Table

Table

- 테이블은 테이블 형태의 정보를 표현하는데 사용합니다.

- 예를 들어 경기 결과, 기차 시간표 등등

Table

기본적인 테이블의 구조

- <table> : 테이블을 생성할 때 사용됩니다.
- <caption> : 테이블의 제목을 나타냅니다. 항상 <table> 의 첫번째 자식 입니다.
- <tr> : table row의 약자. 테이블은 행 단위로 작성되며 행을 의미합니다.
- <th> : table heading의 약자. 행이나 열의 머리말을 나타냅니다.
    - scope 속성을 통해 방향을 표시합니다. (row, col)
- <td> : table data의 약자. 테이블의 각 셀(테이블 내의 각 블록)은 td요소를 이용해 표현합니다.

Table

```
<table>
   <caption>제목입니다.</caption>
   <tbody>
     <tr>
       <th scope="col">내용1 제목</th>
       <th scope="col">내용2 제목</th>
       <th scope="col">내용3 제목</th>
     </tr>
     <tr>
       <td>내용 1</td>
       <td>내용 2</td>
       <td>내용 3</td>
     </tr>
     <tr>
       <td>내용 1</td>
       <td>내용 2</td>
       <td>내용 3</td>
     </tr>
   </tbody>
</table>
```

제목입니다.

내용1 제목	내용2 제목	내용3 제목
내용 1	내용 2	내용 3
내용 1	내용 2	내용 3

Table

- <thead> : 테이블의 머리말입니다.
- <tbody> : 테이블의 본문입니다.
- <tfoot> : 테이블의 꼬리말입니다.
- colspan : 열병합을 해주는 스타일 속성입니다.
- rowspan : 행병합을 해주는 스타일 속성입니다.

Table

- <col> : 컬럼 단위로 값을 주고 싶을 때 사용하는 태그입니다.

- <colgroup> : col 태그의 집합입니다.

Table

테이블 스타일링

- **이제는 쓰지 말아야할 속성들 : width, bgcolor, border, cellspacing, cellpadding**

- 이러한 HTML 속성들은 이제 사용되지 않습니다.

- 스타일링은 CSS를 이용해주세요! (border-collapse, text-align, vertical-align 등등)

만들어 봅시다!

3-1반 역사성적표

이름	반	번호	점수
한재현	1반	1번	70점
랩몬		2번	90점
슈가		3번	93점
평균			84점

http://jsbin.com/simaronexi/edit?html,css,output

Form

Form

- 유저에게 데이터를 입력받아 서버로 전송하거나 다른 일을 처리하기 위한 입력 폼을 정의합니다.

- <form action="데이터를 보낼 url">

Form

- **input : 유저에게 데이터를 입력 받습니다. Type 속성에 따라 다양한 일을 합니다.**

```
<form>

    <input type="text">

    <input type=“password">

</form>
```

Form

- **Label : input 의 용도를 정의합니다.**

```
<form>
    <label for="text">텍스트를 입력하세요</label>
    <input id="text" type="text">
    <label for="pw">패스워드를 입력하세요</label>
    <input id="pw" type="password">
</form>
```

Form

- **type=“radio”. 하나만 선택할 수 있는 input**

```
<form>
    <h3>내 인생 최고의 영화</h3>
    <input id="r_btn" type="radio" name=“movie">
    <label for="r_btn">라따뚜이</label>
    <input id="r_btn2" type="radio" name=“movie">
    <label for="r_btn2">배트맨</label>
</form>
```

Form

- **type="checkbox". 중복하여 선택할 수 있는 input**

```
<form>
    <h3>내 인생 최고의 영화(중복 가능)</h3>
    <label for="r_btn">라따뚜이</label>
    <input id="r_btn" type="checkbox" name="movie">
    <label for="r_btn2">배트맨</label>
    <input id="r_btn2" type="checkbox" name="movie">
</form>
```

Form

- **<fieldset> : input 의 그룹을 나타낼 때 사용합니다.**
- **<legend> : fieldset 의 제목을 나타냅니다.**

```
<form>
    <fieldset>
      <legend>내인생 최고의 영화(중복 가능)</legend>
      <label for="r_btn">라따뚜이</label>
      <input id="r_btn" type="checkbox" name="movie">
      <label for="r_btn2">배트맨</label>
      <input id="r_btn2" type="checkbox" name="movie">
    </fieldset>
</form>
```

Form

- **<button> : 유저가 클릭 할 수 있는 버튼입니다.**

<form>

  <button type=“button”></button> : 기본 타입. 아무런 기능이 없습니다.

  <button type=“submit”></button> : 폼의 데이터를 서버로 전달하는 기능을 합니다.

  <button type=“reset"></button> : 폼의 데이터를 리셋합니다.

</form>

Form

- **<select> : 드롭다운 메뉴를 만듭니다.**
- **<option> : 드롭다운 메뉴의 데이터입니다.**

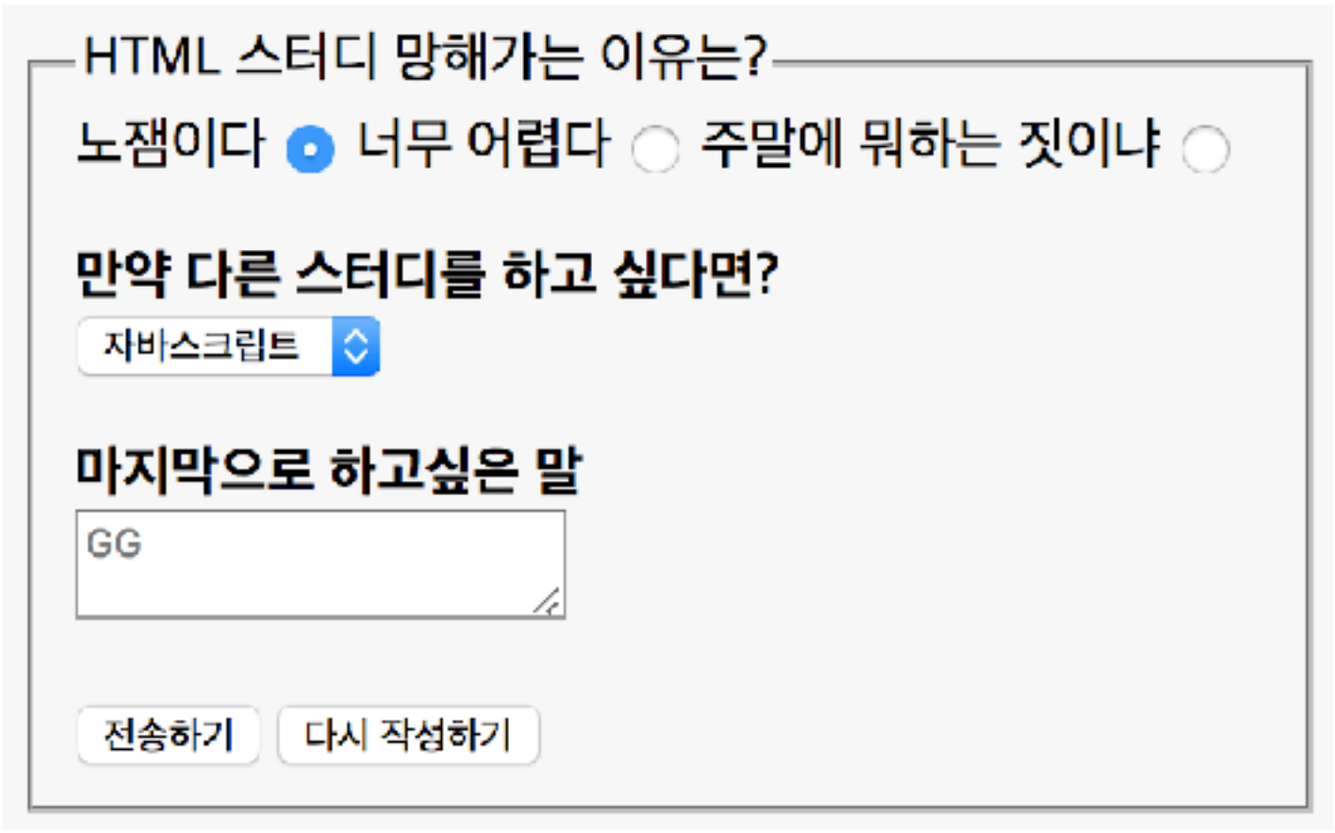
```
<select>
        <option value="콜라">콜라</option>
        <option value="사이다">사이다</option>
        <option value="환타">환타</option>
        <option value="스프라이트">스프라이트</option>
</select>
```

Form

- **<textarea> : 여러줄의 데이터를 입력 받습니다.**

```
<form>
        <textarea></textarea>
</form>
```

만들어 봅시다!



http://jsbin.com/rexifagihe/edit?html,output

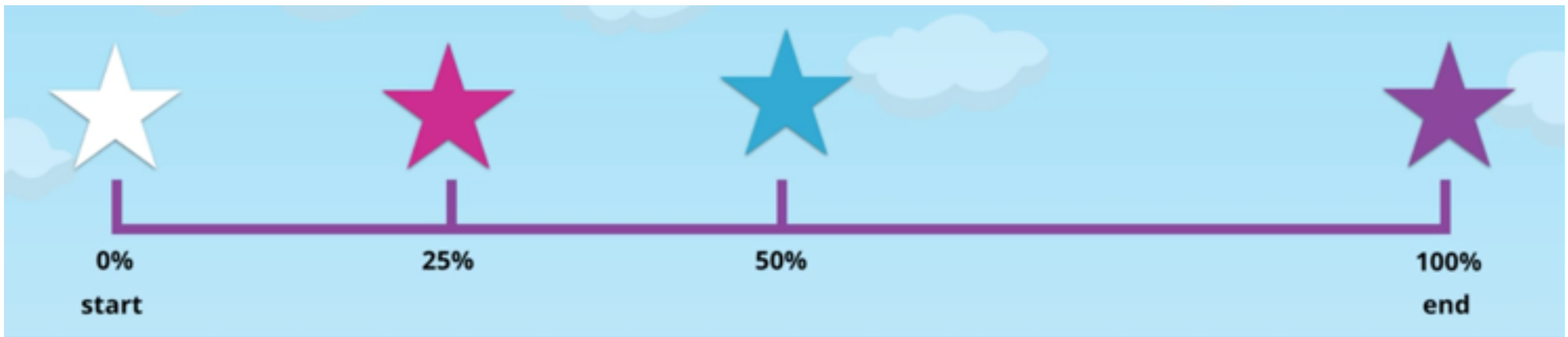
CSS Animation

원하는 속성을 시간에 따라 어떻게 변화할지 선언한 일련의 목록

이거 어디서 본것 같은데?

CSS Animation

Transition 과의 차이는 뭘까?



Key를 통해 정확히 언제 어떠한 변화가 있을지
단계적으로 설정이 가능합니다!

CSS Animation

1. 애니메이션을 만든다!

@keyframes 이름{
  from(시작 시점){
    움직임
  }
  to(끝나는 시점){
    움직임
  }
}

2. 애니메이션을 등록한다!

div{animation: 이름 시간 딜레이 방향 반복 움직임}

CSS Animation

애니메이션 만들기!

애니메이션 방향

1. **normal** : 기본값
2. **reverse** : 반대 방향
3. **alternate** : 애니메이션이 한바퀴 반복될 때마다 반대방향으로 움직임
4. **alternate-reverse** : 애니메이션이 한바퀴 반복될 때마다 반대방향으로 움직이지만 첫번째 재생될때 역재생

애니메이션 반복 : infinite

애니메이션 움직임

- 키워드 혹은 베지어 곡선(transition과 동일)

media query
(반응형 디자인&레티나 대응)

Media-query

- 특정 조건에서 특정 스타일만 적용되도록 만들어줍니다.

- @media 논리연산자 미디어타입 (조건) { 적용할 스타일 }

Media-query

미디어 타입: all, print, screen

- all : 모든 디바이스
- print : 프린터 미리보기 화면 혹은 인쇄물

```
@media print{
    abbr::after{
      content:attr(title)}
  }
```

- screen : 컴퓨터 화면

Media-query

논리 연산자 : only, and, not, 컴마

- and : 조건을 모두 만족하는 경우에만 스타일을 적용합니다.
- not : 조건을 반전하는 경우 스타일을 적용합니다.
- 컴마(,) : 조건중 하나라도 만족하는 경우 스타일을 적용합니다.
- only : 미디어쿼리를 지원 하는 브라우저에서만 작동하게 합니다.
  - (미디어 쿼리 css3 버젼은 IE9 부터 지원하지만 사실 미디어 쿼리 자체는 이미 IE6부터 지원 합니다.(https://www.w3.org/TR/CSS2/media.html#at-media-rule) 하지만 이땐 아직 논리 연산자를 지원하지 않기 때문에 오류를 미연에 방지하고자 only 키워드가 탄생하게 됩니다.)

만들어 봅시다!

미디어쿼리 실습

http://jsbin.com/yipefotexi/edit?html,css,output

https://material.io/guidelines/layout/responsive-ui.html#responsive-ui-breakpoints

레티나 대응

레티나 대응의 원리

사전지식
1. 화소(pixel) : 화면을 구성하는 가장 기본이 되는 단위

2. ppi(pixel per inch) : 화소 밀도. 인치당 화소가 들어가는 양을 나타낸다.

3. dp 혹은 dip(dencity-independent pixel) : 밀도독립화소. 기기의 물리적인 디스플레이 해상도에 영향을 받지 않고 독립적으로 크기를 지정할 수 있도록 하는 가상적 화소(Pixel)의 단위

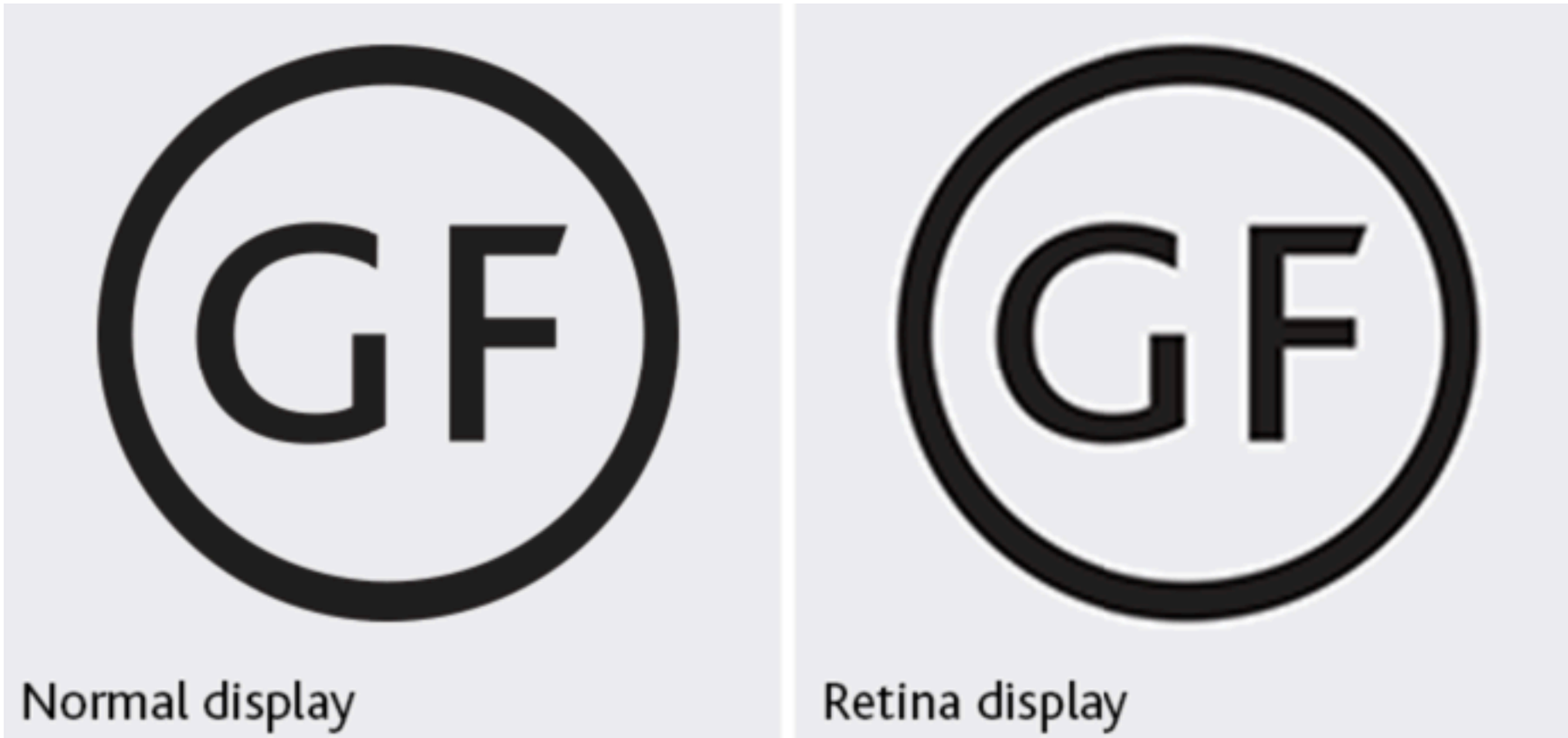
4. 물리픽셀 : 디바이스가 실제로 처리할 수 있는 화소의 기본단위

5. 논리픽셀 : css에서 표현하는 화소의 기본 단위

레티나란??
—> 특정한 시야거리에서 인간의 눈으로는 화소를 구분할 수 없는 화소 밀도(300 PPI가 넘을 경우)를 가진 애플 LCD 제품의 브랜드 이름

레티나 대응

문제는??



일반 디스플레이에서는 잘 랜더링 되던 이미지가 레티나 디스플레이에서 흐려지는 현상 발생!

레티나 대응

원인은 무엇일까?

- 레티나(고해상도 화면)로 넘어오면서 논리픽셀과 물리픽셀의 차이가 발생. 그러나 브라우저는 css에서 정의한 픽셀만큼 이미지를 랜더링 해야하기 때문에 원래는 물리픽셀에 맞게 랜더링된 이미지가 논리픽셀 만큼 커져버리게 된다. 그로 인해 발생

	iPhone	iOS	3.5 in	1.9 × 2.9 in	3 : 2	320 × 480 dp	320 × 480 px
	iPhone 4	iOS	3.5 in	2.0 × 2.9 in	3 : 2	320 × 480 dp	640 × 960 px

레티나 대응

해결책은?

- 간단해요! 우리가 화면에 그리고자 하는 사이즈의 딱 2배 되는 이미지를 준비하면 됩니다.

http://snoweyes1.tistory.com/7

https://www.slideshare.net/JieunLee5/v2-46063246