol
 - XML 기반의 Wireless Markup Language 를 사용.
 - <http://bit.ly/2PI0BG3>
- 2008년 LG 에서 폴라리스 브라우저가 탑재된 기기 출시
 - 선탭재 된 브라우저가 버그 덩어리..
 - 모바일웹의 대중화에 초석
- 2009년 10월 윈도우 모바일 기반의 삼성 옴니아폰 출시
- 2009년 11월 국내 아이폰 3GS 발매
- 2010년 갤럭시A 로 국내 최초 Android 폰 발매.

모바일웹

- 모바일과 PC 를 구분해서 서비스 하는 방법
 - 모바일 디바이스에서는 모바일 전용페이지, 데스크탑에서는 데스크탑 전용페이지를 보여주는 방식



모바일웹

- 모바일, 데스크탑 웹 페이지를 분리하는 방법의 장,단점
- 장점
 - 각 환경 별로 성능을 최적화 할수 있음.
 - 모바일에 대한 퀄리티 를 높일수 있음.
- 단점
 - 데스크탑/모바일 을 별도로 작업해야 하기 때문에 인력 과 시간 이 필요함.
 - 새로운 환경에 대응하기가 어려움.



Responsive Web



kakao

Responsive Web

- 가변그리드, 가변 이미지, 미디어 쿼리를 이용해 하나의 소스(One Code)로 제작된 콘텐츠가 모바일 기기에 맞춰 해상도와 화면이 동적으로 변하는 기술을 말한다.
- 특징
 - 하나의 콘텐츠에 하나의 HTML 소스만 있음, HTML 이 여러벌 있으면 반응형이라 부르지 않음.
 - 특정 장치에 최적화된 페이지로 연결되는 별도의 URL 이 있으면 안됨.
- 목적
 - 사용자에게 디바이스에 최적화되 더 나은 사용성을 제공해줄수 있음.
 - 향후 출시될 디바이스에도 유연하게 대응이 가능함.
- 콘텐츠를 효율적으로 보여주기 위해선 반응형 웹이 반드시 정답은 아니다.

적용방법 - @media queries

- CSS2 의 미디어 타입은 단말기의 종류에 따라 각기 다른 스타일시트를 적용하는 기능, 실제 기기의 특성에 맞춰 적용하기는 어려워 많이 사용되지 않음.
- CSS3 에서 미디어 타입을 개선하여 구체적인 조건에서 스타일을 정확하게 적용할수 있도록 확장. 이를 media queries 라고 한다.

미디어 쿼리 사용예

- 외부 CSS 를 조건에 따라 불러오는 방법

```
<link rel="stylesheet" media="screen and (min-width: 640px)" href="example1.css">  
<link rel="stylesheet" media="screen and (min-width: 1024px)" href="example2.css">  
  
<!-- 화면 너비가 640px 이상이면 example1.css 를 로드, 1024px 이상이면 example2.css 도 같이 로드함 -->
```

미디어 쿼리 사용예

- <style> 태그를 이용하는 방법

```
<style type="text/css" media="screen and (min-width: 640px)">
  body { background-color: #000 }
</style>

<style type="text/css" media="screen and (min-width: 1024px)">
  body { background-color: #fff }
</style>

<!-- 화면 너비가 640px 이상이면 배경색이 검정, 1024px 이상이면 배경색이 흰색으로 캐스캐이딩 됨 -->
```

미디어 쿼리 사용예

- CSS 파일내에서 사용하는 방법

```
/* 1. 모바일 (뷰포트너비 640px 미만) */  
.sample {background-color:black}  
  
/* 2. 패블릿 가로모드/태블릿 (뷰포트 너비 640px 이상) */  
@media all and (min-width:640px) {  
    .sample {background-color:red}  
}  
/* 3. PC (해상도 1024px 이상) */  
@media all and (min-width:1024px) {  
    .sample {background-color:green}  
}
```


미디어 종류

- all : 모든 미디어 타입
- aural : 음성 합성장치
- braille : 점자 표시 장치
- handheld : 손으로 들고다니면서 볼 수 있는 작은 스크린에 대응
- print : 인쇄용도
- projection : 프로젝터 표현용도
- screen : 컴퓨터 스크린을 위한 용도
- tty : 터미널과 같은 고정폰트 장치
- tv : 음성과 영상이 함께 출력되는 TV 같은 곳에 사용
- embossed : 페이지에 인쇄된 점자 표지 장치

미디어 쿼리 문법

- @media 키워드

```
@media only all and (조건문) {실행문}
```

- 기본문법

```
[ONLY | NOT]? s* media_type s* [AND s* expression]* | expression [AND s* expression]
```

- only : 미디어 쿼리 지원 되는 브라우저에서만 실행
- not : 해당 미디어 타입만 제외한 다른 미디어 타입에 대응.

미디어 쿼리 문법

- width/height

- 뷰포트의 너비와 높이. 뷰포트의 크기는 HTML body 콘텐츠를 표시하는 영역으로 실제 스크린의 크기와는 다르다. 반응형 웹 구현시 일반적으로 많이 사용하는 표현

```
/* min-width : 최소 너비를 지정하고 지정된 값보다 너비가 길면 그 값에 대해 반응 */
@media only all and (min-width:981px) {...}

/* max-width : 최대 너비를 지정하고 지정된 값보다 너비가 좁은경우에만 그 값에 대해 반응 */
@media only all and (max-width:980px) {...}

/* 768px 이상 1024px 이하이면 실행 */
@media only all and (min-width:768px) and (max-width:1024px) {...}

/* 너비가 768px 또는 1024px 이면 실행 */
@media only all and (width:768px), (width:1024px) {...}

/* 너비가 768px 이상 그리고 1024px 이하가 아니면 실행 */
@media not all and (min-width:768px) and (max-width:1024px) {...}
```

미디어 쿼리 문법

- device-width/device-height

- 스크린의 너비와 높이, 스크린은 출력장치가 픽셀을 표현할수 있는 모든 영역으로 일반적으로 HTML body 콘텐츠를 표시하는 뷰포트보다 크다.

```
/* 스크린 너비가 320px, 그리고 높이가 480px 이면 실행 */
@media only all and (device-width:320px) and (device-height:480px) {...}

/* 스크린 너비가 최소 320px, 그리고 높이가 최소 480px 이면 실행 */
@media only all and (min-device-width:320px) and (min-device-height:480px) {...}
```

- orientation

- 뷰포트의 너비와, 높이를 이용해서 가로 모드/세로 모드 를 판단한다.

```
/* 세로모드, 뷰포트의 높이가 너비에 비해 상대적으로 크면 실행 */
@media only all and (orientation:portrait) {...}

/* 가로모드, 뷰포트의 너비가 높이에 비해 상대적으로 크면 실행 */
@media only all and (orientation:landscape) {...}
```

미디어 쿼리 문법

- aspect-ratio

- 뷰포트의 너비와 높이에 대한 비율, 너비/높이 순으로 조건을 작성한다. min/max 접두어를 이용해 최소/최대 비율을 설정할수 있다.

```
@media all and (aspect-ratio:5/4) {...} /* 뷰포트의 너비/높이가 5/4 비율이면 실행 */  
@media all and (min-aspect-ratio:5/4) {...} /* 뷰포트의 너비/높이가 5/4 비율 이상이면 실행 */  
@media all and (max-aspect-ratio:5/4) {...} /* 뷰포트의 너비/높이가 5/4 비율 이하이면 실행 */
```

- device-aspect-ratio

- 스크린의 너비와 높이에 대한 비율, 너비/높이 순으로 조건을 작성한다. min/max 접두어를 이용해 최소/최대 비율을 설정할수 있다.

```
@media all and (device-aspect-ratio:5/4) {...} /* 스크린의 너비/높이가 5/4 비율이면 실행 */  
@media all and (device-min-aspect-ratio:5/4) {...} /* 스크린의 너비/높이가 5/4 비율 이상이면 실행 */  
@media all and (device-max-aspect-ratio:5/4) {...} /* 스크린의 너비/높이가 5/4 비율 이하이면 실행 */
```

미디어 쿼리 문법

- color

- 출력장치의 색상 비트수, 출력 장치가 컬러가 아닌경우 '0' 의 값에 대응

```
@media all and (color) {...} /* 출력장치가 컬러를 지원하면 실행 */  
@media all and (color:0) {...} /* 출력장치가 컬러를 지원하지 않으면 실행 */  
@media all and (color:8) {...} /* 출력장치가 8비트 색상이면 실행 */  
@media all and (min-color:8) {...} /* 출력장치가 8비트 이상 색상이면 실행 */  
@media all and (max-color:8) {...} /* 출력장치가 8비트 이하 색상이면 실행 */
```

- color-index

- 출력장치가 색상 색인 테이블을 사용하는 경우 표현할수 있는 색의수

```
@media all and (color-index) {...} /* 출력장치가 색상 색인테이블을 사용하면 실행 */  
@media all and (color-index:0) {...} /* 출력장치가 색상 색인테이블을 사용하면 안으면 실행 */  
@media all and (color-index:256) {...} /* 출력장치가 256색을 지원하면 실행 */  
@media all and (min-color-index:256) {...} /* 출력장치가 256 이상 색을 지원하면 실행 */  
@media all and (max-color-index:256) {...} /* 출력장치가 256 이하 색을 지원하면 실행 */
```

미디어 쿼리 문법

- monochrome

- 출력장치가 흑백인 경우 픽셀당 비트수.

```
@media all and (monochrom) {...} /* 출력장치가 흑백이면 */  
@media all and (monochrom:0) {...} /* 출력장치가 흑백이 아니면 */  
@media all and (min-monochrom:2) {...} /* 출력장치가 흑백이고 2비트 이상이면 실행 */  
@media all and (max-monochrom:2) {...} /* 출력장치가 흑백이고 2비트 이상이면 실행 */
```

- resolution

- 출력장치 의 해상력에 대응한다.

```
@media all and (color-index) {...} /* 출력장치가 색상 색인테이블을 사용하면 실행 */  
@media all and (color-index:0) {...} /* 출력장치가 색상 색인테이블을 사용하면 안으면 실행 */  
@media all and (color-index:256) {...} /* 출력장치가 256색을 지원하면 실행 */  
@media all and (min-color-index:256) {...} /* 출력장치가 256 이상 색을 지원하면 실행 */  
@media all and (max-color-index:256) {...} /* 출력장치가 256 이하 색을 지원하면 실행 */
```

미디어 쿼리 사용 정책

```
/* All Device */
```

모든 해상도를 위한 공통 코드를 작성한다. 모든 해상도에서 이코드가 실행됨

```
/* Mobile Device */
```

768px 미만 해상도의 모바일 기기를 위한 코드를 작성한다. 모든 해상도에서 이 코드가 실행되며, 미디어 쿼리를 지원하지 않는 모바일 기기를 위해 미디어 쿼리 구문을 사용하지 않는다.

```
/* Tablet & Desktop Device */
```

```
@media all and (min-width:768px) {
```

사용자 해상도가 768px 이상일때 이코드가 실행됨. 테블릿과 데스크톱의 공통 코드를 작성한다.

```
}
```

```
/* Tablet Device */
```

```
@media all and (min-width:768px) and (max-width:1024px) {
```

사용자 해상도가 768px 이상이고 1024px 이하일때 이코드가 실행됨. 아이패드 또는 비교적 작은 해상도의 랩탑이나 데스크톱에 대응하는 코드를 작성한다.

```
}
```

```
/* Tablet Device */
```

```
@media all and (min-width:1025px) {
```

사용자 해상도가 1025px 이상일때 이코드가 실행됨. 1025px 이상의 랩탑 또는 데스크톱에 대응하는 코드를 작성한다.

```
}
```


실습

600px 미만은 #000 ,
640px 이상은 #red ,
1025px 이상은 #green

스크린사이즈 별로 배경색을 변경하는 CSS 작성

실습

너비가 600px 이상인 경우 2단 구성을 3단으로 변경