

Cascading Style Sheet

CSS

❖ CSS

- ✓ CSS는 웹 페이지에 "디자인"이라는 시각적 가치를 부여하기 위한 언어
- ✓ CSS는 HTML의 Presentation 모듈을 대체하는 언어로서 개발
 - 기존 HTML 기능 확장 - HTML이 가지는 한계점(소스의 구성, 레이아웃 배치) 극복: <http://www.csszengarden.com/>
 - 줄어든 작업량: 웹 페이지에 자주 적용되는 디자인 부분들을 미리 정리하고 모아서 정의한 뒤 필요로 할 때마다 가져다 적용할 수 있음
 - 접근성이 좋은 사이트 제작: 다양한 기기에 맞게 탄력적으로 변경되는 문서 제작
- ✓ CSS의 표준 : Level CSS2.1, 3
 - 대다수의 웹 브라우저는 CSS2.1 지원
 - IE의 경우 버전 7에서 CSS2.1 지원 시작
 - 현재 상태: <https://www.w3.org/Style/CSS/current-work>
- ✓ W3C에서는 CSS3를 향후 표준화 하기 위해 노력
- ✓ 동작
 - HTML로 Mark Up 한 문서로 시작
 - 특정한 요소를 표시하는 방법에 대한 스타일 규칙을 작성
 - 문서에 스타일 규칙을 첨부

CSS

❖ CSS3

- ✓ 이전 CSS2 보다 정교하고 화려한 화면을 구성할 수 있고 애니메이션까지 지원
- ✓ 배경이나 글꼴, 박스 모델 등 수십 개 기능을 주제 별로 별도의 규약을 지정 – CSS 모듈
- ✓ 모듈 별로 개발이 되다 보니 브라우저 별로 지원 여부가 다름
- ✓ 표준 규약이 아닌 속성들은 접두사를 붙여서 브라우저 별로 지원
 - -webkit- : 사파리, 크롬
 - -moz-: 모질라, 파이어폭스
 - -o-: 오페라
 - -ms-: 마이크로소프트 인터넷 익스플로러
- ✓ 접두사를 붙여서 만들 때는 접두사를 붙인 내용을 먼저 작성하고 표준이 지원할 내용을 마지막에 작성
- ✓ 접두사를 붙이는 것이 번거로울 때는 <http://leaverou.github.io/prefixfree/> 사이트에 접속해서 자바스크립트 파일을 받아서 스타일시트를 적용하기 전에 링크를 설정하면 됨

CSS

❖ CSS3 에 추가된 기능

- ✓ 보더: 측면, 보더 이미지, 라운드 코너에 여러 개의 보더 색상 적용 가능
- ✓ 배경: 단일 엘리먼트에 여러 배경 추가 가능
- ✓ 색상: 색상 투명 설정, 배경의 그라디언트, HSL 색상 값
- ✓ 텍스트: 텍스트 그림자, 오버플로우, 워드랩
- ✓ 변형: 2D, 3D 공간에서의 스케일, 왜곡, 이동, 회전
- ✓ 트랜지션 효과
- ✓ 박스: 그림자 추가, 사용자가 크기 조절 가능한 박스 배치, 오버플로우, 외곽선 위치 조절 값

CSS 적용 방식

❖ 스타일시트 적용 방식

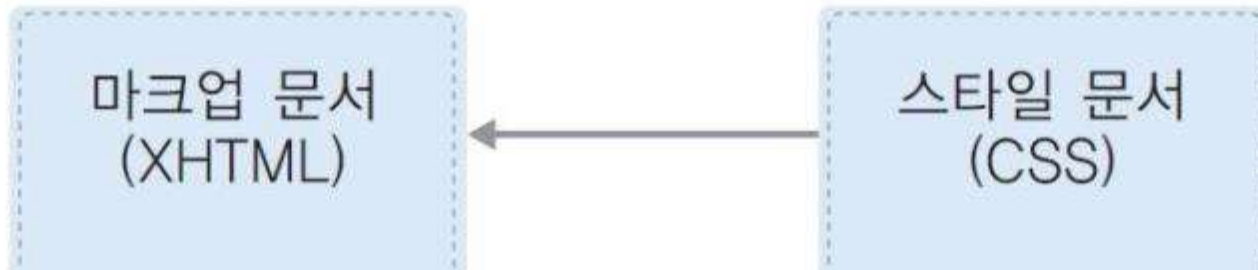
- ✓ External Style Sheet (HTML 문서와 별도로 작성)
 - Linked Style
 - Import Style
- ✓ Internal Style Sheet (HTML 문서 내의 별도의 영역에 작성)
- ✓ Inline Style Sheet (태그 안에 적용)



CSS 적용 방식

- ❖ 외부 스타일 시트 :CSS 파일을 독립적으로 생성하여 웹 문서에 삽입하는 방법
<link rel="stylesheet" href="스타일시트 파일의 url">

외부에 독립적으로 생성한
default.css를 마크업 문서에 연결(link)



CSS 적용 방식

- ❖ 외부 스타일 시트 :CSS 파일을 독립적으로 생성하여 웹 문서에 삽입하는 방법

```
<style type="text/css">
```

```
  @import url("filename"); 또는 @import "filename"
```

```
</style>
```

외부에 독립적으로 생성한 default.css를
마크업 문서에서 가져오기(import)

마크업 문서(XHTML)

```
<style>  
@import  
</style>
```

스타일(CSS)
default.css

base.css 내에 default.css 파일을
가져오기(import)

스타일 문서(CSS)

```
@import
```

스타일(CSS)
default.css

CSS 적용 방식

- ❖ Internal Style Sheet(내부 스타일시트): HTML 문서에 CSS를 선언하기 위한 방법으로 style 요소(Element)를 사용

마크업 문서 내에서
CSS 직접 선언하기

마크업 문서(XHTML)

```
<style>  
    CSS 코드  
</style>
```


CSS 적용 방식

- ❖ Inline Style Sheet(인라인 스타일 시트): 특정 요소에 직접 style 속성을 이용하여 CSS 스타일을 적용하는 방법으로 권장하지 않음

마크업 요소(Element)에
직접 CSS 스타일 적용하기

마크업 문서(XHTML)

```
<h1 style=" CSS 코드"></h1>
```

CSS 작성 방법

❖ 기본 형식

✓ selector { property : value ; }

- selector (선택자): 스타일을 적용하는 대상
- property (속성): 스타일의 종류
- value (값): 속성이 가질 수 있는 값
- 사용 예)

```
body { color : gray ; font-size : small ; }
```

✓ 규칙

- 속성과 값의 쌍으로 이루어져야 하며 콜론으로 구분
- 스타일시트 선언에 키워드를 사용할 수 있는데 키워드에는 큰 따옴표나 작은 따옴표를 붙이면 안됨 - green 과 "green"은 전혀 다른 의미
- 여러 개의 속성을 설정하는 경우에는 ; 으로 구분
- 여러 개의 속성을 설정하는 경우 한 줄에 작성 가능하고 여러 줄에 나누어 작성하는 것도 가능

CSS 작성 방법

❖ Comment

- ✓ 소스 코드 관리와 가독성을 높이기 위한 문자열
 - ✓ `"/**"와 "*/"` 사용
 - ✓ 주석으로 작성한 내용은 스타일에 영향을 미치지 않음
- ```
/* comment */
```

# CSS Selector

## ❖ 기본 선택자

- ✓ \*(asterisk): 전체 선택자로 모든 요소에 적용

```
* {
 margin : 0 ;
 padding : 0 ;
}
```

- ✓ tag이름: 태그 선택자

```
p { color : orange ; }
```

- ✓ .class이름: class 선택자

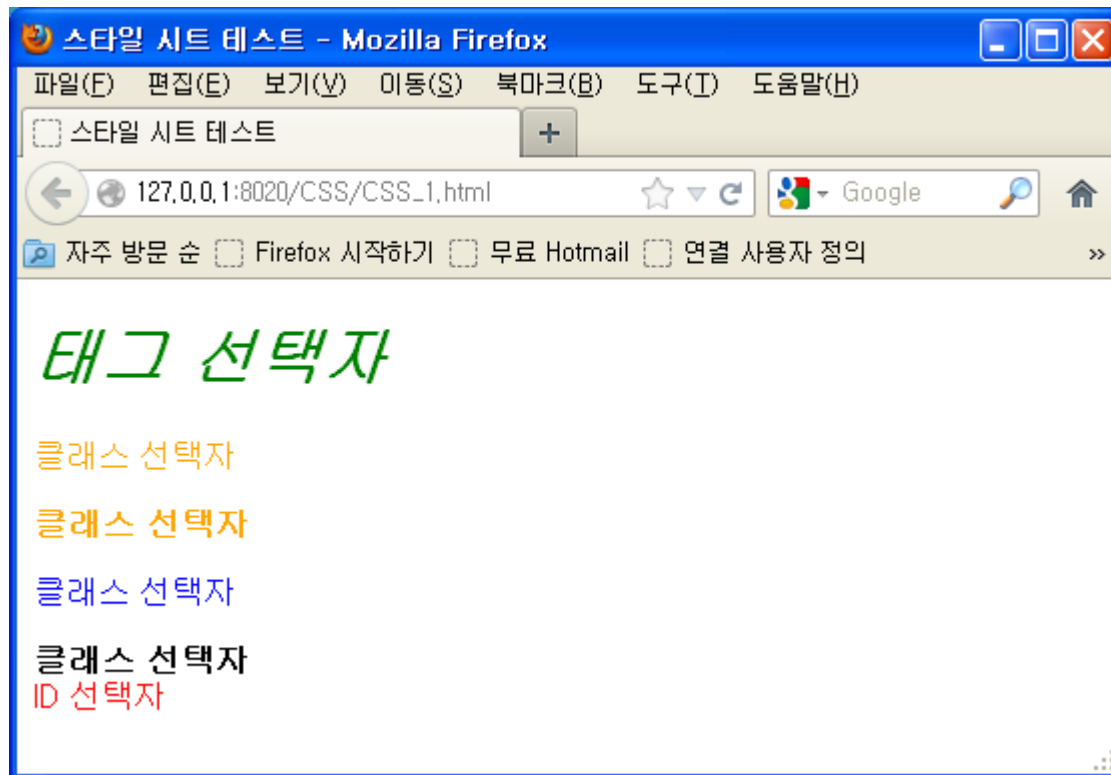
```
.note { color : blue ; }
```

- ✓ #id: id 선택자

```
#gnb {
 list-style-type : none ;
}
```

# CSS Selector

## ❖ 기본 선택자



# CSS Selector

## ❖ 기본 선택자

### ✓ basicselector.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```
<title>기본 선택자</title>
```



# CSS Selector

## ❖ 기본 선택자

### ✓ basicselector.html

```
<style type="text/css">
 h1 {
 font-style: italic;
 color: green;
 }

 .class1 {
 color: orange;
 }

 p.class2 {
 color: blue;
 }

 #name {
 color: red;
 }
</style>
</head>
```

# CSS Selector

## ❖ 기본 선택자

### ✓ basicselector.html

```
<body>
 <h1>태그 선택자</h1>
 <p class="class1">클래스 선택자</p>
 <b class="class1">클래스 선택자
 <p class="class2">클래스 선택자</p>
 <b class="class2">클래스 선택자
 <div id="name">ID 선택자</div>
</body>
</html>
```



# CSS Selector

## ❖ 속성 선택자

✓ 선택자[속성]: 속성을 가지는 선택자에 적용

- disabled 속성을 가진 요소에 적용

```
[disabled]{
 color:red;
}
```

✓ 선택자[속성=값]: 속성의 값이 일치하는 선택자에 적용

- type이 password 인 요소에 적용

```
[type="password"]{
 color:red;
}
```

✓ 속성 선택자는 부분 일치 속성도 사용 가능

- A^=B: A 속성의 값이 B로 시작하는
- A\$=B: A 속성의 값이 B로 끝나는
- A\*=B: A 속성의 값이 B를 포함하는

# CSS Selector

## ❖ 복합 선택자

- ✓ 여러 선택자를 서로 조합해서 사용 가능
- ✓ 선택자 들은 적용 범위에 따라 일치 선택자(Basic Combinator), 하위 선택자(Descendant Compinator), 자식 선택자(Child Combinator), 인접 형제 선택자(Adjacent Sibling Combinator), 일반 형제 선택자(General Sibling Combinator) 등 이 있음
- ✓ 일치 선택자
  - 선택자를 두 개 이상 붙여 작성하는 것으로 사용된 선택자를 동시에 만족시키는 소를 선택 - span 태그 이면서 class 가 orange 인 요소 선택

```
span.orange{
 color:red;
}
```

## ✓ 자식 선택자

- 선택자 > 선택자 (왼쪽 부등호로 구분)
- 부모 요소 바로 뒤의 자식 요소(자식 요소 이하를 포함하지 않음)를 대상으로 스타일을 적용
- wcag라는 id 안에 있는 첫번째 레벨의 p 태그

```
.wcag > p {
 border : 1px solid red ;
}
```

# CSS Selector

## ❖ 복합 선택자

### ✓ 하위 선택자

- 선택자 와 선택자 (공백으로 구분)
- 부모 요소에 포함된 모든 자식 요소를 대상으로 스타일을 적용
- class 가 note 인 요소 안에 포함된 모든 p 태그

```
.note p {
 color : "orange" ;
}
```

### ✓ 인접 형제 선택자

- 선택자 + 선택자 (플러스 기호로 구분)
- 문서 트리 구조 상 동일한 부모 요소를 가진 병렬 관계에 있는 요소 가운데 나중에 등록한 바로 뒤 요소에만 스타일을 적용

```
h1 + h2 {
 color : gray ;
}
```

# CSS Selector

## ❖ 복합 선택자

### ✓ 일반 형제 선택자

- 선택자 ~ 선택자
- 문서 트리 구조 상 동일한 부모 요소를 가진 병렬 관계에 있는 요소 가운데 나중에 있는 모든 요소에 스타일을 적용

```
.orange ~ li {
 color : gray ;
}
```

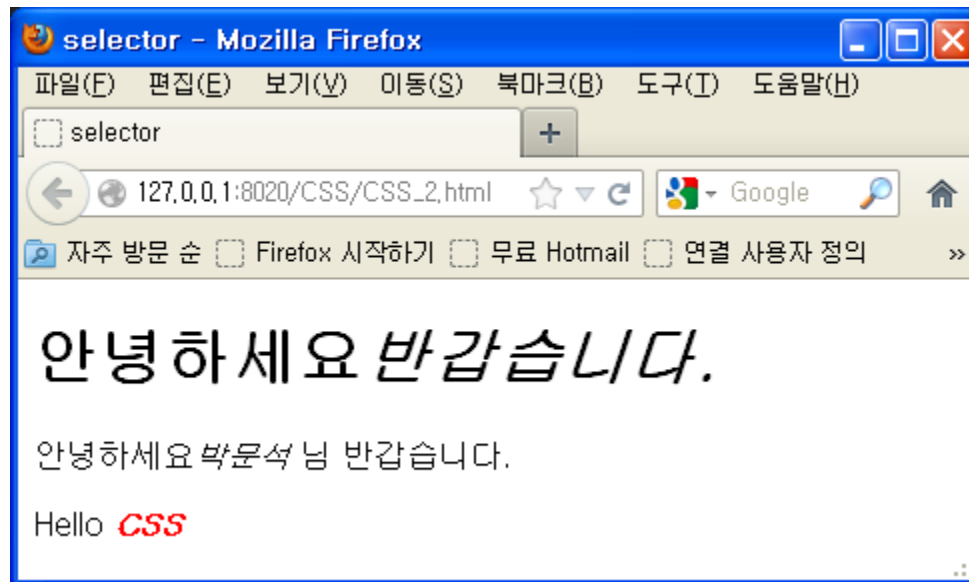
### ✓ 그룹 선택자

- 선택자와 선택자를 콤마(,)로 구분하여 그룹화 하고 복수의 요소에 같은 스타일을 적용
- class 선택자나 id 선택자, 속성 선택자, 하위 선택자, 자식 선택자, 인접 선택자 등을 지정할 때도 콤마(,)로 구분하여 그룹화 가능

```
div > p, #wrap, p.note, blockquote p, h1[title] {
 color : red ;
}
```

# CSS Selector

- ❖ 복합 선택자
  - ✓ combinatorselector.html



# CSS Selector

## ❖ 복합 선택자

### ✓ combinatorselector.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>selector</title>
<style type="text/css">
 p+p em {
 color: red;
 font-weight: bold;
 }
</style>
</head>
```

# CSS Selector

## ❖ 복합 선택자

### ✓ combinatorselector.html

```
<body>
 <div class="copy">
 <h1>
 안녕하세요반갑습니다.
 </h1>
 <p>
 안녕하세요박문석 님 반갑습니다.
 </p>
 <p>
 Hello CSS
 </p>
 </div>
</body>
</html>
```

# CSS Selector

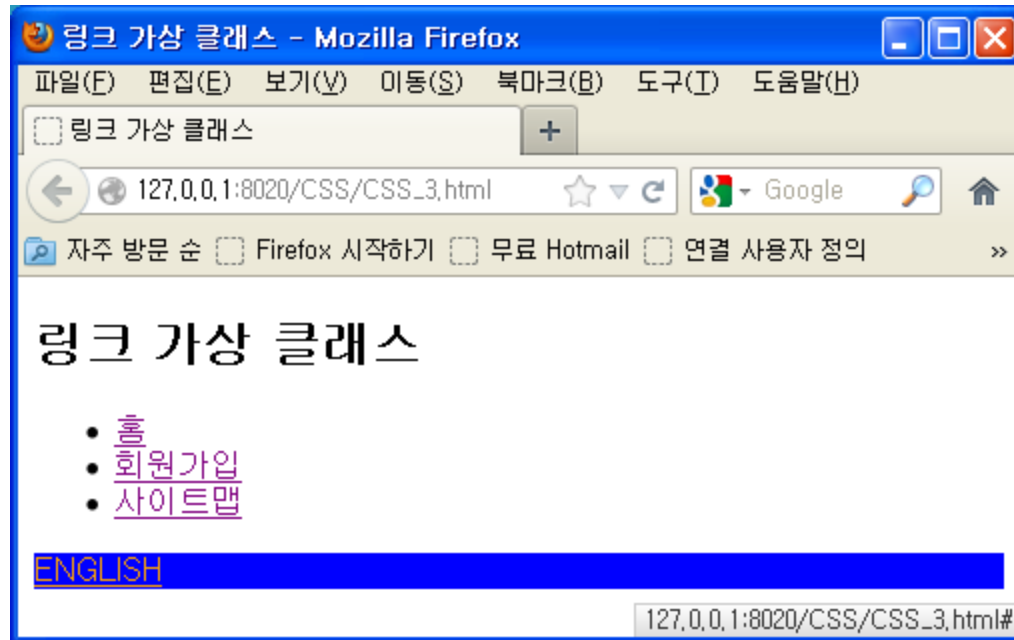
## ❖ 의사 클래스(pseudo-classes)

- ✓ 상황에 따라 스타일을 적용하는 방법으로 선택자 뒤에 : 을 하고 기재
- ✓ Link
  - link
  - visited
- ✓ 사용자 동작
  - hover -> 마우스가 올라오면
  - active -> 클릭하면
  - focus -> 포커스가 오면
- ✓ UI 요소 상태
  - checked -> 체크상태가 되면
  - enabled -> 활성화되면
  - disabled -> 비활성화되면



# CSS Selector

- ❖ 의사 클래스(pseudo-classes)
  - ✓ pseudoclass.html



# CSS Selector

## ❖ 의사 클래스(pseudo-classes)

### ✓ pseudoclass.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 링크 가상 클래스</title>
<style type="text/css">
 a:link {
 color: blue;
 }
 a:visited {
 color: purple;
 }
 a:hover {
 color: orange;
 }
 a:active {
 color: red;
 }
 div:hover {
 background-color: blue;
 }
</style>
</head>
```

# CSS Selector

## ❖ 의사 클래스(pseudo-classes)

### ✓ pseudoclass.html

```
<body>
 <h2> 링크 가상 클래스</h2>

 홈
 회원가입
 사이트맵

 <div>
 ENGLISH
 </div>

</body>
</html>
```

# CSS Selector

- ❖ 의사 클래스(pseudo-classes)
  - ✓ active.html

활성화

비활성화

# CSS Selector

## ❖ 의사 클래스(pseudo-classes)

### ✓ active.html

```
<!DOCTYPE html>
<html>
<head>
<title>가상 선택자</title>
<meta charset="UTF-8" />
<style>
 input:enabled {
 background-color: white;
 }

 input:disabled {
 background-color: gray;
 }

 input:focus {
 background-color: papayawhip;
 }
</style>
</head>
```

# CSS Selector

## ❖ 의사 클래스(pseudo-classes)

### ✓ active.html

```
<body>
 <h2>활성화</h2>
 <input type='text' />
 <h2>비활성화</h2>
 <input type='text' disabled="disabled" />
</body>
</html>
```

# CSS Selector

## ❖ 의사 클래스(pseudo-classes)

### ✓ 구조 선택자

- root: 최상위 요소
- first-child, last-child: 첫 번째 자식 요소 와 마지막 자식 요소
- nth-last-child(n): 뒤에서 n 번째 자식 요소
- nth-child(n): n 번째 자식 요소
- only-child: 자식이 하나인 요소
- not(선택자) : 부정
- empty: 자식이 없는
- first-line, first-letter
- only-child:형제가 없는
- nth-of-type(n), first-of-type, last-of-type, only-of-type(타입이 하나)

### ✓ 기타

- lang(languagecode)  
    \*:lang(en-US) {  
        outline: 2px solid deeppink;  
    }

# CSS Selector

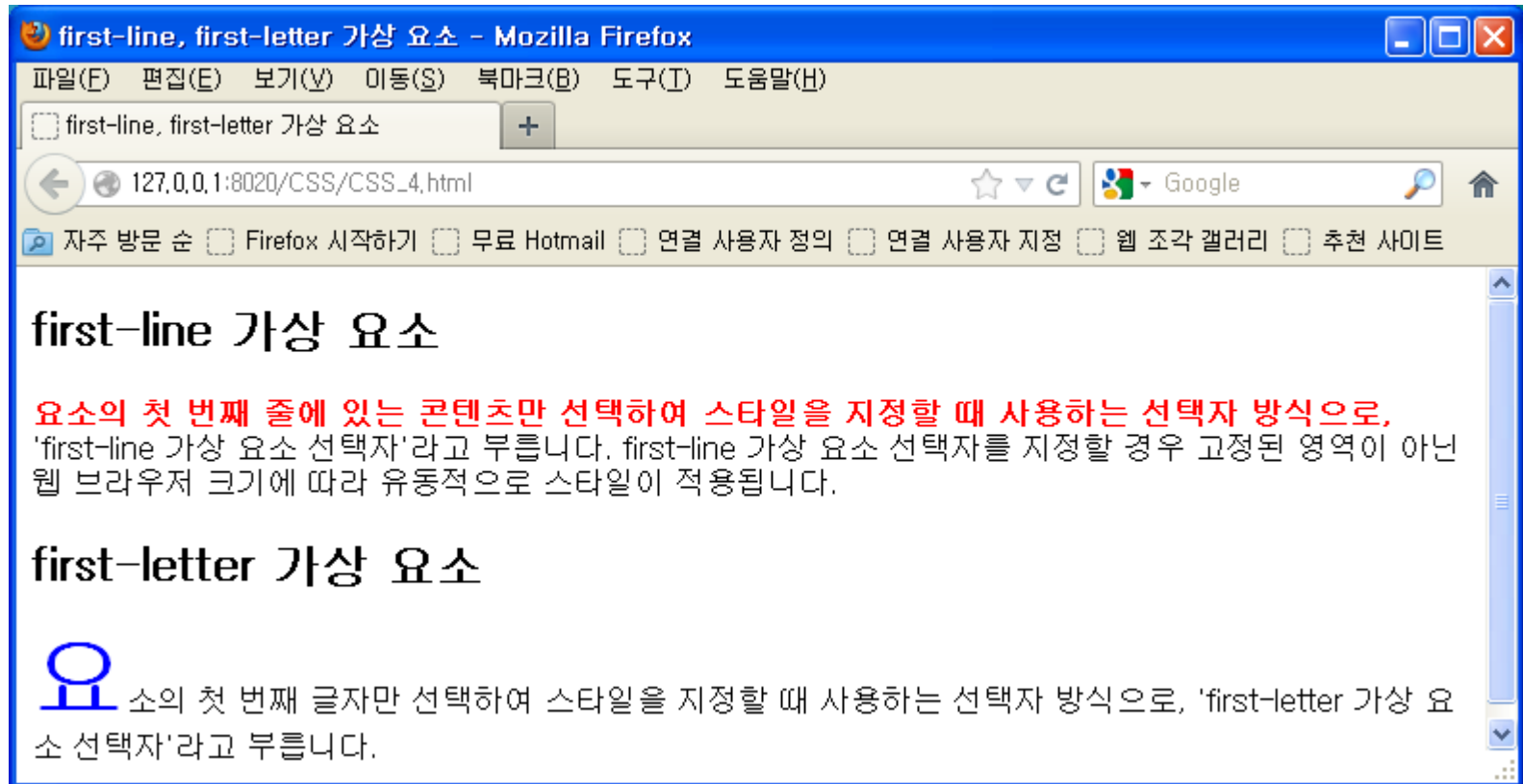
- ❖ 의사 요소(Pseudo-Elements) 선택자
  - ✓ ::first-letter
  - ✓ ::first-line
  - ✓ ::before
  - ✓ ::after
  - ✓ ::selection





# CSS Selector

## ❖ 의사 요소(Pseudo-Elements) 선택자



# CSS Selector

## ❖ 의사 요소(Pseudo-Elements) 선택자

### ✓ pseudo.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>first-line, first-letter 가상 요소</title>
<style type="text/css">
 .line p:first-line {
 color: red;
 font-weight: bold;
 }
 .letter p:first-letter {
 color: blue;
 font-weight: bold;
 font-size: 3em;
 }
</style>
</head>
```

# CSS Selector

## ❖ 의사 요소(Pseudo-Elements) 선택자

### ✓ pseudo.html

```
<body>
```

```
 <h2>first-line 가상 요소</h2>
```

```
 <div class="line">
```

<p>요소의 첫 번째 줄에 있는 콘텐츠만 선택하여 스타일을 지정할 때 사용하는 선택자 방식으로, 'first-line 가상 요소 선택자'라고 부릅니다. first-line 가상 요소 선택자를 지정할 경우 고정된 영역이 아닌 웹 브라우저 크기에 따라 유동적으로 스타일이 적용됩니다.</p>

```
 </div>
```

```
 <h2>first-letter 가상 요소</h2>
```

```
 <div class="letter">
```

<p>요소의 첫 번째 글자만 선택하여 스타일을 지정할 때 사용하는 선택자 방식으로, 'first-letter 가상 요소 선택자'라고 부릅니다.</p>

```
 </div>
```

```
</body>
```

```
</html>
```

# CSS Selector

## ❖ 상속

- ✓ 하위 요소가 상위 요소의 값을 물려 받느냐 아니냐를 나타내는 것
- ✓ 요소의 상속은 되는 것이 있고 안 되는 것이 있음
  - 상속이 가능한 스타일은 대부분 폰트 와 관련된 속성: font-style, font-weight, font-size, line-height, font-family, color, text-align 등
- ✓ 상속되지 않는 경우 inherit 키워드를 이용하여 강제로 상속시킬 수 있음
  - <p>부모 요소에 적용한 스타일이 자식 요소에 계속되는 것을
  - <a href="inherit.html">상속(inherit)</a>라고 한다.

</p>

```
p {
 border : 1px solid red ;
}
a {
 border : inherit ;
}
```

- ✓ 상속된 스타일을 재정의(Overriding)하고 싶다면 원하는 선택자에 적용을 원하는 스타일을 정의

# CSS Selector

## ❖ 우선 순위

- ✓ 스타일의 충돌: 두 개 이상의 규칙이 동일한 한 개의 요소에 적용
- ✓ 스타일은 단계적으로 적용: 특정도가 높은 순서대로 적용
- ✓ 동일한 선택자에 여러 개의 스타일이 설정되면 마지막에 지정된 스타일이 마지막에 적용

```
<link rel="stylesheet" type="text/css" href="default.css" />
```

```
<style type="text/css">
```

```
 @import url(base.css);
```

```
 p { color : gray ; }
```

```
</style>
```

```
<p style="color : blue ;">스타일의 우선순위</p>
```

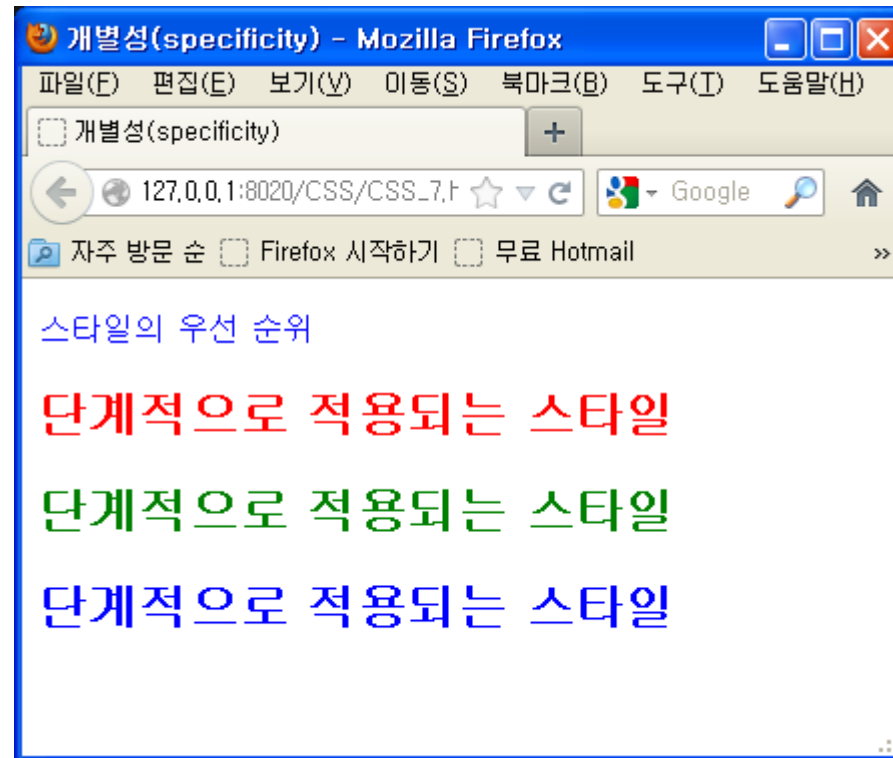
# CSS Selector

## ❖ 우선 순위

선택자	사용 예	특정도
type 선택자	p	1
class 선택자	.note	10
id 선택자	#gnb	100
inline 스타일	style="color:red"	1000
가상 요소	:first-line	1
가상 클래스	:hover	10

# CSS Selector

## ❖ 우선 순위



# CSS Selector

## ❖ 우선 순위

- ✓ basic.css 파일을 생성하고 작성  
p{color:red;}





# CSS Selector

## ❖ 우선 순위

- ✓ priority.html 파일을 생성하고 작성

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<title>개별성(specificity)</title>
```

# CSS Selector

## ❖ 우선 순위

- ✓ priority.html 파일을 생성하고 작성

```
<style type="text/css">
 @import url(base.css);
```

```
p {
 color: gray;
}
```

```
#specifi {
 color: blue;
}
```

```
.cascade {
 color: green;
}
```

```
h2 {
 color: red;
}
</style>
</head>
```

# CSS Selector

## ❖ 우선 순위

- ✓ priority.html 파일을 생성하고 작성

```
<body>
 <p style="color: blue;">스타일의 우선 순위</p>
 <h2>단계적으로 적용되는 스타일</h2>
 <h2 class="cascade">단계적으로 적용되는 스타일</h2>
 <h2 id="specifi">단계적으로 적용되는 스타일</h2>
</body>
</html>
```

# 단위(UNIT)

## ❖ 단위

- ✓ CSS에서 사용할 수 있는 단위(units)는 크게 절대 단위 와 상대 단위 가 있음
- ✓ 단위 종류
  - 절대 단위
    - ◆ px(픽셀 - 1인치의 1/96)
      - 디바이스가 표현할 수 있는 화면의 점을 의미
      - 1920 X 1080 모니터 해상도에서 바로 이 1920, 1080이라는 숫자들이 가로와 세로에 들어가는 픽셀의 수를 의미
      - 픽셀은 화면의 확대/축소 등에 따라 달라질 수 있어 절대 단위가 아닌 상대 단위로 보기도 함
    - ◆ pt(포인트:1인치의 1/72)
    - ◆ cm(센티미터)
    - ◆ mm(밀리미터)
    - ◆ pc(파이 카:12pt)
    - ◆ in(인치)

# 단위(UNIT)

## ❖ 단위

### ✓ 단위 종류

#### ● 상대 단위

◆ px(픽셀)

◆ %(백분율)

◆ em

- font-size 값을 기준으로 크기를 설정
- 글꼴 크기가 16px 이라고 하면 1em 은 16px
- rem은 최상위 요소의 글꼴 크기 기준

◆ 뷰 포트 기중

- vw: 뷰 포트 폭의 1/100)
- vh: 뷰 포트 높이의 1/100
- vmin 과 vmax

◆ ex: 소문자 x의 높이 또는 em의 절반

◆ ch: 숫자 0의 너비 크기

#### ● 각도

◆ deg

◆ rad

# 단위(UNIT)

## ❖ 단위

- ✓ 스크린에 표시할 때는 상대 단위가 적합하고 인쇄를 할 때는 절대 단위가 적합
- ✓ 특별히 글자 크기를 지정하지 않으면 브라우저는 기본 크기대로 표시를 하게 되는데 브라우저의 기본 크기는 12pt, 16px, 1em, 100% 와 같이 표시
- ✓ 웹에서는 글자 크기를 지정할 때 px이나 em을 많이 사용하는데 px 단위로 지정할 경우 항상 일정한 크기를 가짐
- ✓ N-Screen 시대에서는 글꼴이나 표시하는 화면에 따라 적절하게 글꼴 표시가 되는 em 단위를 사용하는 것이 바람직
- ✓ 단위 변환: [www.pxtoem.com](http://www.pxtoem.com)

# Typography

❖ Typography: 문자 또는 활판적 기호를 중심으로 한 2차원적 표현

✓ 종류

- font-family
- @font-face
- font-size
- line-height
- font-weight
- font-style
- font-variant
- font
- text-indent
- text-align
- vertical-align
- text-decoration
- text-transform
- letter-spacing
- white-space



# Typography

## ❖ font-family

- ✓ 글꼴 패밀리를 지정하고자 할 때 사용
  - 값 : [<글꼴 패밀리명>|<대표 패밀리명>]
  - font-family : 돋움, "맑은 고딕", sans-serif
- ✓ 기본값은 브라우저에 따라 다름
- ✓ 모든 요소에 적용이 가능하며 상속됨
- ✓ 글꼴 이름에 공백이 포함되어 있으면 "" 안에 기재
- ✓ 여러 개의 글꼴을 적용할 경우에는 쉼표로 구분
- ✓ 여러 개의 글꼴을 적용하는 이유는 브라우저가 자신이 사용할 수 있는 글꼴만 표시할 수 있기 때문에 첫번째 글꼴을 찾아서 적용해보고 안되면 두번째 글꼴을 찾아서 적용하도록 하기 위해서
- ✓ 일반적인 글꼴 패밀리를 제외하고 모든 글꼴 이름은 첫 글자를 대문자로 표시 – arial이 아니라 Arial
- ✓ 영문 기본 글꼴
  - serif: 끝에 꾸미는 세리프(삐침) 모양이 있는 글꼴(Times new roman, Georgia)
  - sans-serif: 끝에 꾸미는 모양이 없는 글꼴(Verdana, Aria, Trebuchet MS)
  - cursive: 손 글씨 같은 장식용 글꼴(Comic sans)
  - fantasy: 주로 제목이나 디스플레이 용도로 사용되는 굵은 글꼴(Impact)
  - monospace: 자간이 일정한 글꼴(Andale, mono, Courier new)



# Typography

## ❖ @font-face

- ✓ 웹 문서를 작성할 때 웹 문서 안에 글꼴 정보도 저장했다가 사용자가 웹 문서에 접속하면 글꼴을 사용자 시스템으로 다운로드 하게 하는 요소
- ✓ 컴퓨터에서 사용하는 글꼴은 true type 유형이고 확장자는 ttf 인데 이 글꼴들은 모바일 용으로는 부적합
- ✓ IE 8.0 이하 버전에서는 Embedded Open Type 그리고 그 이외 브라우저에서는 Web Open Font Format 형태의 글꼴을 지원하기 때문에 적용이 어려웠는데 지금은 글꼴을 하나로 묶어서 패키지 형태로 사용하는 것이 가능
- ✓ @font-face{
  - font-family:글꼴 이름
  - src: url(글꼴 파일 경로) format(파일 유형)
- ✓ [www.google.com/webfonts](http://www.google.com/webfonts) 에 가면 구글이 제공하는 글꼴을 무료로 사용 가능하고 유료로 사용할 수 있는 사이트로는 Adobe([www.typekit.com](http://www.typekit.com)) 나 Fonts.com([www.fonts.com](http://www.fonts.com)) 등이 있음
- ✓ 최근에는 ttf 파일의 웹의 폰트로 변환해주는 사이트도 있음

# Typography

❖ typography1.html

## **Black Goose Bistro • Summer Menu**

Baker's Corner, Seekonk, Massachusetts

Hours: Monday through Thursday: 11 to 9, Friday and Saturday; 11 to midnight



# Typography

## ❖ typography1.html

```
<!DOCTYPE html >
<html>
<head>
<meta charset="utf-8">
<title>Black Goose Bistro Summer Menu</title>
<link href='http://fonts.googleapis.com/css?family=Marko+One'
 rel='stylesheet'>
<style>
 body {
 font-family: Verdana, sans-serif;

 h1 {
 font-family: "Marko One", Georgia, serif;
 }
 }
</style>
</head>
```

# Typography

## ❖ typography1.html

```
<body>
```

```
 <div id="info">
```

```
 <h1>Black Goose Bistro • Summer Menu</h1>
```

```
 <p>
```

```
 Baker's Corner, Seekonk, Massachusetts
 Hours:
```

```
 Monday through Thursday: 11 to 9, <span
```

```
class="label">Friday
```

```
 and Saturday; 11 to midnight
```

```
 </p>
```

```
 </div>
```

```
</body>
```

```
</html>
```

# Typography

## ❖ font-size

- ✓ 글자 크기를 지정할 경우 사용
- ✓ 모든 요소에게 적용되며 상속됨

값 : 키워드 | 길이 | 퍼센트 | inherit

- font-size: small; -> xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, inherit
- font-size: 12px; -> 절대적인 크기
- font-size: 1.5em; -> 부모 요소의 font-size에 대한 상대적인 크기
- font-size: 150%; -> 부모 요소의 font-size에 대한 상대적인 크기

# Typography

## ❖ font-weight

- ✓ 글꼴의 굵기를 표현할 때 사용
- ✓ 모든 요소에 사용 가능하며 상속됨

값 : 키워드 | 100 ~ 900 | inherit

- font-weight : normal ; -> normal, bold, bolder, lighter, inherit
- font-weight : 400 ; -> 100, 200, 300, 400, 500, 600, 700, 800, 900



# Typography

## ❖ font-style

- ✓ 글꼴의 스타일을 지정할 때 사용
- ✓ 모든 요소에 사용이 가능하며 상속됨
  - 값 : normal | italic | oblique | inherit
    - font-style : normal
    - font-style : italic
    - font-style : oblique
- ✓ Italic은 필기체 형태로 된 별도의 서체 디자인이고 oblique 텍스트는 일반적인 글꼴 디자인



# Typography

## ❖ font-variant

- ✓ 영문 글꼴의 소문자를 작은 대문자로 변형할 때 사용
- ✓ 모든 요소에 적용되며 상속됨
  - 값 : normal | small-caps
  - font-variant : normal ;
  - font-variant : small-caps ;





# Typography

## ❖ font

- ✓ font-family, font-size, line-height, font-weight, font-style, font-variant 의 6가지 속성을 한번에 선언할 때 사용하는 대표 속성으로 각각의 속성을 공백으로 구분  
값 : [ font-weight | font-style | font-variant ] | [ font-size ] | [ /line-height ] | [ font-family ]
  - font : bold italic small-caps 12px /1.6 "돋움",sans-serif;
- ✓ 다른 속성은 생략이 가능하지만 font-size와 font-family는 생략이 불가능
- ✓ 키워드를 이용해서 설정하는 것도 가능
  - caption
  - icon
  - menu
  - message-box
  - small-caption
  - status-bar

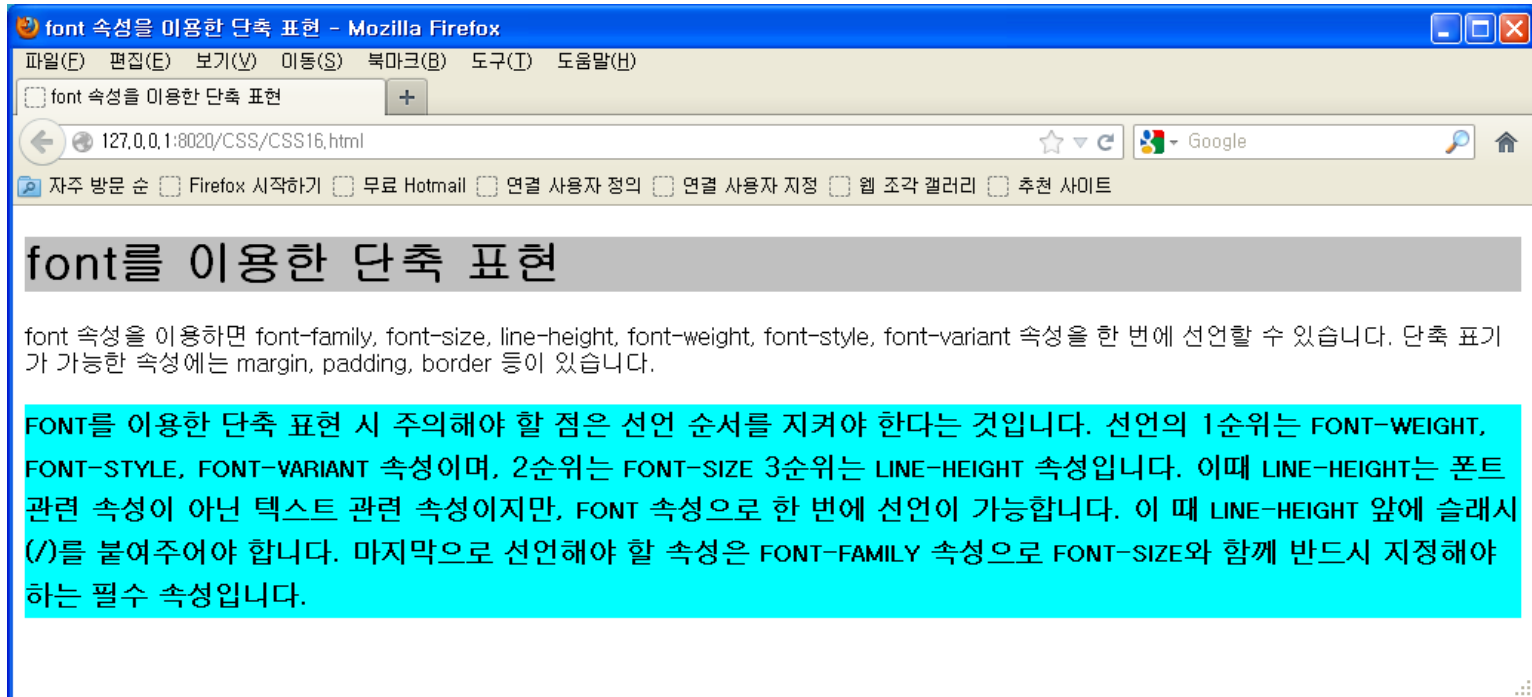
# Typography

## ❖ color

- ✓ 요소의 글자 색상을 지정할 때 사용하는데 경계선에도 영향을 줌  
값 : 색상명 | RGB 16진수 | RGB 10진수 | RGB 백분율 | hsl(색조, 채도, 명도), inherit
  - color : red ;
  - color : #ff0000 ;
  - color : rgb(255,0,0) ;
  - color : rgb(100%,0%,0%) ;
  - color : hsl(100%,0%,0%) ;
- ✓ 표준 색상: black, white, purple, lime, navy, aqua, silver, maroon, fuchsia, olive, blue, orange, gray, red, green, yellow, teal
- ✓ css3에서는 140 여 가지의 색상 사용 가능  
<https://www.learningwebdesign.com/colornames.html>
- ✓ 색상 추출 사이트  
<https://www.webfx.com/web-design/color-picker/>

# Typography

## ❖ typography2.html



# Typography

## ❖ typography2.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>font 속성을 이용한 단축 표현</title>
<style type="text/css">
 h1 {
 background: silver;
 }

 .font {
 font: bold normal small-caps 18px/1.6 "돋움", Dotum, sans-serif;
 background: aqua;
 }
</style>
</head>
```

# Typography

## ❖ typography2.html

<body>

<h1>font를 이용한 단축 표현</h1>

<p>font 속성을 이용하면 font-family, font-size, line-height, font-weight, font-style, font-variant 속성을 한 번에 선언할 수 있습니다. 단축 표기가 가능한 속성에는 margin,

padding, border 등이 있습니다.</p>

<p class="font">font를 이용한 단축 표현 시 주의해야 할 점은 선언 순서를 지켜야 한다는 것입니다.

선언의 1순위는 font-weight, font-style, font-variant 속성이며, 2순위는 font-size

3순위는 line-height 속성입니다. 이때 line-height는 폰트 관련 속성이 아닌 텍스트 관련 속성이지만,

font 속성으로 한 번에 선언이 가능합니다. 이 때 line-height 앞에 슬래시(/)를 붙여주어야 합니다. 마지막으로

선언해야 할 속성은 font-family 속성으로 font-size와 함께 반드시 지정해야 하는 필수 속성입니다.</p>

</body>

</html>

# Typography

## ❖ text-decoration

- ✓ 텍스트에 밑줄을 긋거나 취소선 등의 효과를 낼 경우에 사용
- ✓ 모든 요소에 적용되며 상속 안됨
  - 값 : none | underline | overline | line-through | blink | inherit
  - text-decoration : none;
  - text-decoration : underline;
- ✓ 링크된 텍스트의 밑줄을 없애는 대신 색이나 두께를 지정함으로써 텍스트가 링크임을 알 수 있게 하기를 권장
- ✓ 링크가 아닌 텍스트에 밑줄을 추가하는 것은 사용자를 혼동스럽게 할 수 있으므로 기울임을 사용하는 것을 고려

# Typography

## ❖ text-transform

- ✓ 영문 대소문자를 변환할 때 사용
- ✓ 모든 요소에 적용되며 상속됨

값 : none | uppercase | lowercase | capitalize | inherit

- text-transform : none;
- text-transform : uppercase;
- text-transform : lowercase;
- text-transform : capitalize;

# Typography

## ❖ white-space

✓ 공백 문자를 처리하는 방식을 결정할 때 사용

### ● 값

- ◆ normal: 여러 개의 공백을 하나로 처리
- ◆ nowrap: 여러 개의 공백을 하나로 표시하고 영역 너비를 넘어가는 내용은 줄을 바꾸지 않고 계속 한 줄로 표시
- ◆ pre: 여러 개의 공백을 그대로 표시하고 영역 너비를 넘어가는 내용은 줄을 바꾸지 않고 계속 한 줄로 표시
- ◆ pre-wrap: 여러 개의 공백을 그대로 표시하고 영역 너비를 넘어가는 내용은 줄을 바꿈
- ◆ pre-line: 여러 개의 공백을 그대로 표시하고 영역 너비를 넘어가는 내용은 줄을 바꿈
- ◆ inherit



# Typography

## ❖ paragraph

### ✓ text-align

- 단락의 텍스트를 가로 기준으로 정렬할 때 사용
- 블록 요소와 테이블의 셀, 인라인 요소에 적용 가능하며 상속됨
- 셀이나 인라인 요소는 width가 설정되서 콘텐츠보다 너비가 넓게 설정되어 있는 경우만 적용

값 : start | end | left | center | right | justify | inherit

◆ text-align : left;

◆ text-align : center;

◆ text-align : justify;

# Typography

## ❖ Paragraph

### ✓ text-justify

- 정렬을 할 때 justify를 적용한 경우의 공백 조절
- 값
  - ◆ auto: 웹 브라우저에서 자동 조정
  - ◆ none: 정렬하지 않음
  - ◆ inter-word: 단어 사이의 공백을 조절해서 정렬
  - ◆ distribute: 인접한 글자 사이의 공백을 동일하게 맞추어 정렬

# Typography

## ❖ Paragraph

### ✓ text-indent

- 문단의 첫 줄 들여쓰기를 적용할 때 사용
- 블록 요소와 테이블 셀 그리고 인라인 블록에 사용이 가능하며 상속됨
- 음수를 지정할 수 있는데 음수를 지정하면 첫 줄 내어 쓰기

값 : 길이 | 퍼센트 | inherit

◆ text-indent : 50px;

◆ text-indent : 2em;

# Typography

## ❖ Paragraph

### ✓ letter-spacing

- 문자 간격을 조절할 때 사용
- 모든 요소에 가능하며 상속됨
- 단어 사이의 간격은 word-spacing  
값 : 길이 | normal | inherit
  - ◆ letter-spacing : 1em ;
  - ◆ letter-spacing : -3px ;



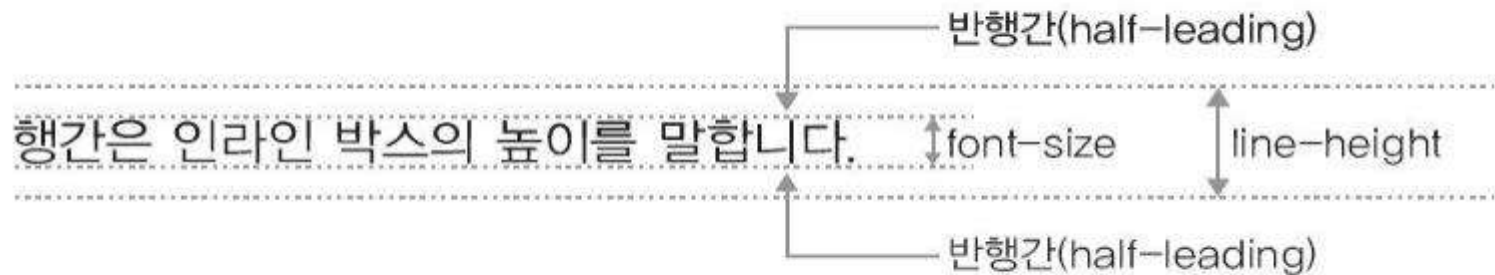
# Typography

## ❖ Paragraph

### ✓ line-height

- 문단의 행간을 지정할 때 사용
  - 모든 요소에 사용 가능하며 상속됨
- 값 : normal | 실수 | 길이 | 퍼센트 | inherit

- ✓ line-height: normal ;
- ✓ line-height: 1.5em ; -> 현재 글꼴의 1.5배
- ✓ line-height: 1.6 ; -> 현재 글꼴의 1.6배
- ✓ line-height: 200% ; -> 현재 글꼴의 2배



# Typography

## ❖ Paragraph

### ✓ word-break

- normal, break-all, keep-all(주로 사용, 한줄일 텍스트의 단어 단위 줄 바꿈 가능)

### ✓ direction

- 텍스트의 쓰기 방향 지정
- 값은 ltr, rtl(아랍어에서 사용)

### ✓ vertical-align

- 인라인 요소끼리의 세로 위치를 정렬할 때 사용

값: baseline | sub | super | top | text-top | middle | bottom | text-bottom | 길이 | 퍼센트 | inherit

# Typography

❖ paragraph.html

## 인라인 요소의 수직 위치 지정

X<sup>2</sup>와 같이 윗첨자가 적용된 모습

H<sub>2</sub>O와 같이 아랫첨자가 적용된 모습



이미지 요소에 vertical-align:top이 적용된 모습



이미지 요소에 vertical-align:middle이 적용된 모습



이미지 요소에 vertical-align:bottom이 적용된 모습

# Typography

## ❖ paragraph.html

```
<!DOCTYPE html>
<html>
<head>
 <meta charset="UTF-8">
 <title>vertical-align 속성을 이용한 수직 위치 지정</title>
 <style type="text/css">
 h1 { background:silver; }
 .super {
 vertical-align:super;
 font-size:0.7em;
 background:yellow;
 }
 .sub {
 vertical-align:sub;
 font-size:0.7em;
 background:aqua;
 }
 .top { vertical-align:top; }
 .middle { vertical-align:middle; }
 .bottom { vertical-align:bottom; }
 </style>
</head>
```



# Typography

## ❖ paragraph.html

```
<body>
```

```
 <h1>인라인 요소의 수직 위치 지정</h1>
```

```
 <p>X2와 같이 윗첨자가 적용된 모습</p>
```

```
 <p>H2O와 같이 아랫첨자가 적용된 모습</p>
```

```
 <p>이미지 요소에 vertical-align:top이 적용된 모습</p>
```

```
 <p>이미지 요소에 vertical-align:middle이 적용된 모습</p>
```

```
 <p>이미지 요소에 vertical-align:bottom이 적용된 모습</p>
```

```
</body>
```

```
</html>
```

# Typography

## ❖ 텍스트 그림자

- ✓ CSS3에서는 text-shadow 속성이 추가되서 깊이 감과 질감을 추가하는 것이 가능
- ✓ 텍스트의 그림자는 텍스트의 뒤에 그려지지만 배경이나 경계선이 있다면 배경이나 경계선의 앞에 배치됨
- ✓ none을 지정할 수 있으며 상하 거리, 좌우 거리, 색상 값을 순서대로 적용할 수 있으며 ,로 구분해서 2중으로 적용 가능

- ✓ IE에서는 10 버전 이상에서 적용

값 : 수평 오프셋, 수직 오프셋, 흐림 반경(기본 값은 0), 색상(기본 값은 현재 글자 색)

```
h1{
```

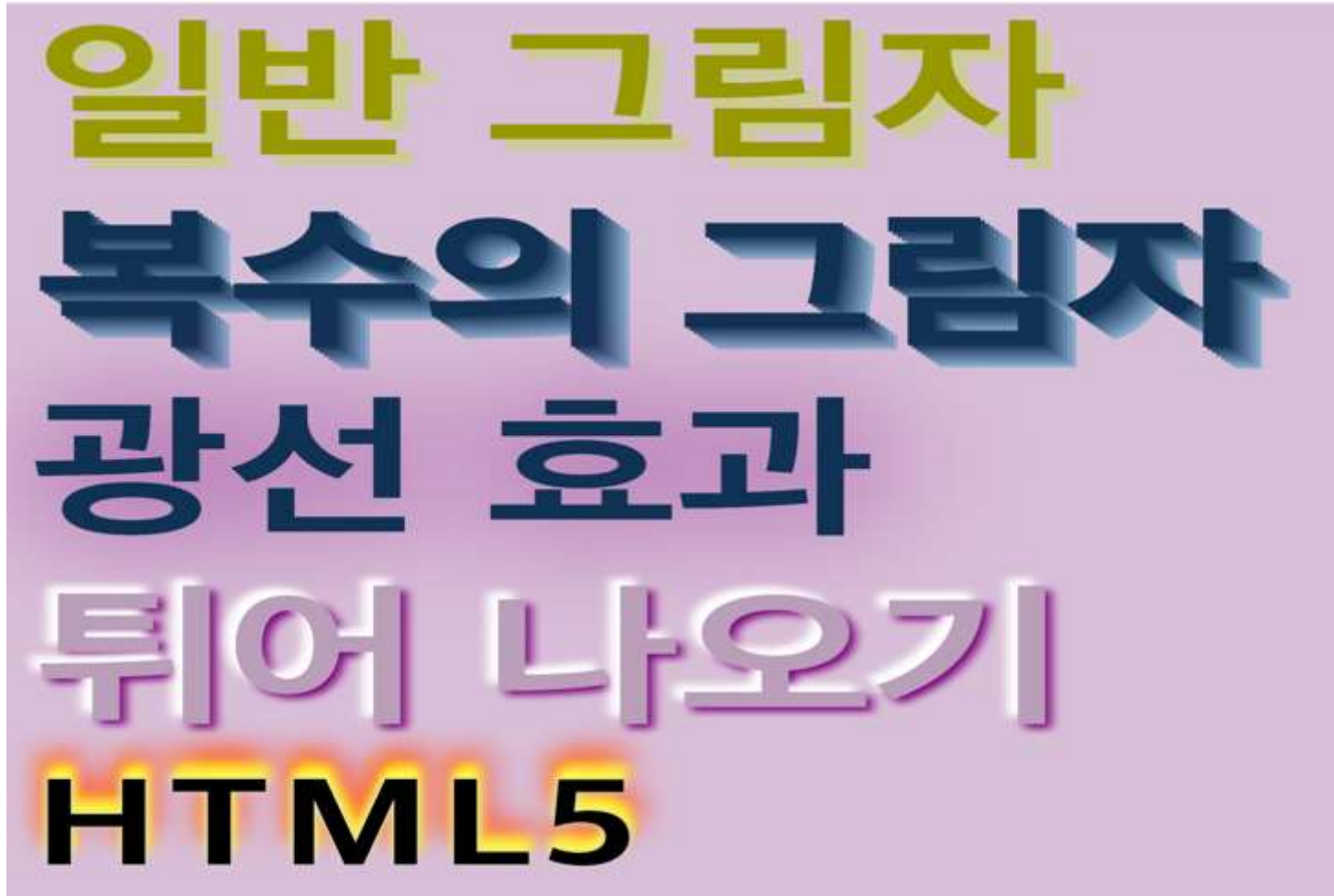
```
 text-shadow: .2em .2em silver
```

```
}
```

- ✓ 수평 오프셋과 수직 오프셋이 동일하다면 하나만 설정 가능
- ✓ 모든 요소에 그림자를 넣으면 읽기 어려워지고 처리 능력이 떨어지는 모바일 브라우저는 문제가 발생할 가능성이 있음

# Typography

## ❖ 텍스트 그림자



# Typography

❖ [textshadow.html](#)

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8">
 <title>text-shadow</title>
 <style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }
 li {
 list-style-type: none;
 }
 body {
 font: 12px "Arial", Gulim;
 margin-left: 20px;
 background-color: thistle;
 }
 h1 {
 font-size: 100px;
 color: #990;
 text-shadow: 7px 7px #cc9;
 }

```

# Typography

❖ textshadow.html

```
h2 {
 font-size: 100px;
 color: #135;
 text-shadow: 2px 2px #246, 4px 4px #357, 6px 6px #468, 8px 8px
#579, 10px 10px #68a, 12px 12px #79b, 14px 14px #8ac;
}
h3 {
 font-size: 100px;
 color: #135;
 text-shadow: 0 0 0.7em purple;
}
h4 {
 font-size: 100px;
 color: #ba9eba;
 text-shadow: -.05em -.05em .05em white, .03em .03em .05em
purple;
}
</style>
</head>
```

# Typography

## ❖ textshadow.html

```
<body>
 <h1>일반 그림자</h1>
 <h2>복수의 그림자</h2>
 <h3>광선 효과</h3>
 <h4>튀어 나오기</h4>
</body>
</html>
```



# 목록

## ❖ list

### ✓ list-style-type:

- 목록을 생성할 때 기본적으로 제공되는 마커를 변경하고자 할 때 사용
- 값: none | disc | circle | square | decimal | decimal-leading-zero | upper-alpha | lower-alpha | upper-roman | lower-roman | upper-latin | lower-latin | lower-greek | armenian | georgian | inherit

### ✓ list-style-image

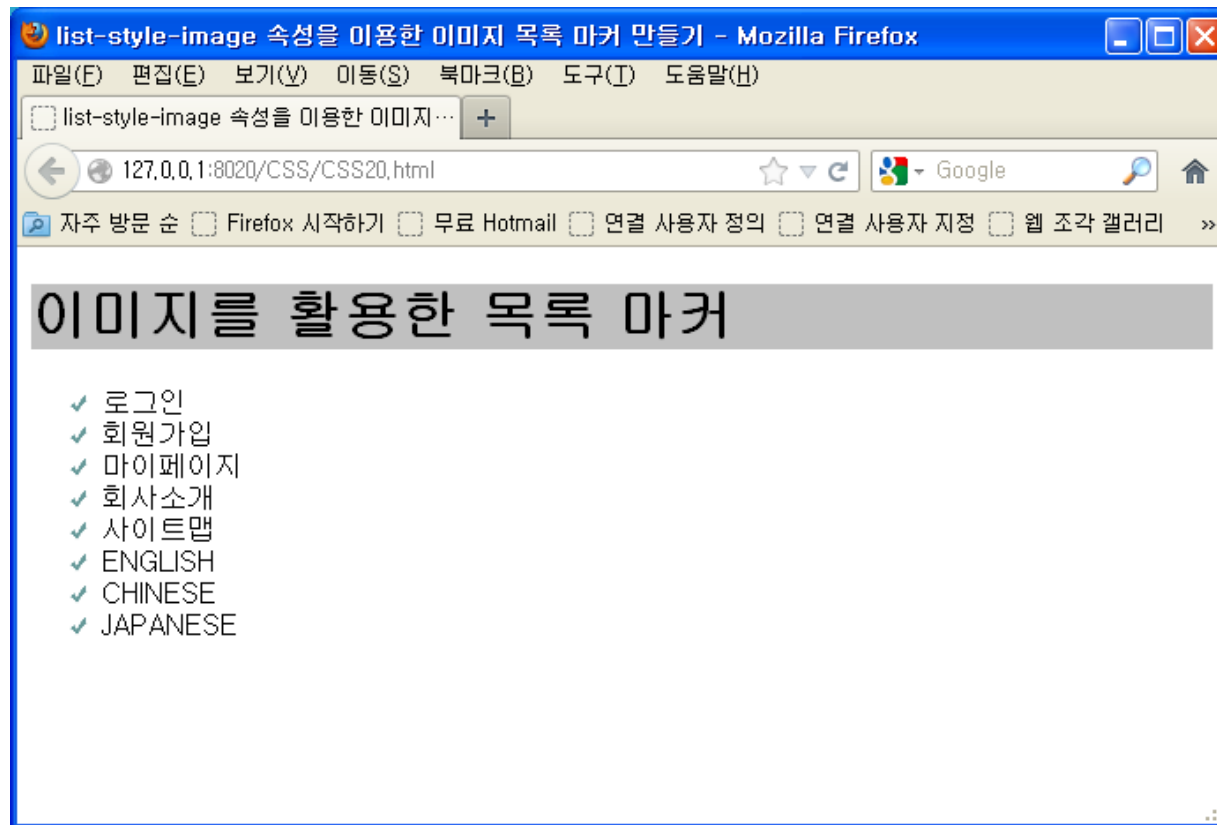
- 목록 마커 대신 이미지로 변경할 때 사용
- 값 : url() | none | inherit

list-style-image : url(images/icon.gif);

- ✓ list-style-position: 목록에 들여쓰기 효과를 설정할 때 사용하는 것으로 inside와 outside 설정 가능
- ✓ list-style 속성으로 3가지를 동시에 설정하는 것이 가능한데 순서는 type, position, image 순

# 목록

❖ list





# 목록

## ❖ list.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>list-style-image 속성을 이용한 이미지 목록 마커 만들기</title>
<style type="text/css">
.marker {
 list-style-image: url(images/fire.png);
}
</style>
</head>
```

# 목록

## ❖ list.html

```
<body>
 <h1> 이미지를 활용한 목록 마커 </h1>
 <ul class="marker">
 로그인
 회원가입
 마이페이지
 회사소개
 사이트맵
 ENGLISH
 CHINESE
 JAPANESE

</body>
</html>
```

# Background

## ❖ background-color

- ✓ 요소의 배경 색상을 지정할 때 사용
- ✓ 모든 요소에 사용이 가능하지만 상속이 되지 않음

값 : 색상명 | RGB 16진수 | RGB 10진수 | RGB 백분율 | inherit

- background-color : red ;
- background-color : #ff0000 ;
- background-color : rgb(255,0,0) ;
- background-color : rgb(100%,0%,0%) ;

# Background

## ❖ background-image

- ✓ 요소의 배경 이미지를 지정할 때 사용
- ✓ 최근에는 img 태그 대신에 배경을 이용해서 이미지를 출력하는 경우가 많음
- ✓ 경로는 현재 URL의 상대적 위치로 설정가능하고 URL로 설정하는 것도 가능
- ✓ 여러 개 설정이 가능한데 여러 개를 설정하면 순서대로 적용

값 : none | uri() | inherit

- background-image: none ;
- background-image: url(images/bg.gif) ;

# Background

## ❖ background-repeat

✓ 요소에 지정한 배경 이미지의 반복 여부를 변경하고자 할 때 사용

값 : repeat | repeat-x | repeat-y | no-repeat | inherit

- background-image: url(images/bg.gif) ;
- background-repeat:repeat-y;

# Background

## ❖ background

### **Black Goose Bistro • Summer Menu**

*Baker's Corner, Seekonk, Massachusetts*

**HOURS: MONDAY THROUGH THURSDAY: 11 to 9, FRIDAY AND SATURDAY: 11 to midnight**

*Appetizers*

*Main Courses*

*Traditional Toasts*

*Dessert Selection*

#### **APPETIZERS**

*This season, we explore the spicy flavors of the southwest in our appetizer collection.*

#### **Black bean purses**

Spicy black bean and a blend of mexican cheeses wrapped in sheets of phyllo and baked until golden. \$3.95

#### **Southwestern napoleons with lump crab — new item!**

Layers of light lump crab meat, bean and corn salsa, and our handmade flour tortillas. \$7.95

# Background

## ❖ background.html

```
<!DOCTYPE html >
<html>
 <!-- BACKGROUND-IMAGE -->
<head>
 <meta charset="utf-8">
 <title>Black Goose Bistro Summer Menu</title>
 <link href='http://fonts.googleapis.com/css?family=Marko+One' rel='stylesheet'
 type='text/css'>
 <style>
 body {
 font-family: Georgia, serif;
 font-size: 100%;
 line-height: 175%;
 margin: 0 15%;
 background-color: #d2dc9d;
 background-image: url(images/bullseye.png);
 }
 #header {
 margin-top: 0;
 padding: 3em 1em 2em 1em;
 text-align: center;
 background-color: rgba(255,255,255,.5);
 }
 </style>
</head>
<body>
```

# Background

## ❖ background.html

```
a {
 text-decoration: none;
}
a:link {
 color: #939;
}
a:visited {
 color: #937393;
}
a:focus {
 background-color: #fff;
 color: #c700f2;
}
a:hover {
 background-color: #fff;
 color: #c700f2;
}
a:active {
 background-color: #fff;
 color: #f0f;
}
```



# Background

## ❖ background.html

```
h1 {
 font: bold 1.5em Georgia, serif;
 text-shadow: .1em .1em .2em gray;
 color: #939;
}
h2 {
 font-size: 1em;
 text-transform: uppercase;
 letter-spacing: .5em;
 text-align: center;
 color: #c60;
}
dt {
 font-weight: bold;
}
strong {
 font-style: italic;
}
ul {
 list-style-type: none;
 margin: 0;
 padding: 0;
}
```

# Background

## ❖ background.html

```
#info p {
 font-style: italic;
}
.price {
 font-family: Georgia, serif;
 font-style: italic;
}
p.warning, sup {
 font-size: small;
}
.label {
 font-weight: bold;
 font-variant: small-caps;
 font-style: normal;
}

h2 + p {
 text-align: center;
 font-style: italic;
}
</style>
</head>
```

# Background

## ❖ background.html

```
<body>
```

```
<div id="header">
```

```
<h1>Black Goose Bistro • Summer Menu</h1>
```

```
<div id="info">
```

```
<p>Baker's Corner, Seekonk, Massachusetts

```

```
Hours: Monday through Thursday: 11 to 9, <span
class="label">Friday and Saturday; 11 to midnight</p>
```

```

```

```
Appetizers
```

```
Main Courses
```

```
Traditional Toasts
```

```
Dessert Selection
```

```

```

```
</div>
```

```
</div>
```

# Background

## ❖ background.html

```
<div id="appetizers">
<h2>Appetizers</h2>
<p>This season, we explore the spicy flavors of the southwest in our appetizer
 collection.</p>

<dl>
<dt>Black bean purses</dt>
<dd>Spicy black bean and a blend of mexican cheeses wrapped in sheets of phyllo and
 baked until golden. $3.95</dd>

<dt class="newitem">Southwestern napoleons with lump crab — new
 item!</dt>
<dd>Layers of light lump crab meat, bean and corn salsa, and our handmade flour tortillas.
 $7.95</dd>
</dl>
</div>
```

# Background

## ❖ background.html

```
<div id="entrees">
<h2>Main courses</h2>
<p>Big, bold flavors are the name of the game this summer. Allow us to assist you with
 finding the perfect wine.</p>

<dl>
<dt class="newitem">Jerk rotisserie chicken with fried plantains — new
 item!</dt>
<dd>Tender chicken slow-roasted on the rotisserie, flavored with spicy and fragrant jerk
 sauce and served with fried plantains and fresh mango. Very spicy.
 $12.95</dd>

<dt>Shrimp sate kebabs with peanut sauce</dt>
<dd>Skewers of shrimp marinated in lemongrass, garlic, and fish sauce then grilled to
 perfection. Served with spicy peanut sauce and jasmine rice. $12.95</dd>

<dt>Grilled skirt steak with mushroom fricasee</dt>
<dd>Flavorful skirt steak marinated in asian flavors grilled as you like it[*].
 Served over a blend of sauteed wild mushrooms with a side of blue cheese mashed
 potatoes. $16.95</dd>
</dl>
</div>
```

# Background

## ❖ background.html

```
<div id="toast">
<h2>Traditional Toasts</h2>
<p>The ultimate comfort food, our traditional toast recipes are adapted from <cite>The
 Whitehouse Cookbook</cite> published in 1887.</p>
<dl>
<dt>Cream toast</dt>
<dd>Simple cream sauce over highest quality toasted bread, baked daily. $3.95 </dd>

<dt>Mushroom toast</dt>
<dd>Layers of light lump crab meat, bean and corn salsa, and our handmade flour tortillas.
 $6.95 </dd>

<dt>Nun's toast</dt>
<dd>Onions and hard-boiled eggs in a cream sauce over buttered hot toast. $6.95 </dd>

<dt>Apple toast</dt>
<dd>Sweet, cinnamon stewed apples over delicious buttery grilled bread. $6.95 </dd>
</dl>
</div>
```

# Background

## ❖ background.html

```
<div id="dessert">
<h2>Dessert Selection</h2>
<p>Be sure to save room for our desserts, made daily by our own Johnson & Wales trained pastry
 chef.</p>

<dl>
<dt class="newitem">Lemon chiffon cake — new item!</dt>
<dd>Light and citrus flavored sponge cake with buttercream frosting as light as a cloud.
 $2.95</dd>

<dt class="newitem">Molten chocolate cake</dt>
<dd>Bubba's special dark chocolate cake with a warm, molten center. Served with or
 without a splash of almond liqueur. $3.95</dd>
</dl>
</div>

<p class="warning">[*] We are required to warn you that undercooked food is
 a health risk.</p>

</body>
</html>
```

# Background

## ❖ background-position

- ✓ 요소에 지정한 배경 이미지의 위치를 변경하고자 할 때 사용
- ✓ 모든 요소에 적용가능하지만 상속은 안됨

값 : left | right | center | top | bottom | 길이 | 퍼센트 | inherit

- background-image: url(images/bg.gif) ;
- background-position:left top;



# Background

## ❖ background-attachment

- ✓ 요소에 지정된 배경 이미지를 스크롤되어도 항상 같은 위치에 고정시키고자 할 때 사용

값 : scroll | fixed | inherit

- background-image: url(images/bg.gif) ;
- background-attachment:fixed;
  - ◆ fixed를 설정하면 배경 이미지 고정
  - ◆ scroll을 설정하면 스크롤하면 배경 이미지도 같이 스크롤

# Background

## ❖ background-size

- ✓ 배경 이미지의 크기를 조절하는 속성으로 CSS3에서 추가된 속성
- ✓ 값
  - auto: 원본 이미지 크기만큼 표시
  - 정수2개나 백분율 2개의 값: 너비와 높이를 적용
  - 값을 1개만 입력하면 너비로 인식하고 높이는 auto 적용
  - cover: 너비와 높이 비율을 맞추어서 확대하거나 축소하는데 큰 값을 적용
  - contain: 작은 값을 적용

# Background

## ❖ background-clip

- ✓ 배경 이미지나 배경색을 어디까지 적용할 것인지를 설정하는 속성
- ✓ 값
  - border-box: 테두리 까지 적용
  - padding-box: padding 범위까지 적용
  - content-box: 내용부분에만 적용
- ✓ background-origin은 background-clip 과 동일한 속성을 사용하는데 이미지의 출력 기준 위치를 설정

# Background

## ❖ background

- ✓ font 속성과 같이 배경과 관련된 다섯 가지 속성을 단축해서 한 번에 선언하고자 할 때 사용
    - 값: background-color | background-image | background-repeat | background-position | background-attachment | inherit
- background: yellow url(images/bg.gif) no-repeat left top fixed ;



# Background

## ❖ background

### ㉡ 아침 식사와 저녁 식사

하루일과를 잘 시작하여 성공적인 업무를 마치려면 아침 식사는 거르지 않는 것이 좋을 것이다. 낮에 취침하고 밤에 일하는 생활 유형을 가진 사람이라면 그 사람은 저녁 식사를 거르지 않도록 하는 것이 더욱 중요할 것 같다.

### ㉡ 간식과 야참

가공식품으로 구성된 간식이면 먹지 않는 것보다 더 좋지 않다. 몸 안의 좋은 성분과 나쁜 성분의 구분할 수 없게 된다. 건강을 생각하여 과일 등의 자연식품으로 차린 야참은 과식만 하지 않는다면 먹어도 좋다.

### ㉡ 보양식과 약

보양식은 일반식사나 간식을 면밀히 검토해야 할 경우가 많하지만 약을 복용중이라면 평소 식사나 간식에 주의해야 한다. 음식도 궁합이 있는데 약의 작용은 더욱 심하기 때문이다.

# Background

## ❖ backgroundimage.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>background</title>
<style type="text/css">
 body {
 font: 12px "굴림", Gulim;
 }

 h1 {
 background: url("images/bg_bul.gif") no-repeat 0 50%;
 padding-left: 20px;
 }

 p {
 width: 400px;
 padding: 25px;
 background: #fef url("images/bg_line.gif") repeat-x bottom;
 }
/* p태그의 여백을 포함한 공간내에 배경이미지를 바닥에 가로반복하여 깔아줌. */
</style>
</head>
```

# Background

## ❖ backgroundimage.html

<body>

<h1>아침 식사와 저녁 식사</h1>

<p>하루일과를 잘 시작하여 성공적인 업무를 마치려면 아침 식사는 거르지 않는 것이 좋을 것이다. 낮에 취침하고 밤에

일하는 생활 유형을 가진 사람이라면 그 사람은 저녁 식사를 거르지 않도록 하는 것이 더욱 중요할 것 같다.</p>

<h1>간식과 야참</h1>

<p>가공식품으로 구성된 간식이면 먹지 않는 것보다 더 좋지 않다. 몸 안의 좋은 성분과 나쁜 성분의 구분할 수 없게

된다. 건강을 생각하여 과일 등의 자연식품으로 차린 야참은 과식만 하지 않는다면 먹어도 좋다.</p>

<h1>보양식과 약</h1>

<p>보양식은 일반식사나 간식을 면밀히 검토해야 할 경우가 덜하지만 약을 복용중이라면 평소 식사나 간식에 주의해야 한다.

음식도 궁합이 있는데 약의 작용은 더욱 심하기 때문이다.</p>

</body>

</html>

# Background

## ❖ 여러 개의 배경 이미지

- ✓ background-image 속성에 여러 개의 값을 쉼표로 구분해서 설정

```
body{
 //여러 개의 이미지를 적용할 수 없을 때 설정
 background: url(images/purpledotted.png) center top repeat-x;
 //여러 개의 이미지를 적용
 background:
 url(images/purpledotted.png) left top repeat-y,
 url(images/purpledotted.png) right top repeat-y,
 url(images/gooseshadow.png) 90% bottom no-repeat;
 //이미지가 없을 때 적용
 background-color: rgba(255,255,255,.5);
}
```



# Background

## ❖ 여러 개의 배경 이미지



# Background

## ❖ 여러 개의 배경 이미지

### ✓ backgroundmultiimage.html

```
<!DOCTYPE html >
<html>
<head>
<meta charset="utf-8">
<title>Black Goose Bistro Summer Menu</title>
<link href='http://fonts.googleapis.com/css?family=Marko+One' rel='stylesheet'
 type='text/css'>
<link rel="stylesheet" href="menustyles.css">
</head>
```

# Background

## ❖ 여러 개의 배경 이미지

- ✓ backgroundmultiimage.html

```
<body>
```

```
 <div id="header">
```

```
 <h1>Black Goose Bistro • Summer Menu</h1>
```

```
 <div id="info">
```

```
 <p>
```

```
 class="label">Hours:
```

```
 9, Friday
```

```
 </p>
```

```

```

```
 Appetizers
```

```
 Main Courses
```

```
 Traditional Toasts
```

```
 Dessert Selection
```

```

```

```
 </div>
```

```
</div>
```

Baker's Corner, Seekonk, Massachusetts<br> <span

Monday through Thursday:</span> 11 to

and Saturday;</span> 11 to midnight

# Background

## ❖ 여러 개의 배경 이미지

### ✓ backgroundmultiimage.html

```
<div id="appetizers">
 <h2>Appetizers</h2>
 <p>This season, we explore the spicy flavors of the southwest in
 our appetizer collection.</p>

 <dl>
 <dt>Black bean purses</dt>
 <dd>
 Spicy black bean and a blend of mexican cheeses
 wrapped in sheets of phyllo and baked until golden. $3.95
 </dd>

 <dt class="newitem">
 Southwestern napoleons with lump crab —
 new
 item!
 </dt>
```

# Background

## ❖ 여러 개의 배경 이미지

- ✓ backgroundmultiimage.html

```

 <dd>
 Layers of light lump crab meat, bean and corn salsa,
 and our
 handmade flour tortillas. $7.95
 </dd>
 </dl>
 </div>
```

# Background

## ❖ 여러 개의 배경 이미지

- ✓ backgroundmultiimage.html

```
<div id="entrees">
```

```
 <h2>Main courses</h2>
```

```
 <p>Big, bold flavors are the name of the game this summer. Allow
 us to assist you with finding the perfect wine.</p>
```

```
 <dl>
```

```
 <dt class="newitem">
```

```
 Jerk rotisserie chicken with fried plantains —
```

```
 new
```

```
 item!
```

```
 </dt>
```

```
 <dd>
```

```
 with spicy
```

```
 plantains and fresh
```

```
 Tender chicken slow-roasted on the rotisserie, flavored
 and fragrant jerk sauce and served with fried
 mango. Very spicy. <span
```

```
 class="price">$12.95
```

```
 </dd>
```

# Background

## ❖ 여러 개의 배경 이미지

✓ backgroundmultiimage.html

```
<dt>Shrimp sate kebabs with peanut sauce</dt>
<dd>
 Skewers of shrimp marinated in lemongrass, garlic,
 and fish sauce
 then grilled to perfection. Served with spicy peanut
 sauce and
 jasmine rice. $12.95
</dd>

<dt>Grilled skirt steak with mushroom fricasee</dt>
<dd>
 Flavorful skirt steak marinated in asian flavors grilled
 as you like
 it[*]. Served over a blend of sauteed wild
 mushrooms with a
 side of blue cheese mashed potatoes. $16.95
</dd>
</dl>
</div>
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ backgroundmultiimage.html

```
<div id="toast">
 <h2>Traditional Toasts</h2>
 <p>
 The ultimate comfort food, our traditional toast recipes are
 adapted
 from <a
 href="http://www.gutenberg.org/files/13923/13923-
 h/13923-h.htm"> <cite>The
 Whitehouse Cookbook</cite>
 published in 1887.
 </p>
 <dl>
 <dt>Cream toast</dt>
 <dd>
 Simple cream sauce over highest quality toasted
 bread, baked daily.
 $3.95
 </dd>
```



# Background

## ❖ 여러 개의 배경 이미지

- ✓ backgroundmultiimage.html

```
<dt>Mushroom toast</dt>
<dd>
 Layers of light lump crab meat, bean and corn salsa,
and our
 handmade flour tortillas. $6.95
</dd>
<dt>Nun's toast</dt>
<dd>
 Onions and hard-boiled eggs in a cream sauce over
buttered hot
 toast. $6.95
</dd>
<dt>Apple toast</dt>
<dd>
 Sweet, cinnamon stewed apples over delicious buttery
grilled bread.
 $6.95
</dd>
</dl>
</div>
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ backgroundmultiimage.html

```
<div id="dessert">
 <h2>Dessert Selection</h2>
 <p>
 Be sure to save room for our desserts, made daily by our own Johnson & Wales
 trained pastry chef.
 </p>
 <dl>
 <dt class="newitem">
 Lemon chiffon cake — new
item!
 </dt>
 <dd>
 Light and citrus flavored sponge cake with
 buttercream frosting as
 light as a cloud. $2.95
 </dd>
```

# Background

## ❖ 여러 개의 배경 이미지

- ✓ backgroundmultiimage.html

```
<dt class="newitem">Molten chocolate cake</dt>
<dd>
 Bubba's special dark chocolate cake with a warm,
 molten center.
 Served with or without a splash of almond liqueur.
 $3.95
 </dd>
</dl>
</div>

<p class="warning">
 [*] We are required to warn you that undercooked food is a
 health risk.
</p>

</body>
</html>
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ menustyles.css

```
body {
 font-family: Georgia, serif;
 font-size: 100%;
 line-height: 175%;
 margin: 0 15%;
 background: #d2dc9d url(images/blackgoose.png) no-repeat center 100px fixed;

}
#header {
 margin-top: 0;
 padding: 3em 1em 2em 1em;
 text-align: center;
 background-image: url(images/purpledotted.png) center top repeat-x;
 background:
 url(images/purpledotted.png) left top repeat-y,
 url(images/purpledotted.png) right top repeat-y,
 url(images/gooseshadow.png) 90% bottom no-repeat;
 background-color: rgba(255,255,255,.5);

}
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ menustyles.css

```
a {
 text-decoration: none;
}
a:link {
 color: #939;
}
a:visited {
 color: #937393;
}
a:focus {
 background-color: #fff;
 color: #c700f2;
}
a:hover {
 background-color: #fff;
 color: #c700f2;
}
a:active {
 background-color: #fff;
 color: #f0f;
}
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ menustyles.css

```
h1 {
 font: bold 1.5em Georgia, serif;
 text-shadow: .1em .1em .2em gray;
 color: #939;
}
h2 {
 font-size: 1em;
 text-transform: uppercase;
 letter-spacing: .5em;
 text-align: center;
 color: #c60;
}
dt {
 font-weight: bold;
}
strong {
 font-style: italic;
}
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ menustyles.css

```
ul {
 list-style-type: none;
 margin: 0;
 padding: 0;
}
#info p {
 font-style: italic;
}
.price {
 font-family: Georgia, serif;
 font-style: italic;
}
p.warning, sup {
 font-size: small;
}
.label {
 font-weight: bold;
 font-variant: small-caps;
 font-style: normal;
}
```

# Background

## ❖ 여러 개의 배경 이미지

### ✓ menustyles.css

```
h2 + p {
 text-align: center;
 font-style: italic;
}
```





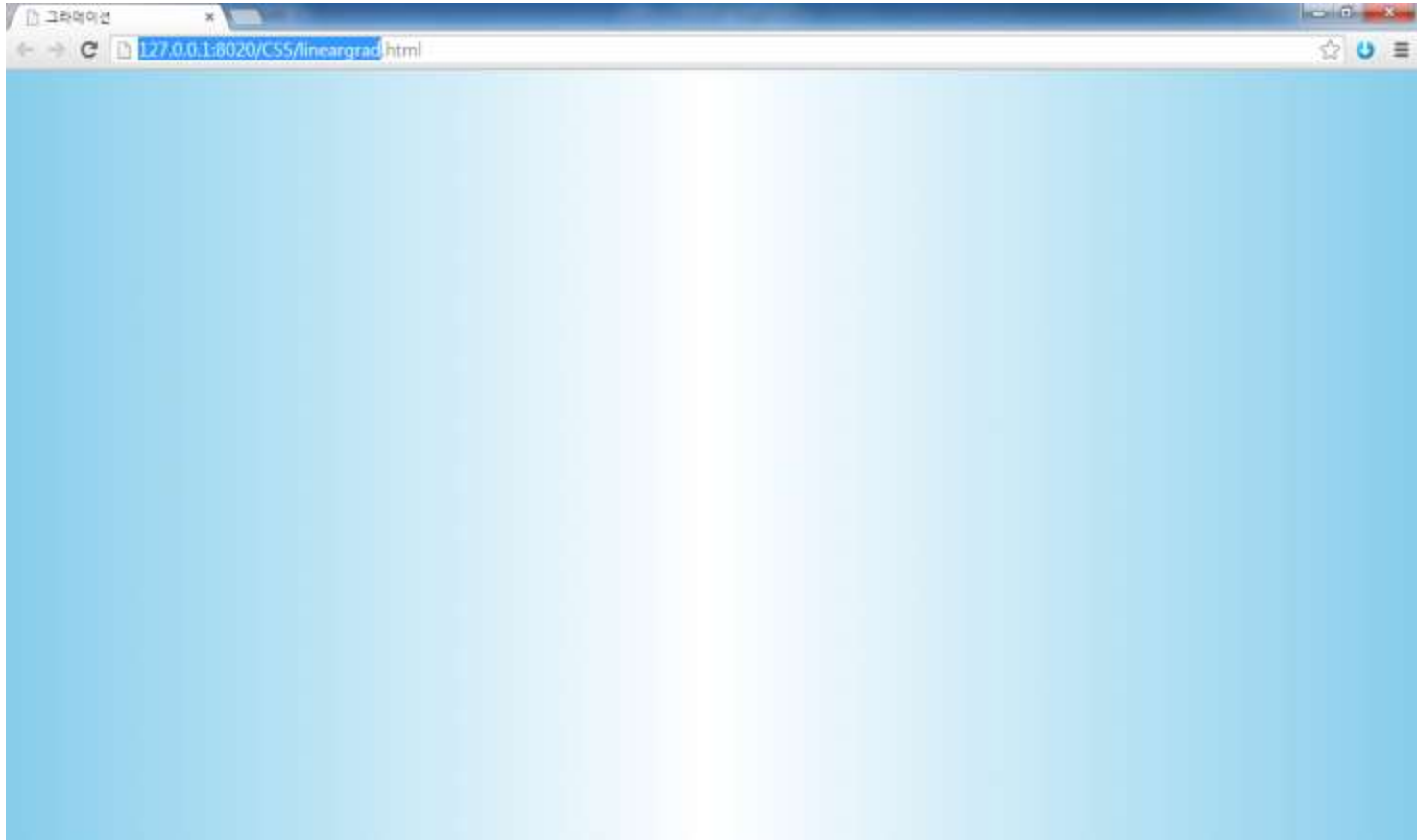
# Background

## ❖ gradation

- ✓ 하나의 색에서 다른 색으로 변화하는 것
- ✓ CSS3에서는 배경에 그라데이션을 설정하는 것이 가능
- ✓ 선형 그라데이션
  - linear-gradient(각도 또는 위치, color-stop, [color-stop]...)으로 설정
  - 각도는 시작 위치를 각도로 설정해도 되고 totop, toleft, toright, tobottom 의 방향으로 설정
  - color-stop에는 색상 값을 설정
- ✓ 그라데이션은 CSS3에서는 표준이지만 브라우저 개발사들은 완전히 합의되기 전에 자신만의 그라데이션을 만들어서 사용했는데 이 기능을 사용할 때는 브라우저 종류를 명시 (vendor-prefix)해야 함
- ✓ vendor-prefix: 브라우저 종류 명시 할 때 붙이는 접두어
  - 사파리나 크롬은 -webkit-
  - 파이어 폭스는 -moz-
  - IE는 -ms-
  - Opera는 -o-

# Background

❖ gradation



# Background

## ❖ lineargradation.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>그라데이션</title>
<style>
body{
 background : -webkit-linear-gradient(150deg, skyblue, white, skyblue);
 background : -moz-linear-gradient(150deg, skyblue, white, skyblue);
 background : -ms-linear-gradient(150deg, skyblue, white, skyblue);
}
</style>
</head>

<body>
</body>
</html>
```

# Background

## ❖ gradation

### ✓ 원형 그라데이션

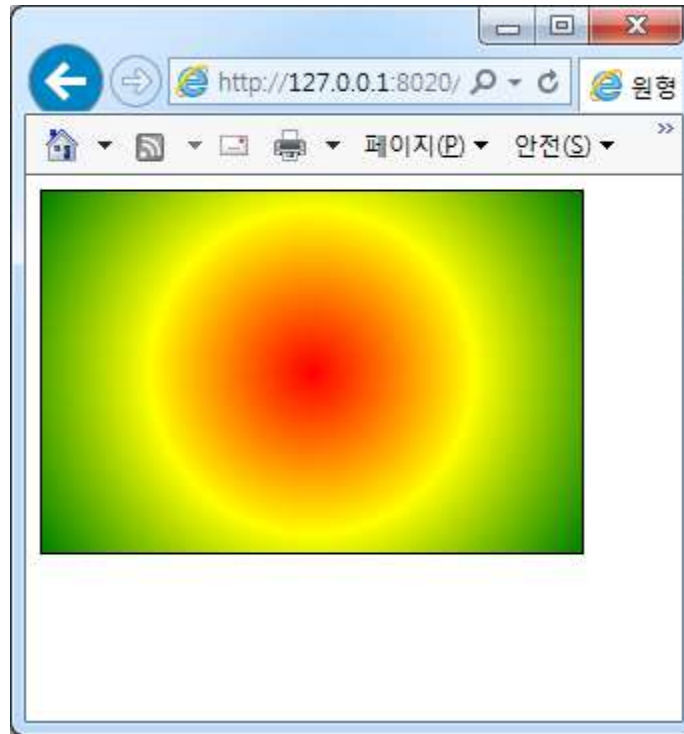
- 한 점에서 원형으로 퍼져나가는 그라데이션
- radial-gradient(시작점의 위치, circle, color-stop, [color-stop]...)
- 색상은 색상과 함께 시작 위치 설정 가능
- circle을 입력할 때는 closest-side, closest-corner, farthest-side, farthest-corner 등의 값을 함께 사용하는 것이 가능

### ✓ 그라데이션을 직접 만드는 것은 어려운 작업이라서 아래 사이트에서 작업 한 후 코드를 복사해서 사용하는 것이 편리

<http://www.colorzilla.com/gradient-editor/>

# Background

❖ gradation



# Background

## ❖ radialgradation

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 원형 그라데이션 </title>
<style>
 div{
 position:relative;
 width:300px;
 height:200px;
 border:1px solid black;
 margin-bottom:10px;
 background:-webkit-radial-gradient(circle, red, yellow, green);
 background:-moz-radial-gradient(circle, red, yellow, green);
 background:-ms-radial-gradient(circle, red, yellow, green);
 }
</style>
</head>
<body>
<div> </div>
</body>
</html>
```

# Box Model

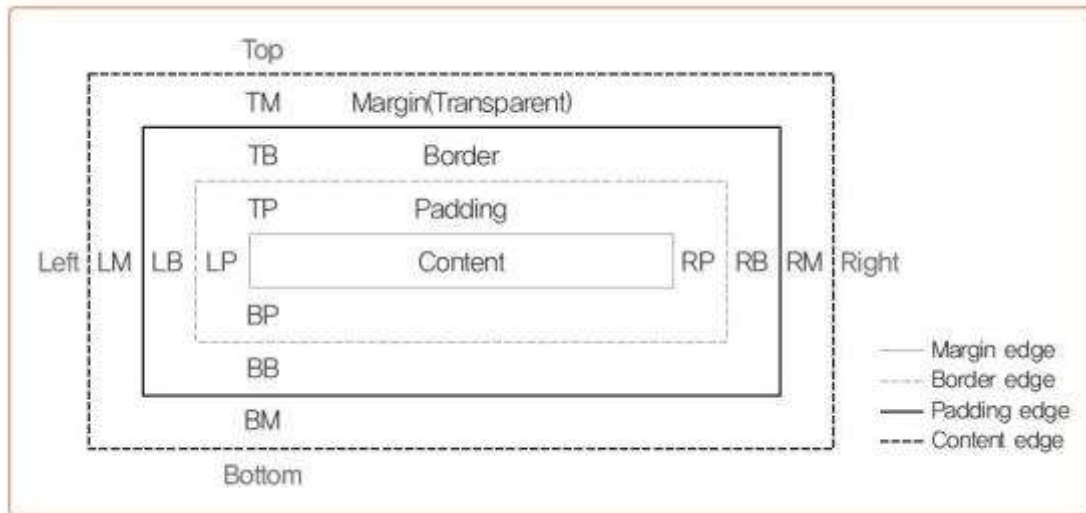
## ❖ box model

- ✓ 박스 모델은 블록 박스와 인라인 박스가 화면에 표시되는 영역과 다른 박스와의 배치 등에 대한 내용
- ✓ 화면에 표시되는 박스의 크기를 결정할 때 width, height, border, margin, padding 등의 속성을 사용할 수 있음
  - 콘텐츠 영역: 내용이 배치되는 영역
  - border: 콘텐츠 영역의 경계선
  - padding: 콘텐츠 영역과 경계선(border) 사이의 여백
  - margin: 경계선과 다른 콘텐츠의 경계선과의 여백
- ✓ 박스의 크기는 다음의 계산식을 통해 결정
  - 실제 화면에 차지하는 가로 영역 크기 계산식  
 $\text{width} + \text{margin} + \text{border} + \text{padding}$
  - 실제 화면에 차지하는 세로 영역 크기 계산식  
 $\text{height} + \text{margin} + \text{border} + \text{padding}$

# Box Model

## ❖ Box Model

- ✓ width: 콘텐츠의 가로 크기, 기본값은 auto
- ✓ height: 콘텐츠의 세로 크기, 기본값은 auto
- ✓ border: 콘텐츠 테두리
- ✓ margin: 콘텐츠 바깥쪽 여백(border 기준)으로 하나의 값을 설정하면 상하좌우에 동일하게 적용되며 margin-top, right, bottom, left의 형태로 각각 적용이 가능
- ✓ padding: 콘텐츠 안쪽 여백(border 기준)으로 사용 방법은 margin 과 동일





# Box Model

## ❖ Box Model



# Box Model

## ❖ boxmodel.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<meta charset="UTF-8" />
<title>Box Model</title>
<style type="text/css">
 body {
 font: 12px "굴림", Gulim;
 background: url("images/bg_grid.gif");
 }
 p {
 width: 500px;
 background: #abc;
 padding: 10px;
 margin: 20px 0;
 }
 .cloud {
 width: 200px;
 margin: 0 auto;
 }
</style>
</head>
```

# Box Model

## ❖ boxmodel.html

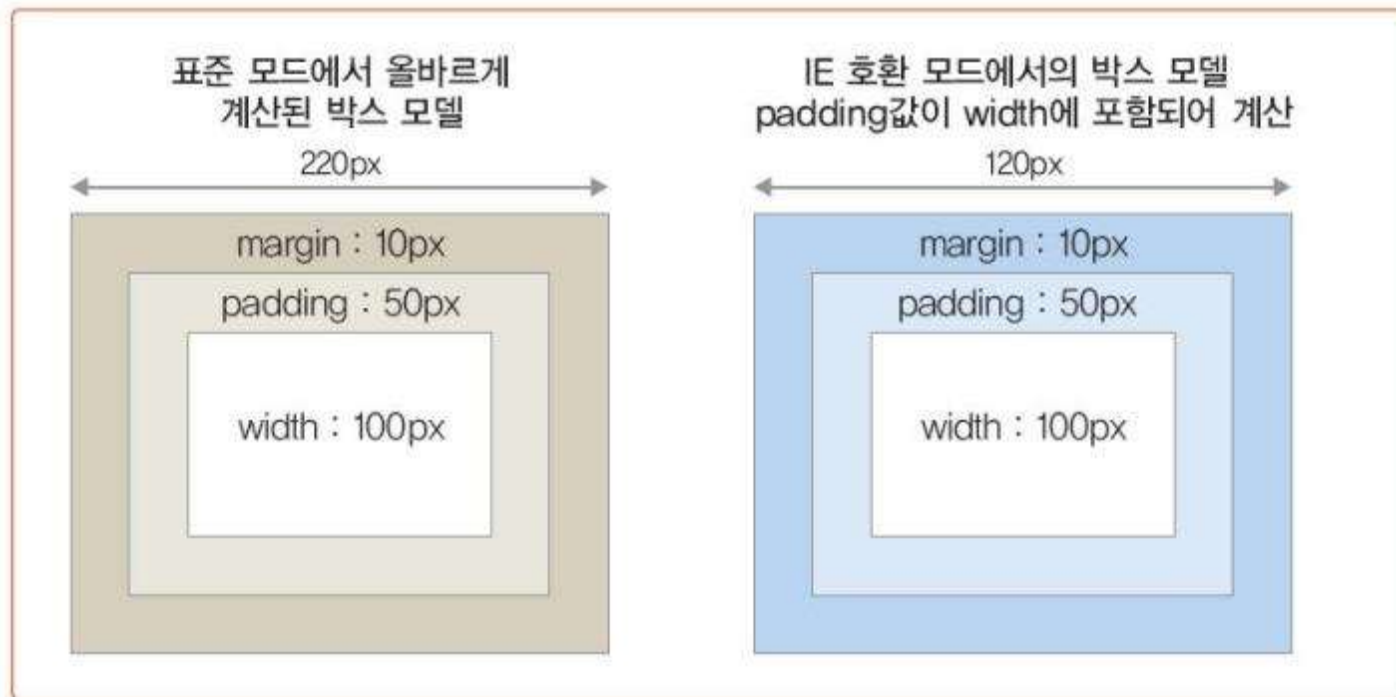
```
<body>
 <p>오늘은 비가 옵니다.</p>
 <p>오늘은 눈이 옵니다.</p>
 <p>오늘은 해가 납니다.</p>
 <p class="cloud">오늘은 흐립니다.</p>
</body>

</html>
```

# Box Model

## ❖ Box Model

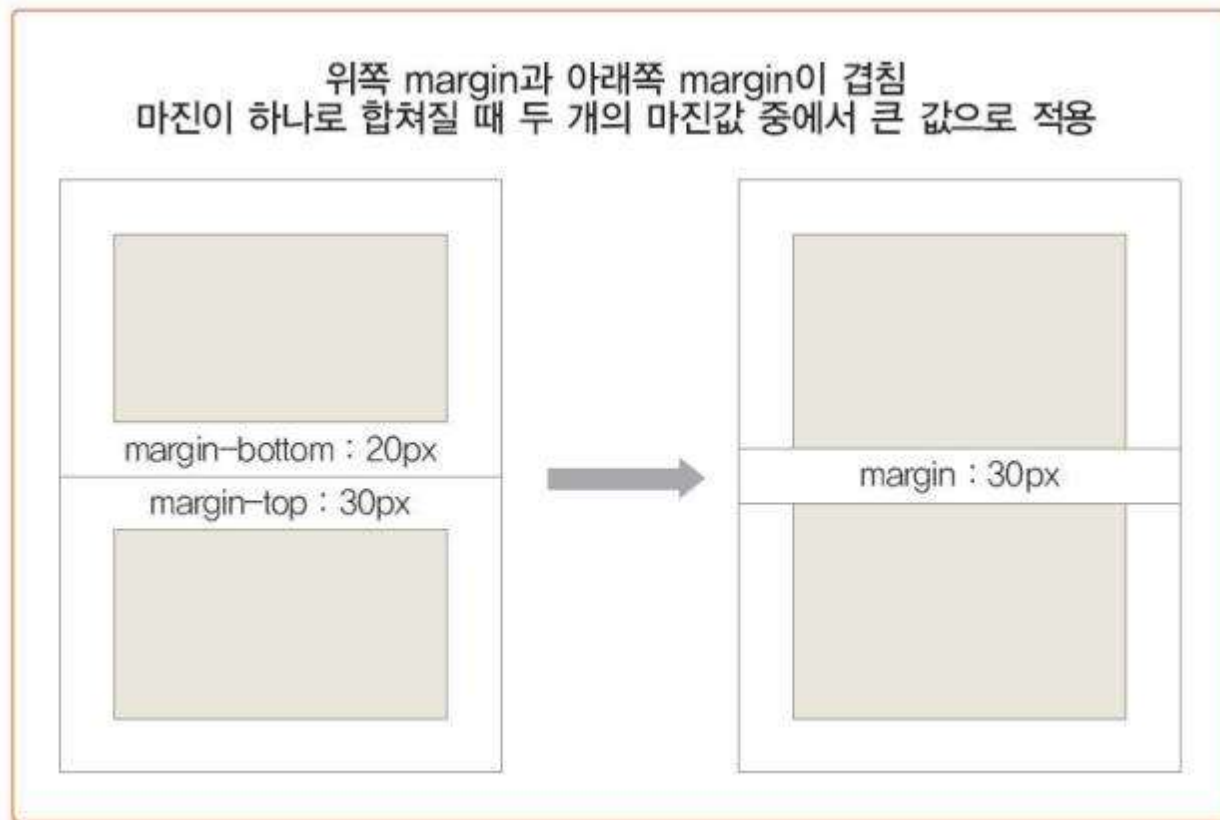
- ✓ XHTML 문서가 웹 브라우저에서 표준 모드로 렌더링 될 경우와 호환 모드로 렌더링될 경우 CSS의 박스 모델은 다르게 적용
- ✓ 인터넷 익스플로러의 경우 호환 모드에서는 padding 속성에 지정된 값을 width에 포함시켜 계산하기 때문에 의도한 것보다 박스가 작게 출력 될 수 있음



# Box Model

## ❖ Box Model

- ✓ margin collapsing (마진 겹침 현상): 두 개 이상의 margin값이 세로 방향으로 만났을 때 하나의 margin으로 합쳐지는 현상



# Box Model

## ❖ Box Model

- ✓ box-sizing
  - box 의 크기 설정 기준
  - content-box(내용 기준) 나 border-box(테두리 기준) 설정
- ✓ width
  - 콘텐츠의 너비를 설정하는 것으로 길이 단위나 백분율 사용
  - 상속 안됨
- ✓ height
  - 콘텐츠의 높이를 설정하는 것으로 길이 단위나 백분율 사용
  - 상속 안됨
- ✓ max-height, max-width, min-height, min-width 를 이용해서 상자의 최대 최소 크기를 설정

# Box Model

## ❖ 여백 설정

### ✓ padding

- 콘텐츠와 경계선 사이의 여백
- 단위를 이용한 크기 나 백분율로 설정
- 상하좌우 모든 여백이 동일하게 설정되는데 각각의 여백을 설정하고자 하는 경우는 padding-left, padding-right, padding-top, padding-bottom을 이용해서 설정

### ✓ margin

- 주위 박스 와 여백
- padding 과 동일한 방법으로 설정

# Box Model

## ❖ 보기 모드

### ✓ visibility

- 요소를 숨기고자 할 때 사용
- visible, hidden, collapse, inherit

### ✓ overflow

- 크기에 비해 요소의 콘텐츠가 너무 많은 경우에 사용하는 속성
- visible, hidden, scroll, auto

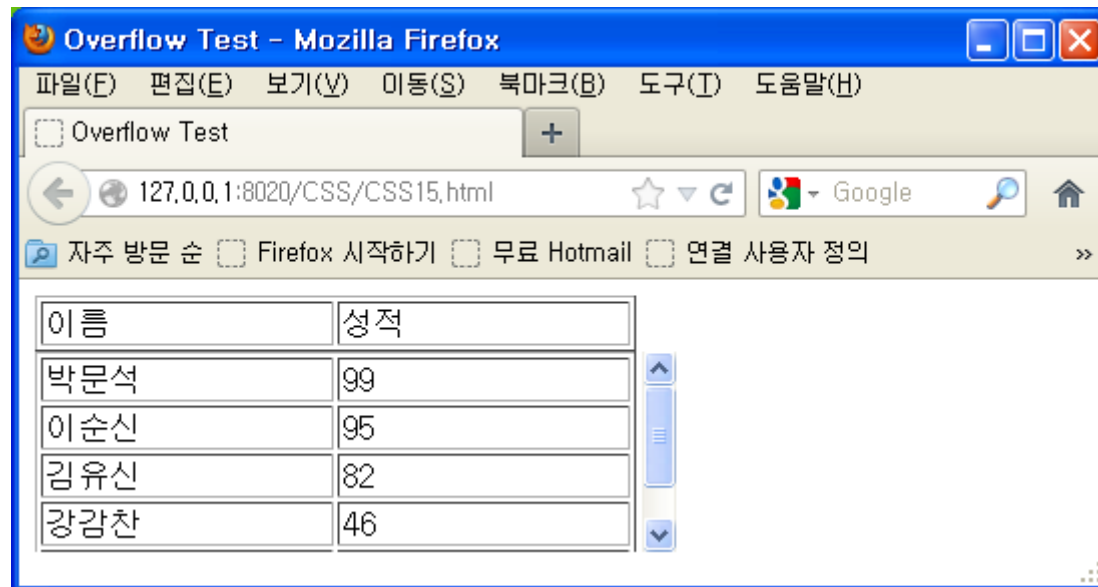
### ✓ text-overflow

- 박스 모델 안에서 텍스트를 표시할 때 줄을 바꾸지 못하도록 지정하고 넘치는 텍스트는 화면에 보이지 않도록 처리하는 경우에 사용할 수 있는 속성
- 설정 값으로는 넘치는 텍스트를 잘라버리는 clip과 텍스트가 잘렸다는 의미로 ... 을 표시해주는 ellipsis가 있음



# Box Model

## ❖ 보기 모드



# Box Model

## ❖ display.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Overflow Test</title>
</head>
<body>
 <table style="width:300px" border="1">
 <tr>
 <td width="150">이름 </td>
 <td width="150">성적 </td>
 </tr>
 </table>
```

# Box Model

## ❖ display.html

```
<div style="width: 320px; height: 100px; overflow: auto">
 <table style="width:300px" border="1">
 <tr>
 <td width="150">박문석</td>
 <td width="150">99</td>
 </tr>
 <tr>
 <td width="150">이순신</td>
 <td width="150">95</td>
 </tr>
 <tr>
 <td width="150">김유신</td>
 <td width="150">82</td>
 </tr>
 <tr>
 <td width="150">강감찬</td>
 <td width="150">46</td>
 </tr>
 <tr>
 <td width="150">을지문덕</td>
 <td width="150">72</td>
 </tr>
```

# Box Model

❖ display.html

```
<tr>
 <td width="150">연개소문</td>
 <td width="150">56</td>
</tr>
</table>
</div>
</body>
</html>
```



# Box Model

## ❖ float

- ✓ 요소의 박스를 화면에서 떠 있도록 하는 것으로 세로로만 나열되는 블록요소들을 가로로 어울리게 하기 위해서 사용
- ✓ float 한 박스에 가로 사이즈를 함께 주어야만 구형 브라우저에서도 적용됨
  - 값: left | right | none | inherit
    - ◆ left
      - 왼쪽으로 플로트 된 박스를 생성
      - 뒤에 계속되는 내용은 블록박스의 윗변의 오른쪽에서부터 시작
    - ◆ right
      - 오른쪽으로 플로트 된 박스를 생성
      - 뒤에 계속되는 내용은 블록박스의 윗변의 왼쪽에서부터 시작

# Box Model

## ❖ float

### ✓ 인라인 요소 플로팅

- 이미지는 폭을 지정할 필요 없지만 텍스트 요소는 폭을 지정해야 하는데 지정하지 않으면 콘텐츠 영역이 최대한 확장됨
- 플로팅된 텍스트 요소는 블록 요소처럼 디스플레이
- 플로팅된 요소의 여백은 축소되지 않음

### ✓ 블록 요소 플로팅

- 블록 요소도 폭을 지정하는 것이 좋음
- 플로팅된 블록 요소는 소스에서 나오는 위치에 상대적으로 왼쪽이나 오른쪽으로 플로팅

# Box Model

## ❖ float

### ✓ 해제

- clear 속성을 이용하여 해제
  - ◆ 값 : none | left | right | both | inherit
  - ◆ 블록 레벨 요소에만 적용 가능
- overflow 속성을 이용하여 해제
  - ◆ 값 : auto | hidden
  - ◆ [플로트 된 요소의 부모 요소에 지정]
- float 속성을 이용하여 해제
  - ◆ 값 : left | right
  - ◆ [플로트 된 요소의 부모 요소에 지정]

# Box Model

## ❖ 크기 조절

- ✓ `resize` 속성을 이용해서 박스의 모서리를 잡아당겨 크기를 조작할 수 있도록 함
- ✓ `horizontal`, `vertical`, `both` 으로 설정



# Box Model

## ❖ 그림자 영역

- ✓ box-shadow: 수평 수직 거리(오프셋), 흐릿함 과 확산 정도, 색상
- ✓ 첫번째 값으로 inset을 적용하면 그림자가 콘텐츠 안으로 생성

# Box Model

❖ 그림자 영역



# Box Model

## ❖ boxshadow.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Box-Shadow</title>
<style type="text/css">
 body {
 font: 12px "Arial";
 margin: 20px;
 }

 h1 {
 float: left;
 margin-right: 20px;
 }
```

# Box Model

## ❖ boxshadow.html

```
h1 a {
 display: block;
 font-size: 40px;
 text-decoration: none;
 text-align: center;
 color: #000;
 width: 200px;
 padding: 30px;
 border: 1px solid rgb(18, 52, 86);
 border-radius: 15px;
 background-color: #eee;
 box-shadow: inset 0px 0px 20px 15px rgba(18, 52, 86, 0.8);
}

h1 a:hover {
 background-color: #aff;
}
</style>
</head>
```

# Box Model

## ❖ boxshadow.html

```
<body>
 <h1>
 대한민국
 </h1>
</body>
</html>
```



# Box Model

## ❖ Border

- ✓ style은 경계선 모양으로 설정하지 않으면 경계선이 보이지 않으며 이 속성을 설정하지 않으면 다른 옵션은 무시되며 border-top-style, border-right-style, border-bottom-style, border-left-style 형태로 적용 가능
  - none, dotted, dashed, solid, double, groove, ridge, inset, outset 등의 옵션이 있음
- ✓ width는 테두리 두께로 각각은 border-top-width, border-right-width, border-bottom-width, border-left-width 로 지정 가능
  - 값으로는 thin, medium, thick 등이 올 수 있으며 숫자를 직접 기재 가능
- ✓ color는 색상으로 border-top-color, border-right-color, border-bottom-color, border-left-color로 각각 지정 가능
  - 투명한 보더(transparent) 사용 가능
- ✓ border: 두께 종류 색상 의 형태로 한꺼번에 설정하는 것이 가능

# Box Model

## ❖ Border

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에 뜬다.

# Box Model

## ❖ border.html

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="UTF-8" />
 <title>Box Model</title>
 <style type="text/css">
 p { width :600px ; padding : 5px ; }
 .p1{border:5px solid #123456;}
 .p2{border:5px dotted #123456;}
 .p3{border:5px dashed #123456;}
 .p4{border:10px groove #06c;}
 .p5{border:10px ridge #f06;}
 .p6{border:10px inset #0c3;}
 .p7{border:10px outset #c63;}
 .p8{border:none;}

 </style>
 </head>
```



# Box Model

❖ border.html

<body>

<p class="p1">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p2">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p3">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p4">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p5">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p6">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

<p class="p7">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

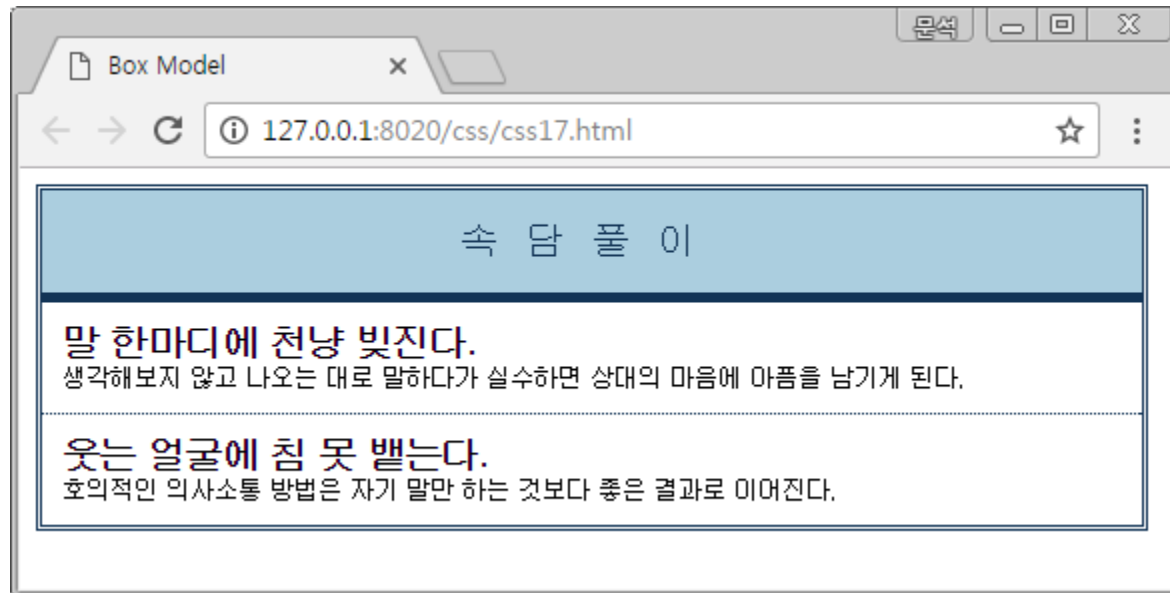
<p class="p8">파일을 편집하다 보면 문서를 검색하는 도중 프로젝트 도구가 브라우저 창에  
뜯는다. </p>

</body>

</html>

# Box Model

❖ borderline.html



# Box Model

## ❖ borderline.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<meta charset="UTF-8" />
<title>Box Model</title>
<style type="text/css">
 dl, dt, dd, p, h2 {
 margin: 0;
 padding: 0;
 }

 body {
 font: 12px "굴림", Gulim;
 }
```

# Box Model

## ❖ borderline.html

```
dl.box {
 width: 550px;
 border: 3px #123456 double;
}

dt {
 background: #abcedf;
 color: #123456;
 text-align: center;
 font-size: 18px;
 letter-spacing: 15px;
 padding: 15px;
 border-bottom: #123456 5px solid;
}

dd {
 padding: 10px;
}

dd.line {
 border-bottom: #123456 1px dotted;
}
</style>
</head>
```

# Box Model

## ❖ borderline.html

```
<body>
 <dl class="box">
 <dt>속담풀이</dt>
 <dd class="line">
 <h2>말 한마디에 천냥 빚진다.</h2>
 <p>생각해보지 않고 나오는 대로 말하다가 실수하면 상대의 마음에 아픔을 남
기게 된다.</p>
 </dd>
 <dd>
 <h2>웃는 얼굴에 침 못 뱉는다.</h2>
 <p>호의적인 의사소통 방법은 자기 말만 하는 것보다 좋은 결과로 이어진
다.</p>
 </dd>
 </dl>
</body>
</html>
```

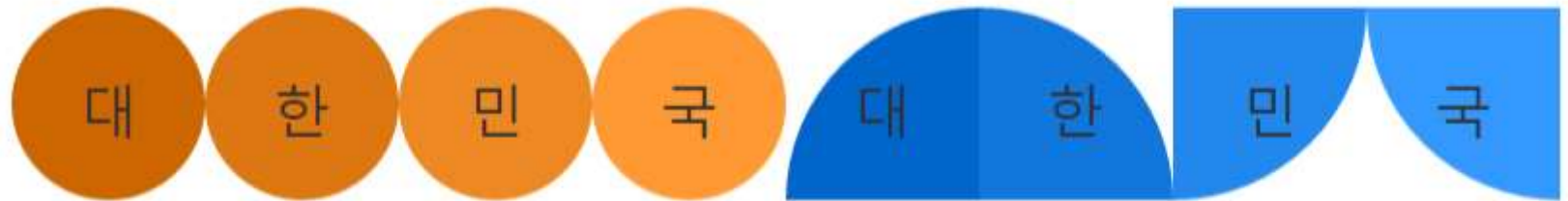
# Box Model

## ❖ border-radius

- ✓ css3에서 추가된 속성으로 border-radius를 이용해서 선에 라운드 효과를 설정 가능
- ✓ 값은 1개만 설정해서 4 방향 모두 설정 가능하고 순서대로 4개를 지정 할 수 있으면 ,로 구분해서 2번 이상 적용할 수 있음
- ✓ 원, 반원, 1/4 조각의 원 등을 만들 수 있음

# Box Model

❖ border-radius



# Box Model

## ❖ border-radius.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<meta charset="UTF-8" />
<title>Border Radius</title>
<style type="text/css">
 body {
 font: 30px "Arial Black";
 color: #333;
 margin: 20px;
 }

 li {
 width: 100px;
 text-align: center;
 height: 70px;
 float: left;
 padding-top: 30px;
 border-radius: 50px;
 list-style: none;
 }
```



# Box Model

## ❖ border-radius.html

```
.c1 {
 background-color: #C60;
}
```

```
.c2 {
 background-color: #d71;
}
```

```
.c3 {
 background-color: #e82;
}
```

```
.c4 {
 background-color: #f93;
}
```



# Box Model


## ❖ border-radius.html

```
.quarter1 {
 background-color: #06C;
 border-radius: 200px 0 0 0;
}

.quarter2 {
 background-color: #17d;
 border-radius: 0 200px 0 0;
}

.quarter3 {
 background-color: #28e;
 border-radius: 0 0 200px 0;
}

.quarter4 {
 background-color: #39f;
 border-radius: 0 0 0 200px;
}
</style>
</head>
```



# Box Model

## ❖ border-radius.html

```
<body>

 <li class="c1">대
 <li class="c2">한
 <li class="c3">민
 <li class="c4">국

 <li class="quater1">대
 <li class="quater2">한
 <li class="quater3">민
 <li class="quater4">국

</body>
</html>
```

# Box Model

## ❖ 이미지 보더

- ✓ 이미지 지정: `border-image-source:url("경로");`
- ✓ 이미지 슬라이스: `border-image-slice:값 값 값 값;`
- ✓ 반복 설정: `border-image-repeat: stretch, repeat, round, space`
- ✓ 경계선의 두께: `border-image-width`
- ✓ 경계선 밖으로 나오는 이미지의 거리: `border-image-outset`

# Box Model

- ❖ display: 요소의 출력 특성을 설정하는 속성
  - ✓ block: 박스를 블록 요소로 설정
  - ✓ inline: 박스를 인라인 요소로 설정 – 주위에 콘텐츠가 배치되지 않으며 height를 무시
  - ✓ inline-block: 박스를 인라인 요소로 변경하지만 주위에 콘텐츠가 배치되지 않고 height가 적용됨
  - ✓ flex: 1차원 레이아웃
  - ✓ grid: 2차원 레이아웃
  - ✓ none: 박스를 보이지 않도록 설정

# Box Model

❖ display

주메뉴

COMPANY

PRODUCT

SERVICE

COMMUNITY



# Box Model

## ❖ menu.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>display</title>
<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 body {
 font: 12px "굴림", Gulim;
 margin: 20px;
 }
```

# Box Model

❖ menu.html

```
#gnb li {
 display: inline;
}

#gnb li a {
 display: inline-block;
 background: #c00;
 color: #fff;
 width: 100px;
 height: 23px;
 padding-top: 10px;
 text-align: center;
}

#gnb li a:hover {
 background: #369;
}
</style>
</head>
```



# Box Model

❖ menu.html

```
<body>
```

```
 <div id="container">
```

```
 <h1>주메뉴</h1>
```

```
 <ul id="gnb">
```

```
 COMPANY
```

```
 PRODUCT
```

```
 SERVICE
```

```
 COMMUNITY
```

```

```

```
 </div>
```

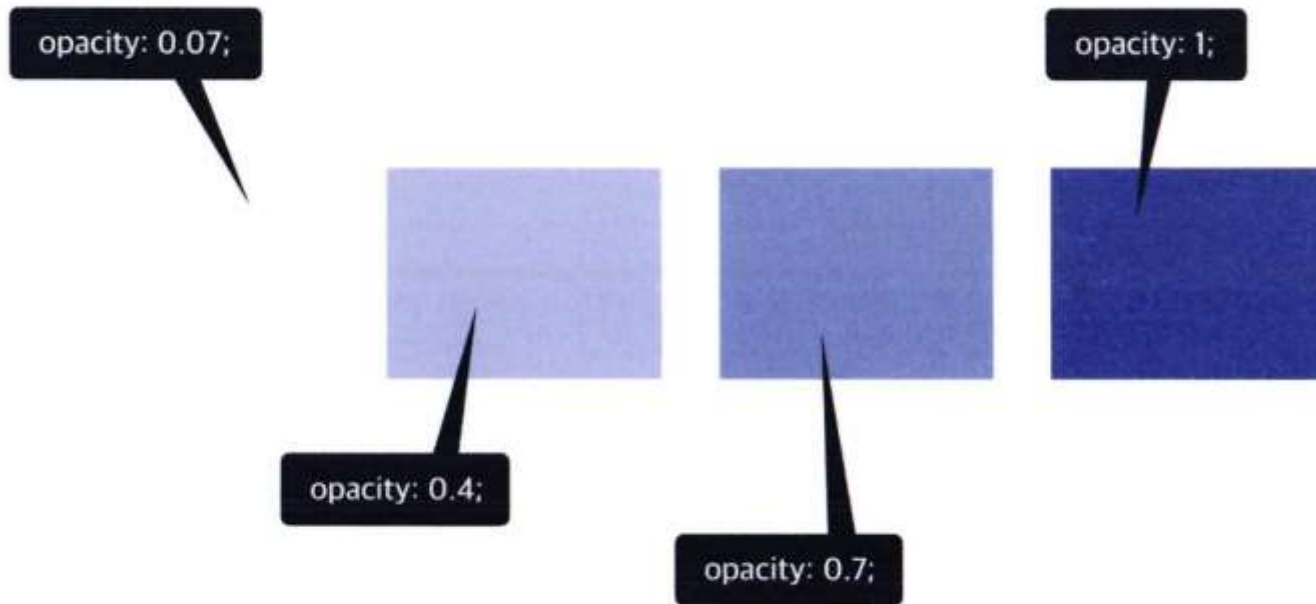
```
</body>
```

```
</html>
```

# Box Model

## ❖ opacity

- ✓ CSS에서는 opacity 속성으로 요소의 투명도(불투명도)를 지정할 수 있음
- ✓ 값으로 0과 1 사이의 소수를 사용하는데 기본값은 1이고 이것은 100% 불투명함을 의미하는데 투명하지 않음을 의미하며 opacity: 0.4;는 40%의 불투명도를 의미하고 opacity: 0;은 100% 투명함을 의미



# Box Model

## ❖ 배치

### ✓ position

- 요소의 배치 방법을 지정하는 속성으로 기본값은 static이고 요소의 위치 기준을 지정
  - ◆ static: 요소를 문서의 흐름에 맞추어서 배치(left, top 사용 불가능)하는 것으로 기본 값 - 위치 기준이 없는 것
  - ◆ relative: 이전 요소에 연결해서 상대적으로 배치하는 것으로 정상적인 문서 플로우에서 요소가 차지하던 공간이 공백으로 유지되며 left와 top 속성으로는 이동할 수 있음
  - ◆ absolute: 절대 배치를 하는 것으로 포함하고 있는 블록(부모 요소)의 네 변 기준
  - ◆ fixed: 웹 브라우저의 스크린을 기준으로 배치하는 방식으로 내용이 잘릴 수 있는데 문서가 스크롤되는 경우에도 특정 위치에 고정되는데 구형 브라우저에서는 문제가 발생할 수 있음
  - ◆ sticky: 스크롤 영역을 기준으로 배치
- position 속성에 absolute 또는 fixed 값이 지정된 경우 그 요소는 display 속성의 값이 block 으로 자동 변경되는데 블록 요소가 아닌 특정 요소에 position: absolute;나 position: fixed;를 입력했다면 별도로 display: block;을 지정할 필요가 없지만 display 속성에 flex와 grid 값을 지정 한 요소는 속성의 값이 block으로 자동 변경되지 않음

# Box Model

## ❖ 배치

### ✓ 요소의 위치

- top, right, bottom, left 속성을 이용해서 설정 가능
- 기본값은 auto 로 위치 값이 없음을 의미
- 단위(px, %, em)로 값을 지정하는데 여기에다가 음수를 사용할 수 도 있음
- 다음과 같이 입력하면 부모 요소를 기준으로 아래에서 30px, 오른쪽에서 30px 지점에 요소를 배치할 수 있음

position: absolute;

bottom: 30px;

right: 30px;

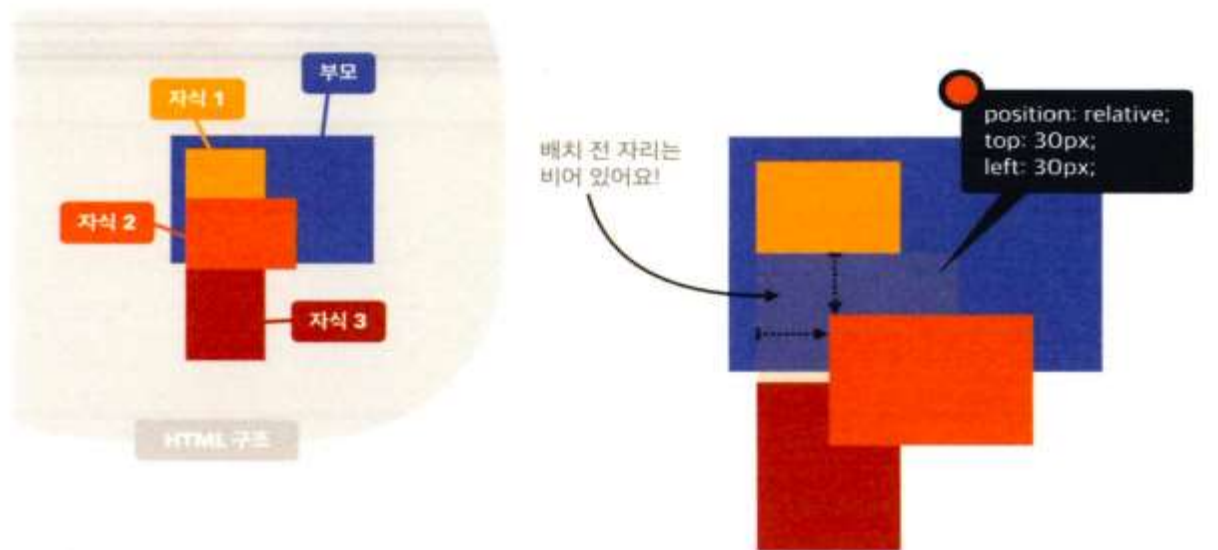
# Box Model

## ❖ 배치

### ✓ 기준에 대한 이해

#### ● relative

- ◆ relative 값은 요소들을 자기 자신을 기준으로 배치하는데 원래 자신이 있던 위치 기준으로 배치해야 하기 때문에 실제로는 움직이지 않음
- ◆ 이미지의 왼쪽을 보면 자식 요소 1~3번이 위에서부터 수직으로 쌓여있는데 여기서 자식 2 요소에 오른쪽과 같이 배치 스타일(relative)을 추가하더라도 자식 3 요소의 위치는 변하지 않기 때문에 자식 2 요소는 움직이지 않아 자식 3 요소가 그대로 쌓이고 자식 2 요소는 화면에 보이는 위치만 달라짐



# Box Model

## ❖ 배치

### ✓ 기준에 대한 이해

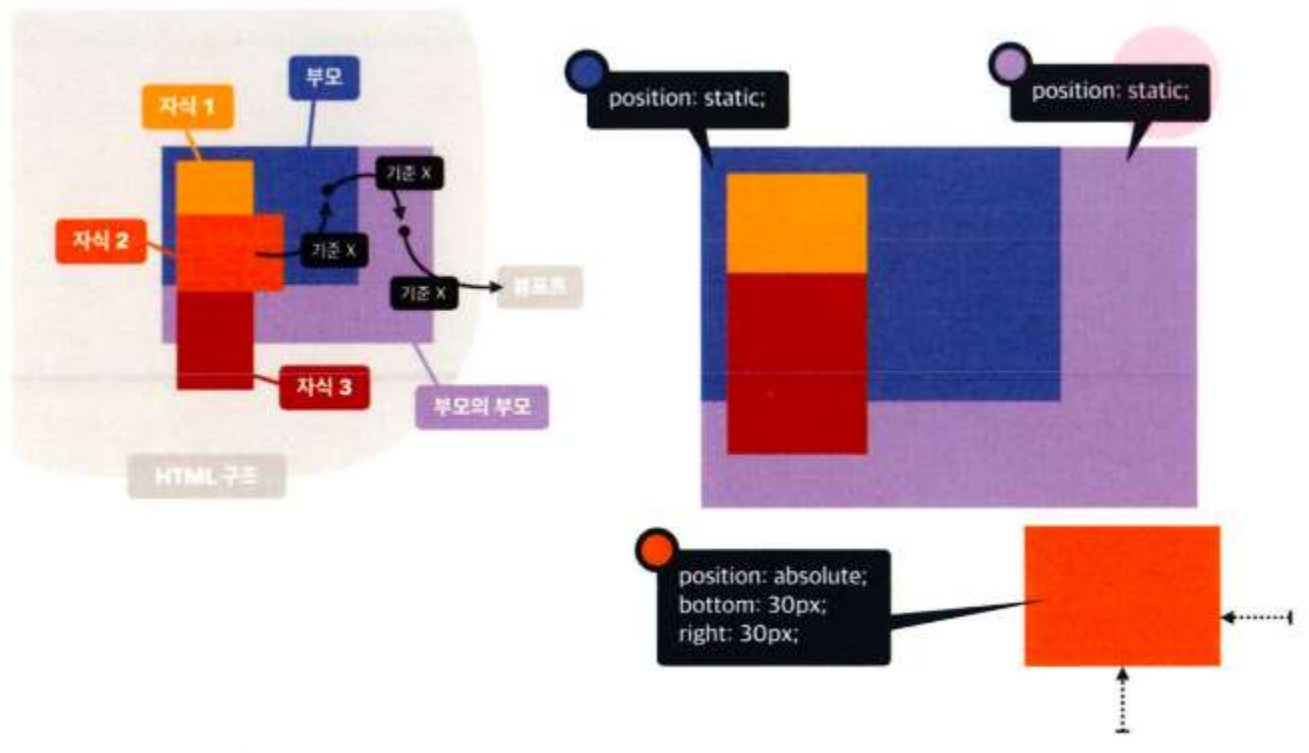
#### ● absolute

- ◆ absolute 값은 요소들을 위치상 부모 요소를 기준으로 배치하는데 위치상 부모 요소란 absolute 값을 가진 해당 배치 요소의 상위 요소 중 position 속성의 값 (static 제외)이 존재하는 가장 가까운 요소
- ◆ 이미지의 왼쪽을 보면 자식 요소 1~3번이 위에서부터 수직으로 쌓여있고 파란색의 부모 요소와 보라색의 상위 요소(부모의 부모)가 보이는데 자식 2 요소에 배치 스타일(absolute)을 추가하면 이 요소는 위치상 부모 요소를 찾아 그것을 기준으로 배치하기 때문에 가장 먼저 부모 요소를 확인하는데 오른쪽 이미지에서 확인할 수 있는 것처럼 파란색의 부모 요소는 position 속성의 값이 static이기 때문에 기준으로 삼을 수 없어서 한 단계 상위로 올라가 보라색 요소를 확인하지만 이 역시 position 속성의 값이 static이기 때문에 기준으로 삼을 수 없는데 이렇게 계속 상위 요소를 타고 올라가다가 더 이상 기준으로 삼을 요소를 발견하지 못하면 요소를 뷰 포트를 기준으로 배치

# Box Model

## ❖ 배치

- ✓ 기준에 대한 이해
  - absolute



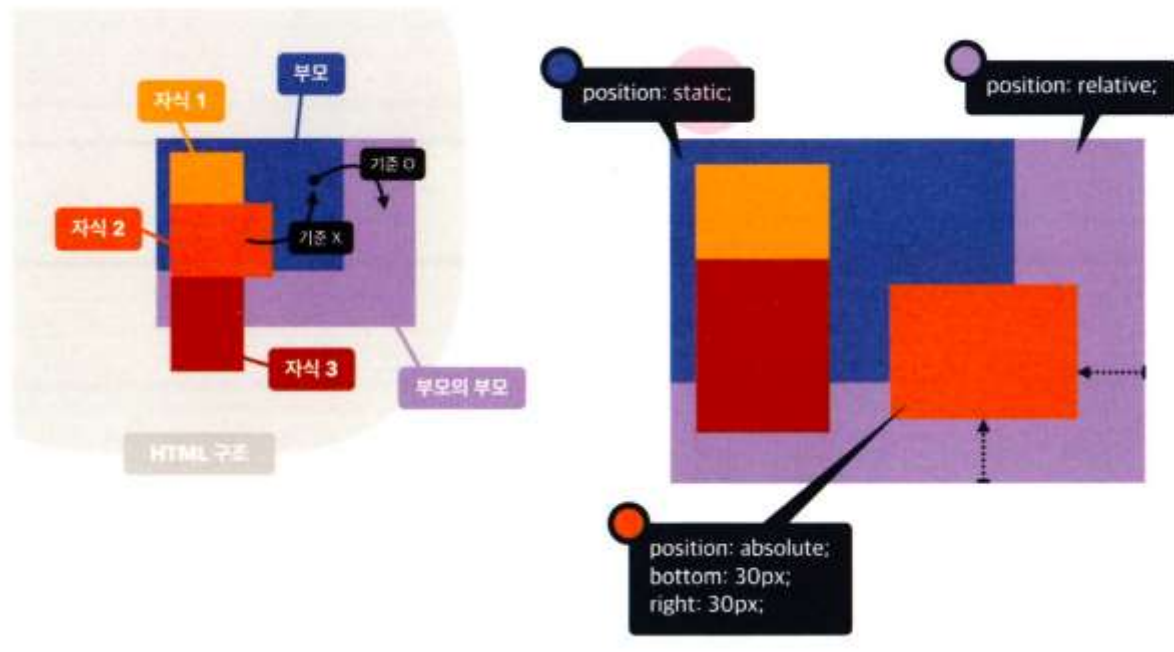
# Box Model

## ❖ 배치

### ✓ 기준에 대한 이해

#### ● absolute

- ◆ absolute 값을 제대로 사용하려면 상위 요소 중 기준으로 삼을 요소에 position 속성의 값을 추가해야 하는데 이미지와 같이 자식 2 요소의 부모의 부모 요소에 position: relative;와 같이 값을 지정하면 그 상위 요소를 기준으로 배치되며 상위 요소를 자식 2 요소의 위치상 부모 요소라고 볼 수 있음





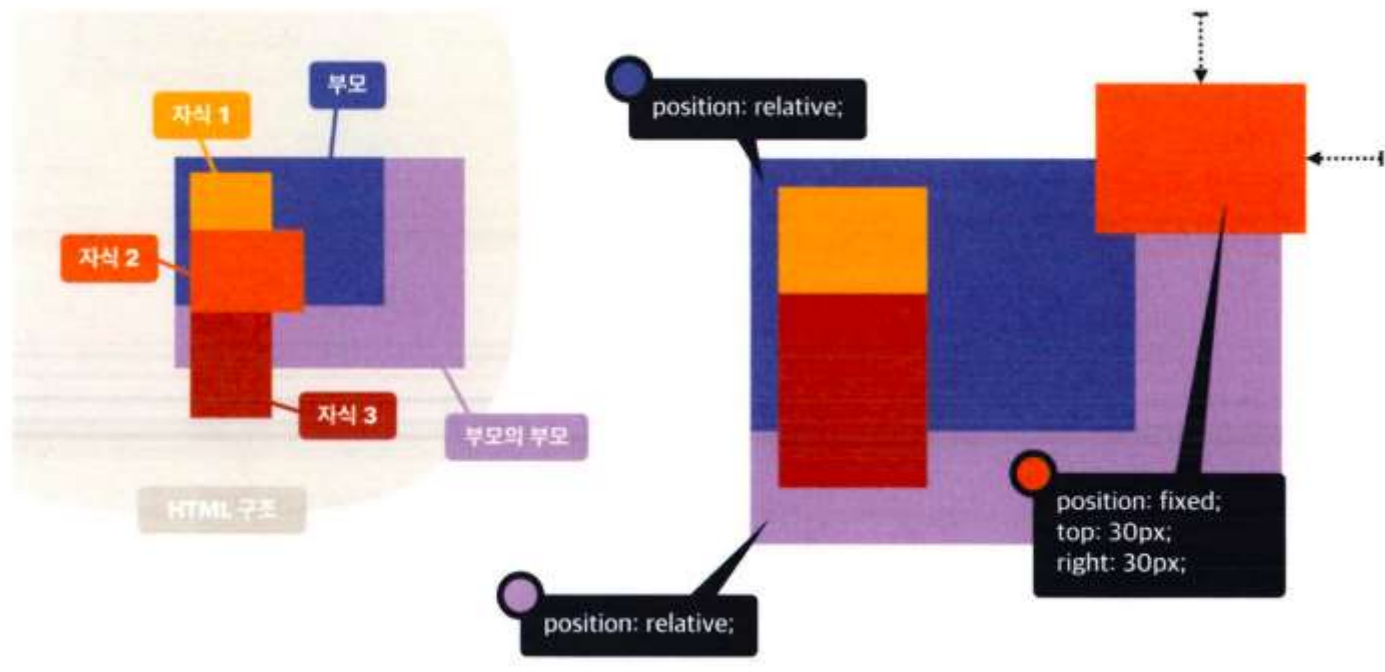
# Box Model

## ❖ 배치

### ✓ 기준에 대한 이해

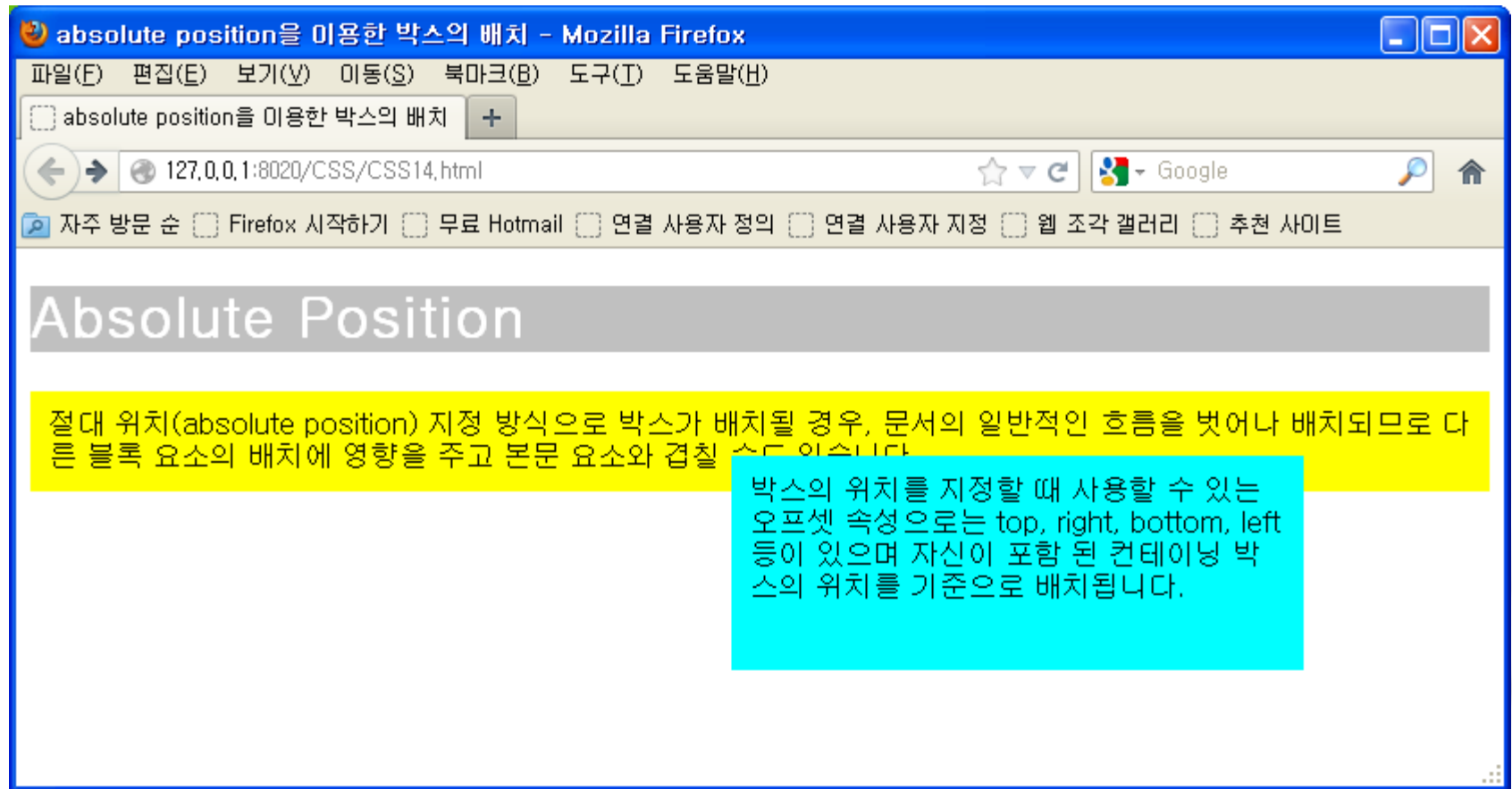
#### ● fixed

- ◆ fixed 값은 뷰포트를 기준으로 요소를 배치
- ◆ 화면을 스크롤해도 배치된 해당 요소의 위치가 고정
- ◆ 보통 광고 배너나 최상단으로 이동하는 버튼을 만들 때 사용



# Box Model

## ❖ 배치



# Box Model

## ❖ absolutePosition.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>absolute position을 이용한 박스의 배치</title>
<style type="text/css">
 h1 {
 background: silver;
 color: white;
 }
 p {
 background: yellow;
 padding: 10px;
 }
 .abs {
 position: absolute;
 top: 100px;
 left: 400px;
 width: 300px;
 height: 100px;
 background: aqua;
 }
</style>
</head>
```

# Box Model

## ❖ position.html

```
<body>
```

```
 <h1>Absolute Position</h1>
```

```
 <p class="abs">박스의 위치를 지정할 때 사용할 수 있는 오프셋 속성으로는 top, right,
 bottom,
```

```
 left 등이 있으며 자신이 포함 된 컨테이닝 박스의 위치를 기준으로 배치됩니다.</p>
```

```
 <p>절대 위치(absolute position) 지정 방식으로 박스가 배치될 경우, 문서의 일반적인 흐름
 을 벗어나
```

```
 배치되므로 다른 블록 요소의 배치에 영향을 주고 본문 요소와 겹칠 수도 있습니다.</p>
```

```
</body>
```

```
</html>
```

# Box Model

❖ 배치



뜨거운 웹표준,  
시원하게 알려드려요!

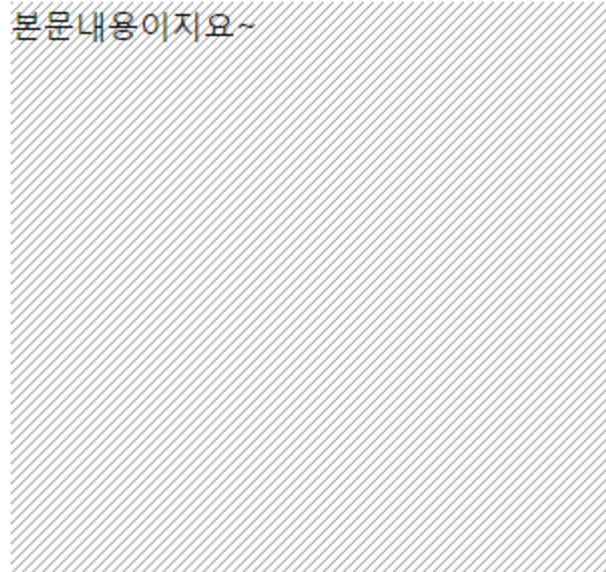
COMPANY

PRODUCT

SERVICE

COMMUNITY

본문내용이지요~



# Box Model

❖ [fixedposition.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>fixed position을 이용한 박스의 배치</title>
<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 body {
 margin: 10px;
 }
</style>
</head>
<body>
```

# Box Model

## ❖ fixedposition.html

```
#header {
 position: fixed;
 width: 200px;
 left: 50px;
 top: 30px;
}
```

```
h1 {
 width: 250px;
 font: bold 15px "굴림", Gulim;
 color: #123456;
}
```

```
h1 img {
 display: block;
}
```

# Box Model

## ❖ fixedposition.html

```
#gnb {
 width: 150px;
 margin: 10px 0;
}

#gnb li a {
 display: block;
 background: #123456;
 margin: 2px 0;
 padding: 5px;
 color: #fff;
 font: bold 16px "Arial";
 text-decoration: none;
}

#gnb li a:hover {
 background: #456789;
}
```



# Box Model

## ❖ fixedposition.html

```
#container {
 width: 300px;
 height: 1500px;
 margin-left: 250px;
 background: url("images/bg_sasun.gif");
}

#footer {
 width: 550px;
}

address {
 font-style: normal;
 background: #000;
 padding: 10px;
 color: #fff;
}
</style>
</head>
```

# Box Model

## ❖ fixedposition.html

```
<body>
 <div id="header">
 <h1>
 뜨거운 웹표준,
 시원
 하게
 열려드려요!
 </h1>
 <ul id="gnb">
 COMPANY
 PRODUCT
 SERVICE
 COMMUNITY

 </div>
 <div id="container">
 <p>본문내용이지요~
 <p>
 </div>
 <div id="footer">
 <address>Copyright Allright Reserved.</address>
 </div>
</body>
</html>
```

# Box Model

## ❖ 배치

### ✓ z-index

- 요소 박스가 겹쳐지는 순서를 지정할 때 사용하며 숫자가 높을 수록 위에 배치
- 값 : auto | 정수(1, 2, 3 ...)

z-index : auto;

z-index : 1;

z-index : -1;

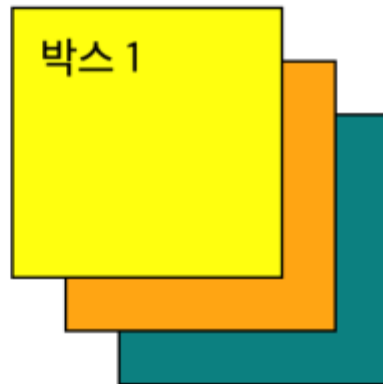
### ✓ clip

- 영역을 잘라내고자 하는 경우 사용하며 값은 rect(top, right, bottom, left) 형태로 지정

# Box Model

❖ 배치

## 박스의 겹치는 순서 지정하기



# Box Model

## ❖ zindex.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>z-index 속성을 이용한 박스의 겹치는 순서</title>
<style type="text/css">
 h1 {
 background-color: silver;
 }

 p {
 border: 1px solid black;
 padding: 10px;
 width: 80px;
 height: 80px;
 position: absolute;
 }
```

# Box Model

❖ zindex.html

```
.box1 {
 background: yellow;
 top: 100px;
 left: 100px;
 z-index: 3;
}

.box2 {
 background: orange;
 top: 120px;
 left: 120px;
 z-index: 2;
}

.box3 {
 background: teal;
 top: 140px;
 left: 140px;
 z-index: 1;
}
</style>
</head>
```

# Box Model

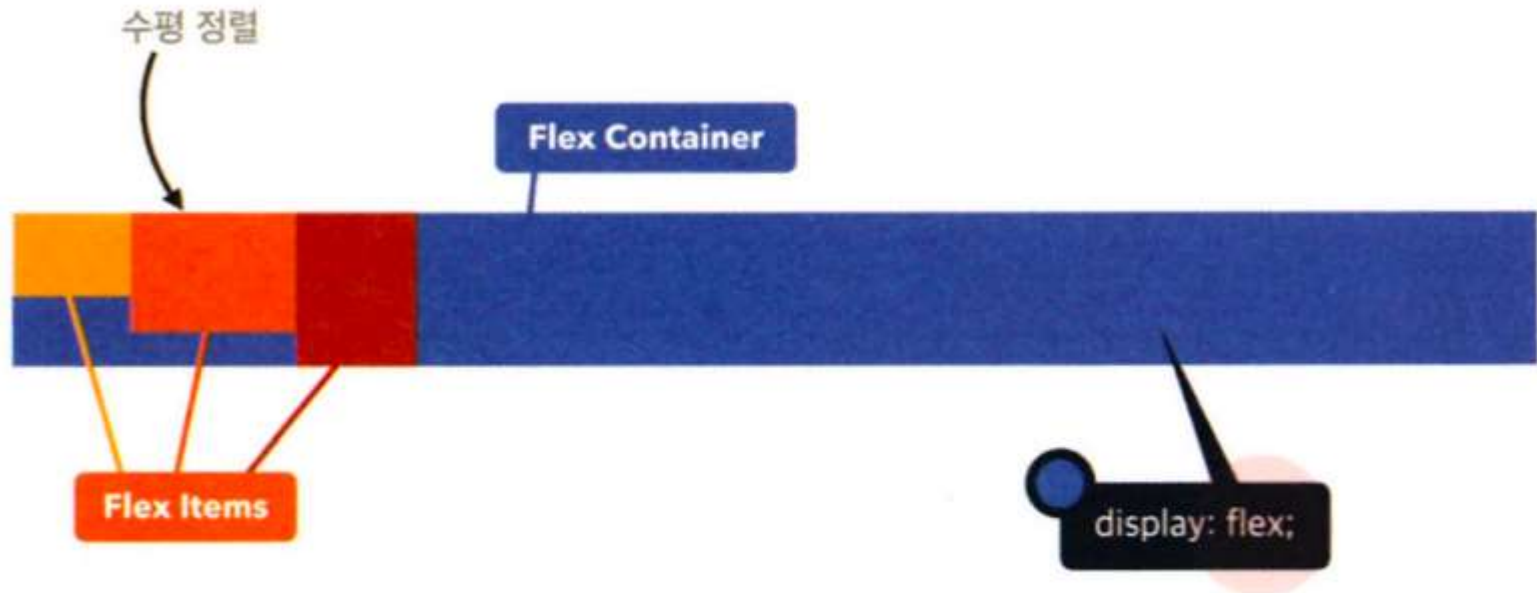
❖ zindex.html

```
<body>
 <h1>박스의 겹치는 순서 지정하기</h1>
 <p class="box1">박스 1</p>
 <p class="box2">박스 2</p>
 <p class="box3">박스 3</p>
</body>
</html>
```

# Flex

## ❖ flex

- ✓ 플렉스(Flex)는 1차원 레이아웃을 만드는 기술로 차원이란 x축(수평)이나 y축(수직) 같은 하나의 축(axis)을 의미하는데 1차원 레이아웃이란 수평 축으로만 정렬하거나 수직 축으로만 정렬하는 방법을 의미
- ✓ 플렉스는 크게 Flex Container와 Flex Items라는 두 가지 개념으로 살펴볼 수 있음
- ✓ Flex Container는 정렬할 요소들의 묶음을 의미하고 Flex Items는 정렬할 요소를 의미



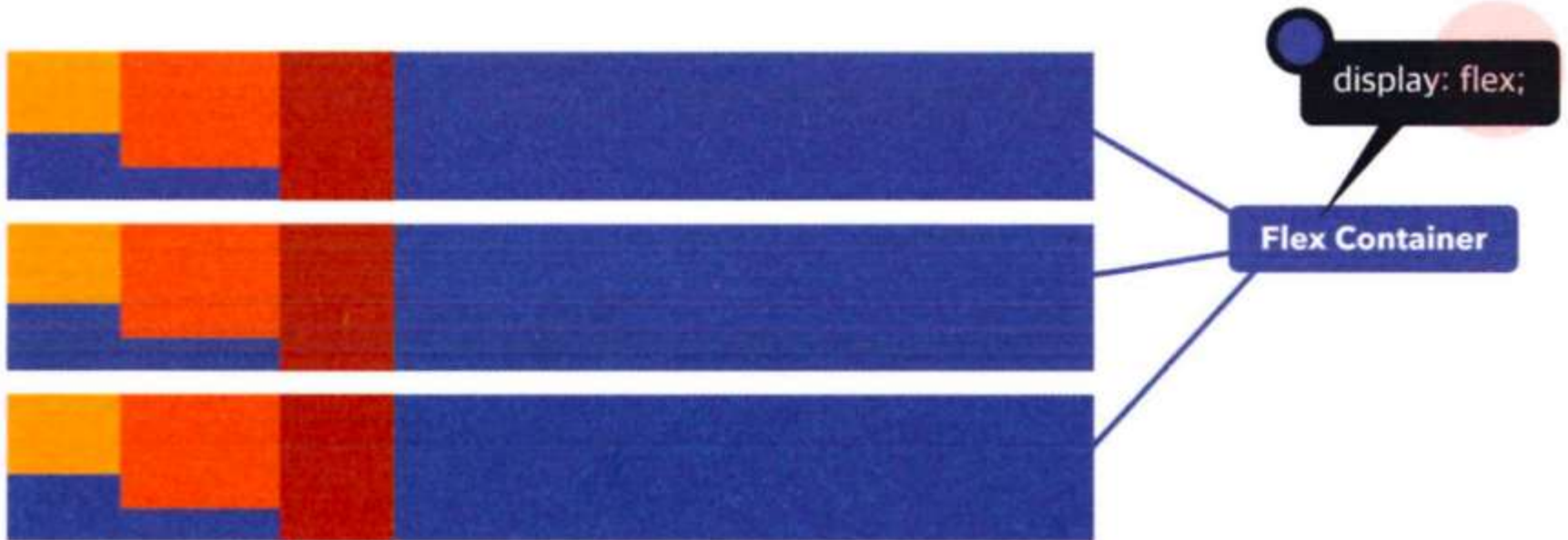
Flex의 개념



# Flex

## ❖ display

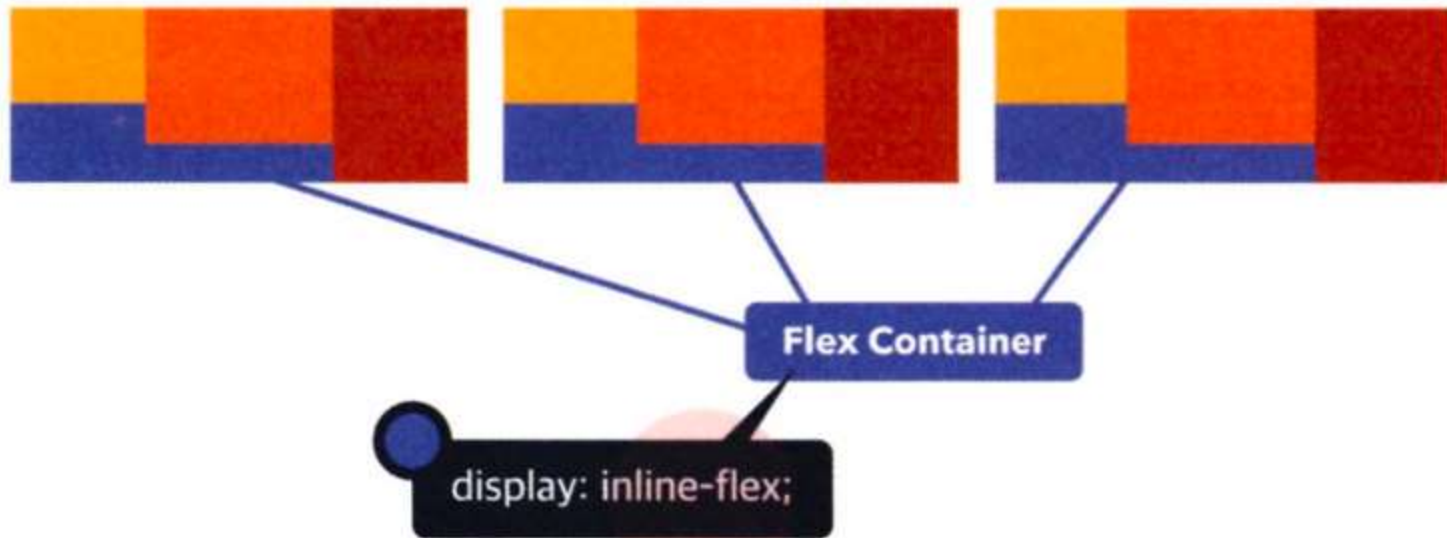
- ✓ flex 값을 적용하면 요소가 Flex Container로 정의되며 해당 요소의 자식 요소들은 모두 Flex Items로 자동 정의됨



# Flex

## ❖ display

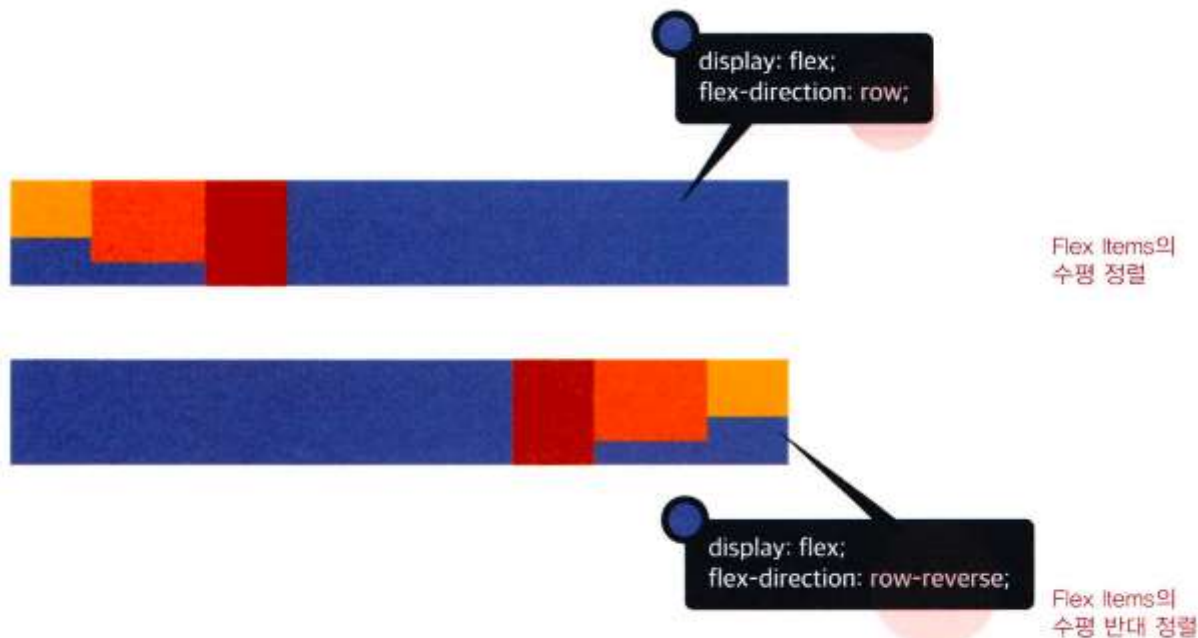
- ✓ 값으로 inline-flex를 적용하면 Flex Container가 인라인 요소의 특징을 가지는데 하나의 Flex Container?} 글자처럼 취급



# Flex

## ❖ flex-direction

- ✓ display 속성을 통해 요소를 Flex Container로 정의했다면 flex-direction 속성을 사용할 수 있는데 이는 Flex Container의 주축(Main-axis, 수평인지 수직인지)의 방향을 지정해주는 속성으로 기본값은 row이며 row는 행 축을 의미하며 Flex Items가 수평으로 정렬되며 값으로 column을 지정할 수도 있는데 column은 열 축을 의미하며 Flex Items가 수직으로 정렬



# Flex

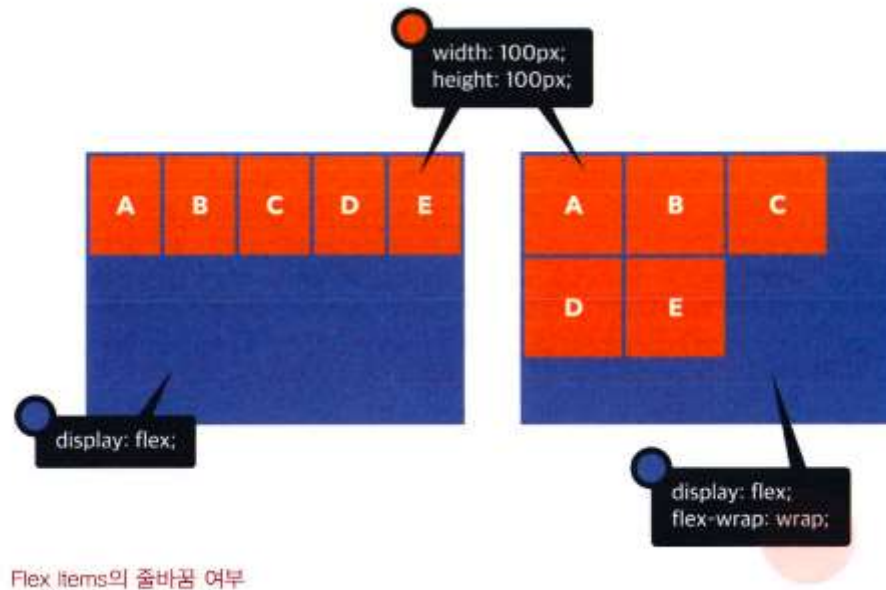
## ❖ flex-direction



# Flex

## ❖ flex-wrap

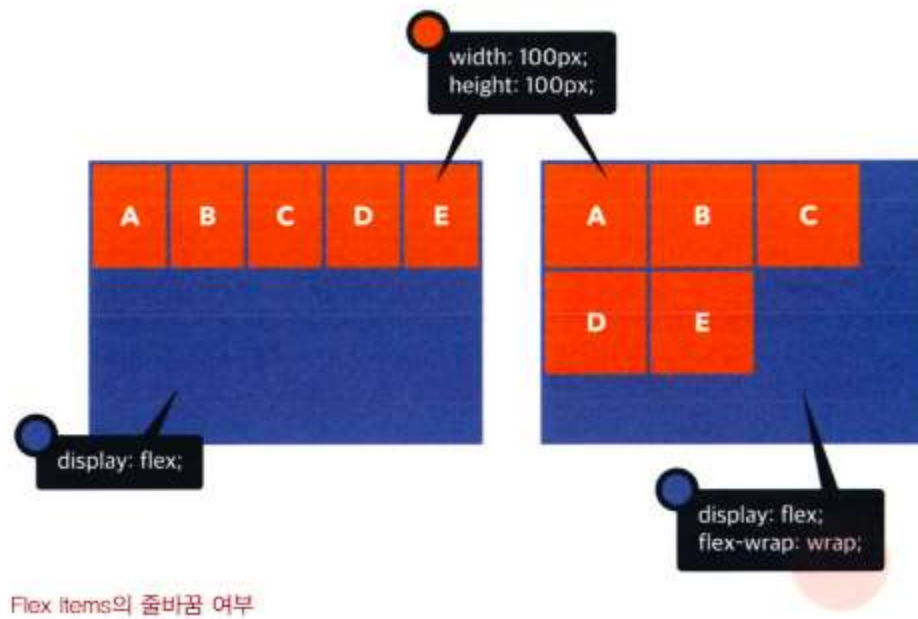
- ✓ flex-wrap 속 성은 Flex Items의 줄 바꿈(묶음) 여부를 지정하는 속성
- ✓ 기본값은 nowrap으로 줄 바꿈(묶음)이 없음을 의미하기 때문에 Flex Items는 도형들을 한 줄로만 정렬하고 정렬할 자리(너비)가 모자라면 Flex Items의 너비가 자동으로 줄어듬
- ✓ wrap 값을 지정할 경우 Flex Items를 정렬할 자리가 모자라면 자동으로 줄 바꿈을 시도



# Flex

## ❖ justify-content

- ✓ flex-wrap 속 성은 Flex Items의 줄 바꿈(묶음) 여부를 지정하는 속성
- ✓ 기본값은 nowrap으로 줄 바꿈(묶음)이 없음을 의미하기 때문에 Flex Items는 도형들을 한 줄로만 정렬하고 정렬할 자리(너비)가 모자라면 Flex Items의 너비가 자동으로 줄어듬
- ✓ wrap 값을 지정할 경우 Flex Items를 정렬할 자리가 모자라면 자동으로 줄 바꿈을 시도



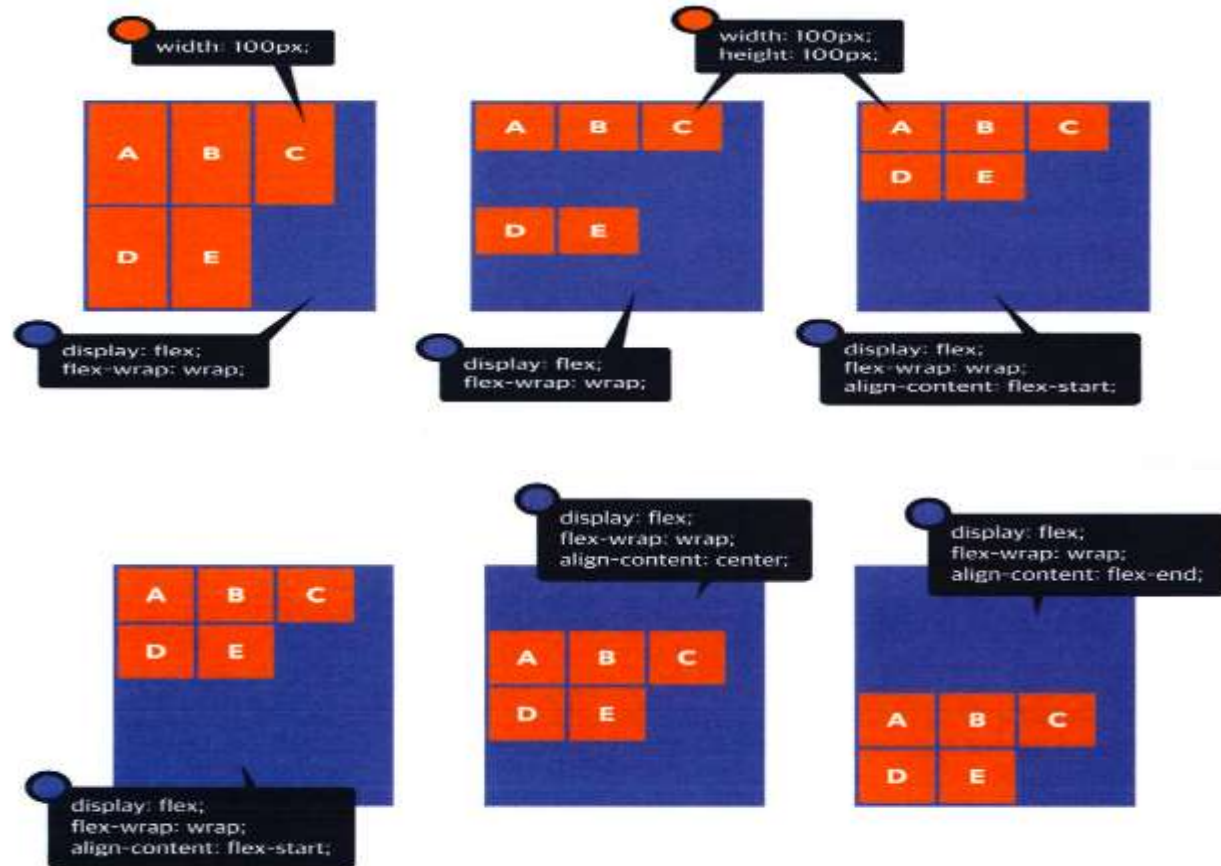
# Flex

## ❖ align-content

- ✓ align-content 속성은 Flex Container 교차 축(Cross-axis)의 정렬 방법을 지정하며 기본값은 Stretch
- ✓ 수평 정렬을 기준으로 설명하면, Flex Items에 세로 너비가 auto인 경우 다음 이미지의 첫 번째 예시처럼 최대한 늘어나고 그 외 필수로 Flex Items의 세로 너비가 있고 flex-wrap: wrap; 속성과 값을 통해 Flex items가 2줄 이상 줄 바꿈된 경우 다음 이미지와 같이 flex-start, flex-end, center 등의 값을 지정해 정렬할 수 있음
- ✓ align-content 속성은 요소가 딱 한 줄만 있는 경우에는 사용할 수 없음

# Flex

## ❖ align-content

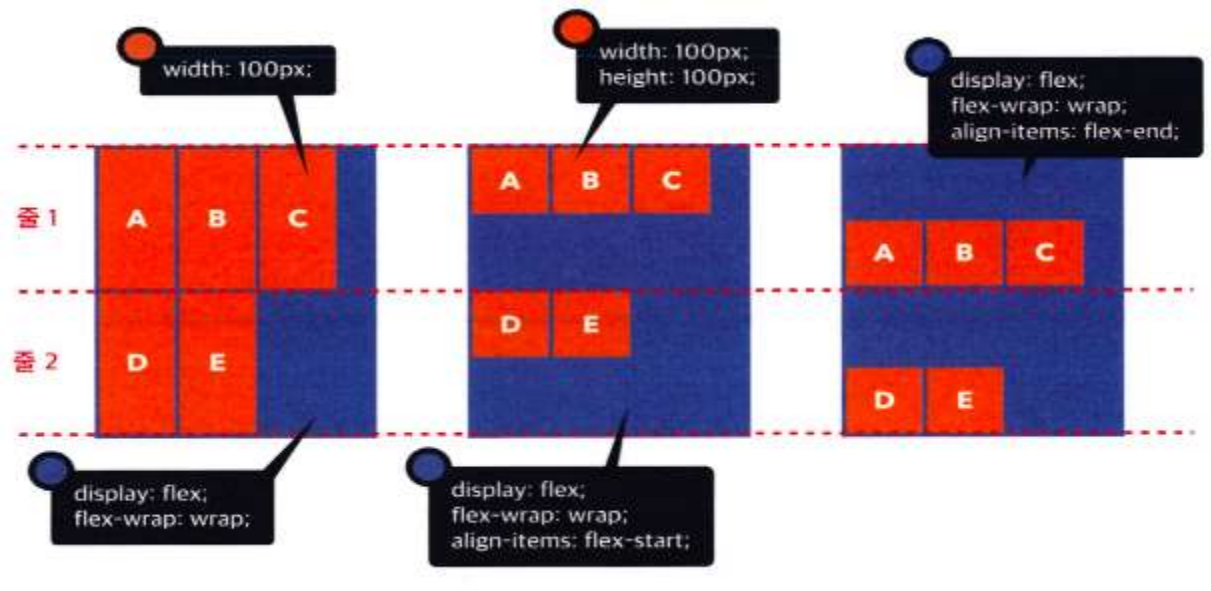




# Flex

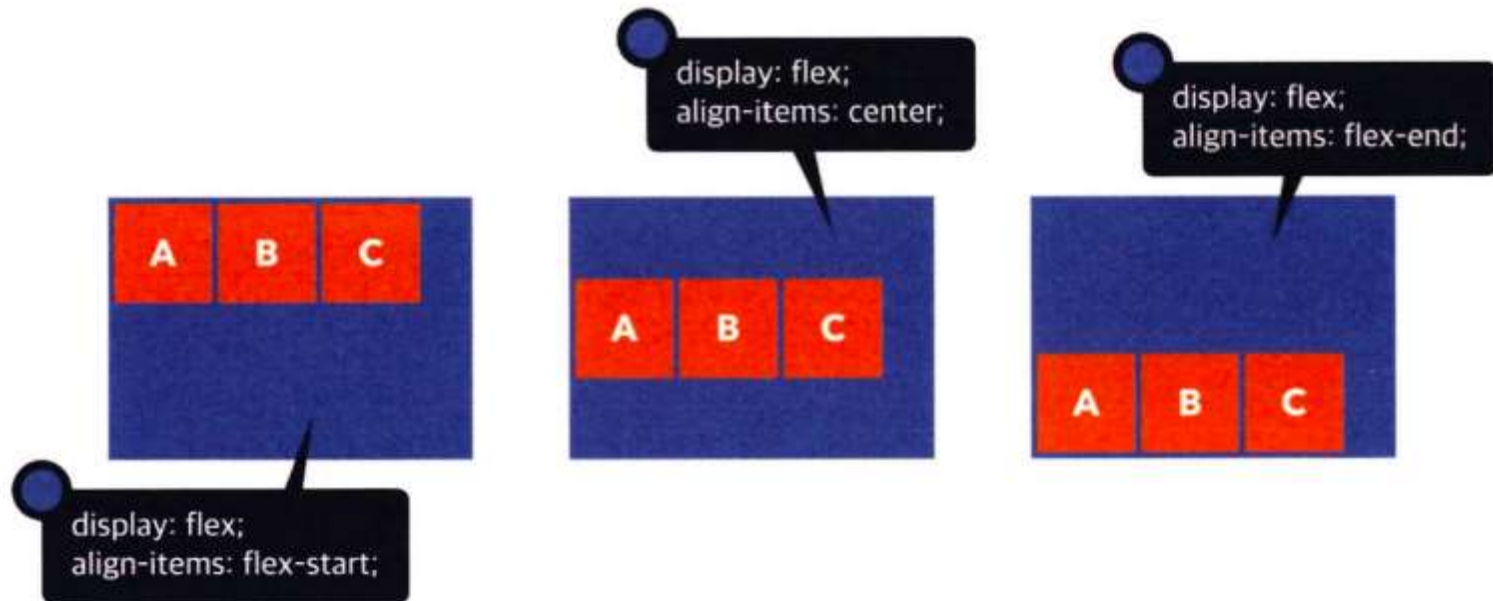
## ❖ align-items

- ✓ align-items 속성은 align-content 속성과 같이 Flex Container의 교차축(Cross-axis) 정렬 방법을 지정
- ✓ 한 줄에서의 정렬 방법을 지정
- ✓ 기본값은 align-content 속성과 같이 stretch
- ✓ flex-start, center, flex-end 등의 값을 지정하는 것도 가능
- ✓ align-content 속성과 개념이 다르긴 하지만 한 줄만 있어도 동작하기 때문에 유용한 속성



# Flex

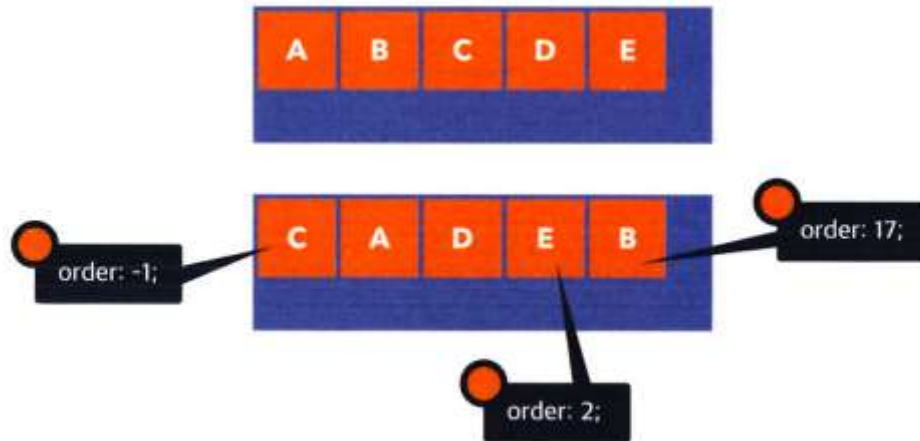
## ❖ align-items



# Flex

## ❖ order

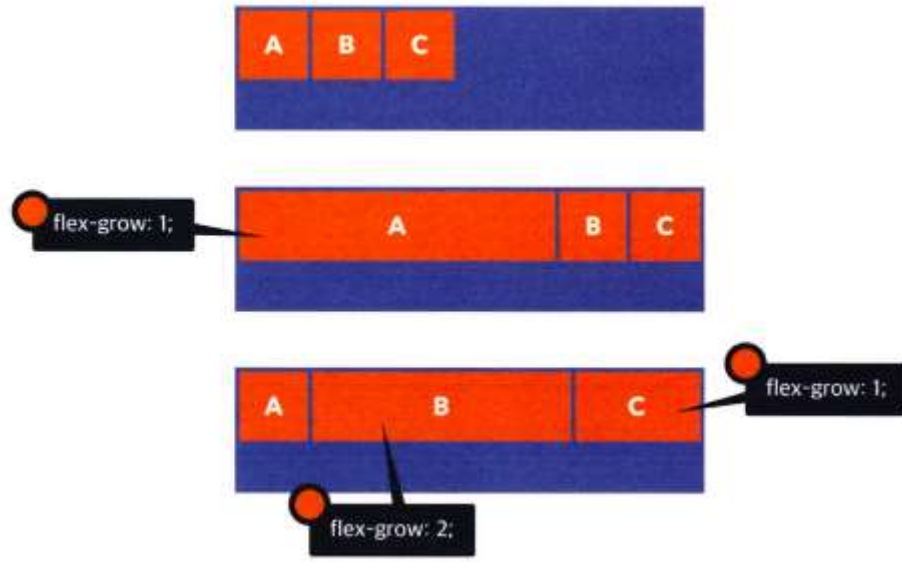
- ✓ order 속성은 Flex Items의 순서를 지정
- ✓ 기본값은 0으로 순서가 없음을 의미
- ✓ 값으로는 숫자를 입력하는데 여기에는 음수 사용이 허용되는데 지정된 숫자가 클수록 순서가 뒤로 밀리는 성질이 있음



# Flex

## ❖ flex-grow

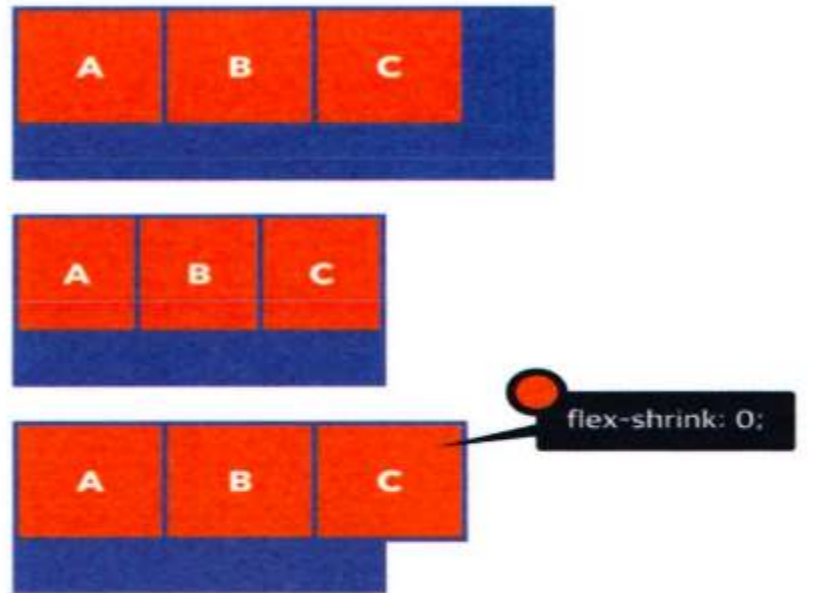
- ✓ flex-grow 속성은 Flex Items의 증가 너비 비율을 지정
- ✓ 기본값은 0 으로 증가되는 너비의 비율이 없는 것인데 단위를 사용하지도 않음
- ✓ 증가 너비 비율이 클수록 Flex Container 안에서 더 많은 너비를 차지
- ✓ 수평 정렬을 기준으로 width 속성을 통해 요소의 가로 너비를 지정해둔다고 해도 flex-grow 속성으로 증가 너비 비율을 지정하는 순간 요소의 너비가 늘어남



# Flex

## ❖ flex-shrink

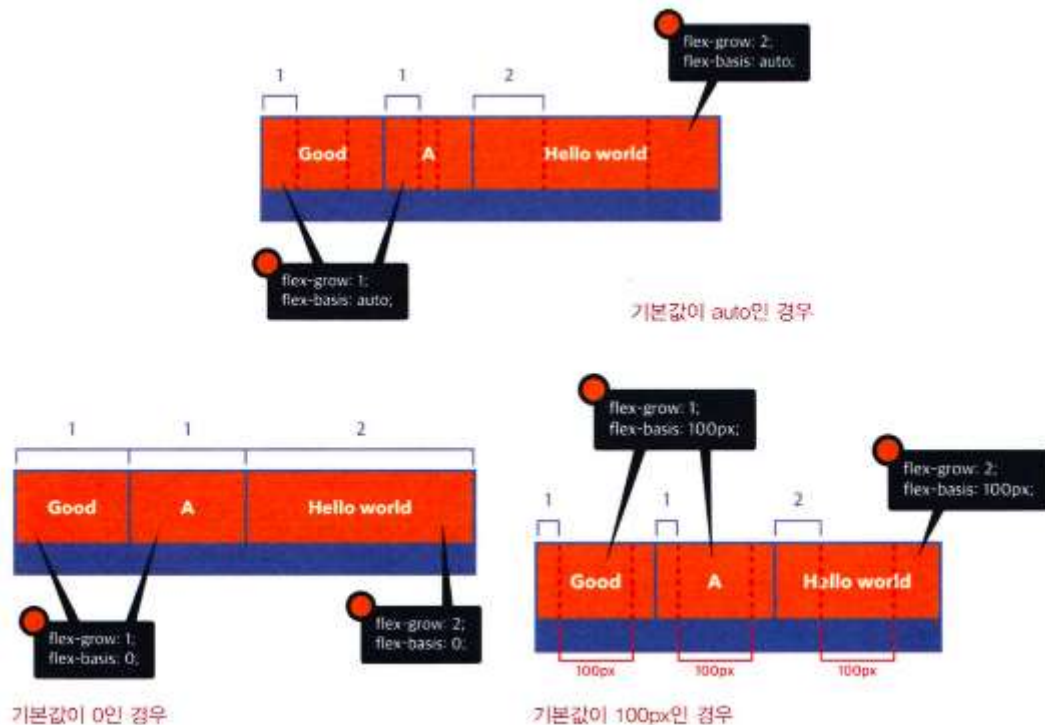
- ✓ flex-shrink 속성은 Flex Items의 감소 너비 비율을 지정
- ✓ 기본값은 1로 감소되는 자동으로 감소
- ✓ 감소 너비 비율을 0으로 설정하면 자동으로 너비가 감소되지 않음



# Flex

## ❖ flex-basis

- ✓ flex-basis 속성은 Flex Items의 공간이 배분되기 전의 기본 너비를 지정
- ✓ 기본값은 auto로 요소 내용(Content)의 너비를 기본 너비로 사용
- ✓ 값을 0으로 지정하여 기본 너비가 없도록 만들 수도 있는데 만약 기본 너비가 있는 경우 기본 너비를 제외한 나머지 너비를 증가 너비 비율로 계산



# 다단

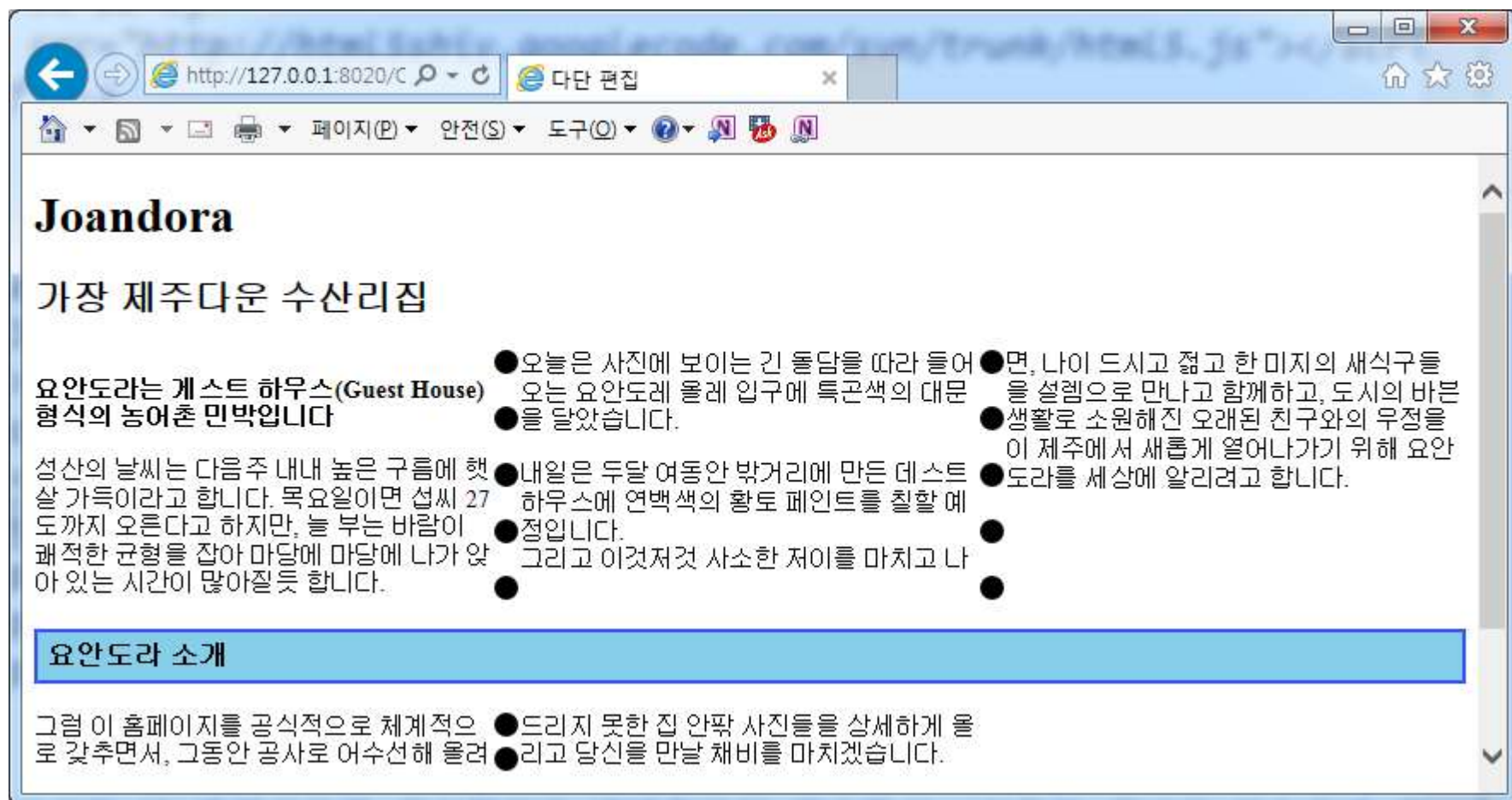
## ❖ 다단

- ✓ 이전에는 div를 이용해서 다단을 생성
- ✓ column 으로 시작하는 속성을 이용해서 다단 생성 가능한데 아직은 Vendor Prefix를 이용
  - column-count: 단의 개수
  - column-gap: 단 사이의 여백
  - column-rule-color: 단 사이의 구분선 색상
  - column-rule-style: 구분선의 모양
  - column-rule-width: 구분선의 두께
  - column-span: 단 합치기



# 다단

❖ multistage.html





# 다단

## ❖ multistage.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>다단 편집</title>
<style>
 .multicol {
 -webkit-columns: auto 3;
 -webkit-column-rule: 1px dotted black;
 -moz-columns: auto 3;
 -moz-column-rule: 1em dotted black;
 columns: auto 3;
 column-rule: 1em dotted black;
 }
 .subtitle {
 -webkit-column-span: all;
 -moz-column-span: all;
 column-span: all;
 background: skyblue;
 padding: 5px;
 border: 2px double blue;
 }
</style>
</head>
```

# 다단

## ❖ multistage.html

```
<body>
 <div class="container">
 <h1>Joandora</h1>
 <h2>가장 제주다운 수산리집</h2>
 <section class="multicol">
 <article>
 <p>
 요안도라는 게스트 하우스(Guest House) 형식
의 농어촌 민박입니다
 </p>
 <p>성산의 날씨는 다음주 내내 높은 구름에 햇살 가득이라고 합니
다. 목요일이면 섭씨 27도까지 오른다고
 하지만, 늘 부는 바람이 쾌적한 균형을 잡아 마당에 마
당에 나가 앉아 있는 시간이 많아질듯 합니다.</p>
 <p>오늘은 사진에 보이는 긴 돌담을 따라 들어오는 요안도레 올레
입구에 특곤색의 대문을 달았습니다.</p>
 <p>
 내일은 두달 여동안 밖거리에 만든 데스트 하우스에 연
백색의 황토 페인트를 칠할 예정입니다.
 그리고
 이것저것 사소한 저이를 마치고 나면, 나이 드시고 젊고
한 미지의 새식구들을 설렘으로 만나고 함께하고, 도시의 바쁜
 생활로 소원해진 오래된 친구와의 우정을 이 제주에서
새롭게 열어나가기 위해 요안도라를 세상에 알리려고 합니다.
 </p>
 </article>
 </section>
 </div>
</body>
```

# 다단

❖ multistage.html

```
<h3 class="subtitle">소개</h3>
<p>그럼 이 홈페이지를 공식적으로 체계적으로 갖추면서, 그동안
공사로 어수선했던 올려드리지 못한 집 안팎 사진들을
상세하게 올리고 당신을 만날 준비를 마치겠습니다.
다.</p>
</article>
</section>
<footer>
<p>제주특별자치도 남제주군 성산읍 수산리 000 번지 요안도라</p>
<address>
예약문의 070-###-####, yoan##@naver.com
</address>
<p>Copyright ©. All rights reserved.</p>
</footer>
</div>
</body>
</html>
```

# Table

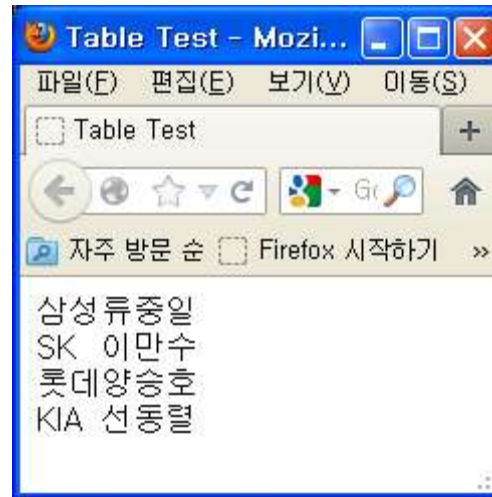
## ❖ Table

### ✓ display 속성

- table: 표를 생성
- inline-table: html에서의 table
- table-row: 행(tr)
- table-row-group: 행(tbody)
- table-header-group: 행(thead)
- table-footer-group: 행(tfoot)
- table-column: 열(col)
- table-column-group: 열(colgroup)
- table-cell: 셀(td)
- table-caption: 제목
- list-item: html의 li
- inherit

# Table

## ❖ Table



# Table

## ❖ tabledisplay.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Table Test</title>
<style type="text/css">
 baseball {
 display: table;
 }

 data {
 display: table-row-group;
 }

 team, director {
 display: table-cell;
 }
</style>
</head>
```

# Table

## ❖ tabledisplay.html

```
<body>
 <baseball> <data> <team>삼성</team> <director>류중일</director>
 </data> <data> <team>SK</team> <director>이만수</director> </data> <data>
 <team>롯데</team> <director>양승호</director> </data> <data> <team>KIA</team>
 <director>선동렬</director> </data> </baseball>
</body>
</html>
```

# Table

## ❖ Table

### ✓ 셀 너비 지정

- table-layout 에 fixed를 설정하면 셀의 너비는 고정되고 auto를 설정하면 셀의 너비가 셀의 내용에 따라 너비가 설정됨

### ✓ 캡션

- 테이블의 캡션은 caption-side 속성으로 지정하며 top과 bottom 값을 지정 가능

### ✓ 테두리(border)

- 표와 셀의 테두리가 구분되는 separated borders model과 한 묶음으로 합쳐진 듯한 collapsing borders model
- separated borders model의 테두리 타입과 collapsing borders model 타입은 border-collapse 값에 따라 결정
- border-collapse의 값은 collapse와 separate가 지정 가능
- 기본 값은 separate

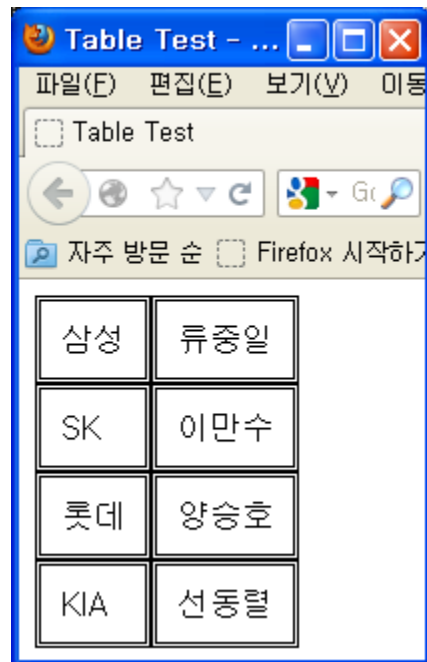
### ✓ 선 간격

- separated border model에서 지정하며 border-spacing으로 설정



# Table

## ❖ Table



# Table

## ❖ tableborder.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Table Test</title>
<style type="text/css">
baseball {
 display: table;
 border-collapse: separate;
}

data {
 display: table-row-group;
}

team, director {
 display: table-cell;
 border: 3px double black;
 padding: 10px;
}
</style>
</head>
```

# Table

## ❖ tableborder.html

```
<body>
 <baseball> <data> <team>삼성</team> <director>류중일</director>
 </data> <data> <team>SK</team> <director>이만수</director> </data> <data>
 <team>롯데</team> <director>양승호</director> </data> <data> <team>KIA</team>
 <director>선동렬</director> </data> </baseball>
</body>
</html>
```

# Table

## ❖ Table

### ✓ 높이와 너비 및 정렬

- 테이블의 높이와 너비는 height와 width를 이용해서 적용
- 상하 정렬은 vertical-align으로 baseline, top, middle, bottom 지정 가능
- 좌우 정렬은 text-align으로 지정하며 left, right, center, justify 지정 가능

### ✓ border-spacing: 인접한 셀 테두리 사이의 거리

### ✓ 빈 셀 표시 여부

- empty-cells 프로퍼티를 이용하면 되는데 이 속성의 값이 show이면 빈 셀을 표시하고 hide이면 보이지 않게 됨

# 사용자 인터페이스

## ❖ 사용자 인터페이스

### ✓ cursor

- 마우스 커서 모양을 변경할 수 있는 속성
- url(파일위치), auto, default, pointer, crosshair, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help, progress 등을 이용해서 설정하는 것이 가능

### ✓ outline

- 외곽선의 모양
- border와 동일한 형태로 지정
- color에 invert 값을 지정할 수 있음

# 사용자 인터페이스

## ❖ 사용자 인터페이스

### Crossbrowsing

표준을 위해 크로스 브라우징할 때에 1픽셀의 오차도 없게 디자인하는 것이 크로스브라우징이 아니다. 정보 격차가 심하게 일어나거나 정보 누락이 생기지 않는다면 그 한도 내에서의 약한 오차는 넘어갈 수 있다고 본다.

# 사용자 인터페이스

## ❖ 사용자 인터페이스

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>박스 크기 조절</title>
<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 body {
 margin: 20px;
 }
```

# 사용자 인터페이스

## ❖ 사용자 인터페이스

```
.content {
 font: 12px "굴림", Gulim;
 margin-top: 10px;
 width: 300px;
 padding: 10px 40px;
 background: #fdd;
 border: 2px solid #123456;
 outline: 2px solid #fa0;
 outline-offset: 2px;
 resize: both;
 overflow: auto;
}
</style>
</head>
```



# 사용자 인터페이스

## ❖ 사용자 인터페이스

```
<body>
```

```
 <h1>Crossbrowsing</h1>
```

```
 <p class="content">표준을 위해 크로스 브라우징할 때에 1픽셀의 오차도 없게 디자인하는
 것이
```

```
 크로스브라우징이 아니다. 정보 격차가 심하게 일어나거나 정보 누락이 생기지 않는다면
 그 한도 내에서의 약한 오차는 넘어갈 수
 있다고 본다.</p>
```

```
</body>
```

```
</html>
```

# 변환 과 애니메이션

## ❖ Transform(변환)

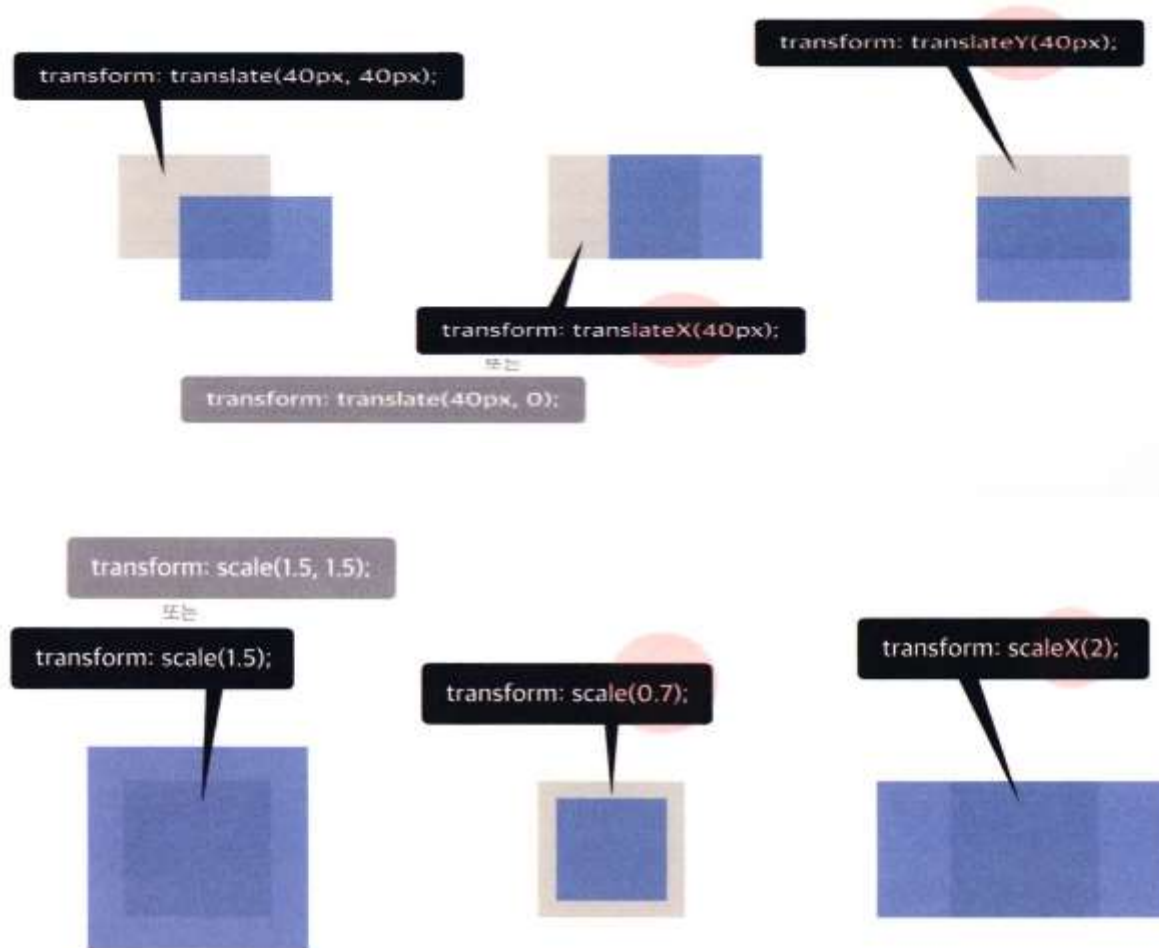
### ✓ 2D 변환

- 이동
  - ◆ translate(x축 이동값, y축 이동값)
  - ◆ translateX(x축 이동값)
  - ◆ translateY(y축 이동값)
- 크기 변환
  - ◆ scale(가로 크기, 세로 크기)
  - ◆ scaleX(가로 크기)
  - ◆ scaleY(세로 크기)
- 회전
  - ◆ rotate(각도deg)
- 비틀기(왜곡)
  - ◆ skew(x축 값, y축 값)
  - ◆ skewX(X축 값)
  - ◆ skewY(Y축 값)
- 한꺼번에 적용 - matrix (scaleX, skewX, skewY, scaleY, translateX, translateY)
- 변환 기준점 변경 - transform-origin

# 변환 과 애니메이션

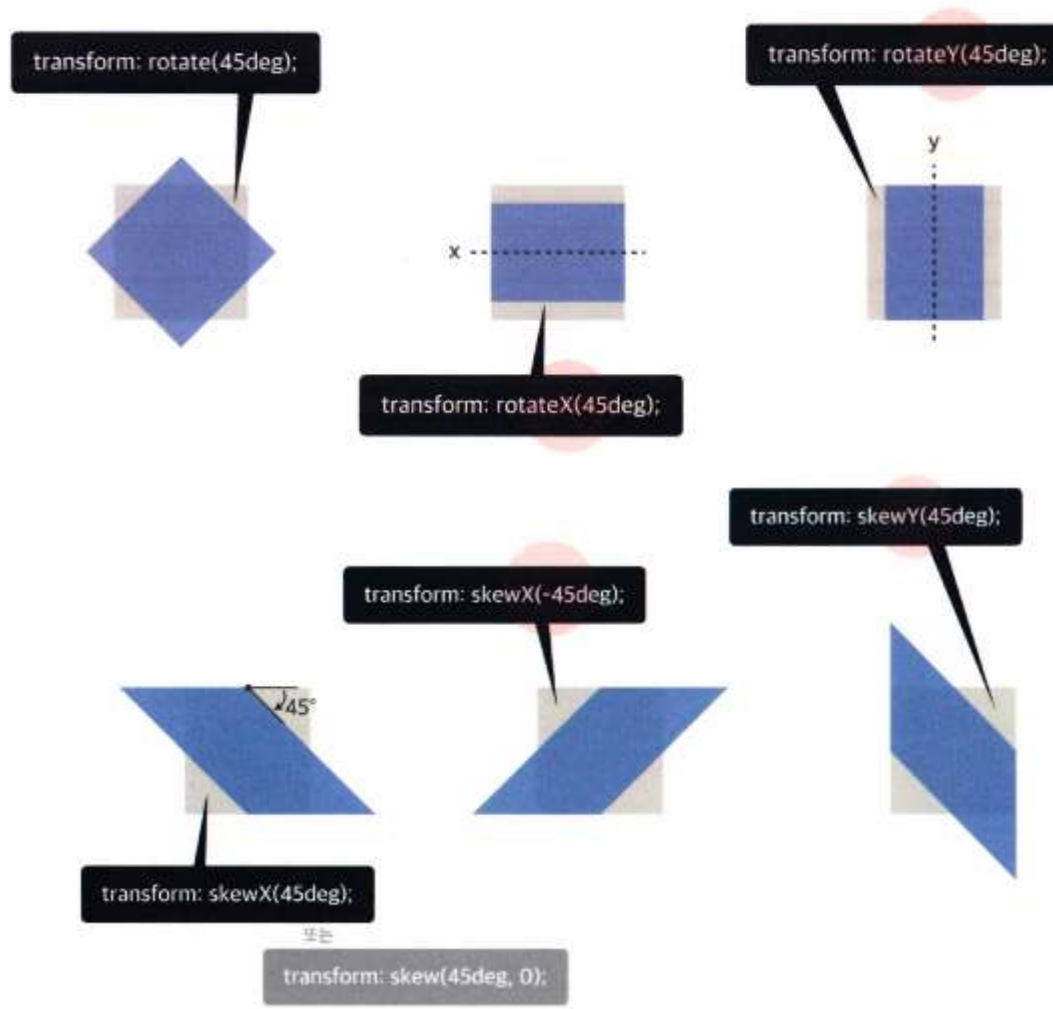
## ❖ Transform(변환)

### ✓ 2D 변환



# 변환 과 애니메이션

- ❖ Transform(변환)
  - ✓ 2D 변환



# 변환 과 애니메이션

## ❖ Transform(변환)

### ✓ 3D 변환

#### ● 이동

- ◆ translate(z 축 이동값)
- ◆ translate3d(x, y, z)

#### ● 크기 변환

- ◆ scaleZ(z)
- ◆ scale3d(x, y, z)

#### ● 회전

- ◆ rotateX(각도deg)
- ◆ rotateY(각도deg)
- ◆ rotateZ(각도deg)
- ◆ rotate3d(x, y, z, a)

- perspective: 원래 있던 위치에서 사용자가 있는 방향이나 반대방향으로 잡아당기거나 밀어내어 원근감을 갖도록 하는 설정
- 한꺼번에 적용 – matrix3d(n x 16)

# 변환 과 애니메이션

## ❖ Transform(변환)

### ✓ 3D 변환

#### ● 이동

- ◆ translate(z 축 이동값)
- ◆ translate3d(x, y, z)

#### ● 크기 변환

- ◆ scaleZ(z)
- ◆ scale3d(x, y, z)

#### ● 회전

- ◆ rotateX(각도deg)
- ◆ rotateY(각도deg)
- ◆ rotateZ(각도deg)
- ◆ rotate3d(x, y, z, a)

- perspective: 원래 있던 위치에서 사용자가 있는 방향이나 반대방향으로 잡아당기거나 밀어내어 원근감을 갖도록 하는 설정
- 한꺼번에 적용 – matrix3d(n x 16)

# 변환 과 애니메이션

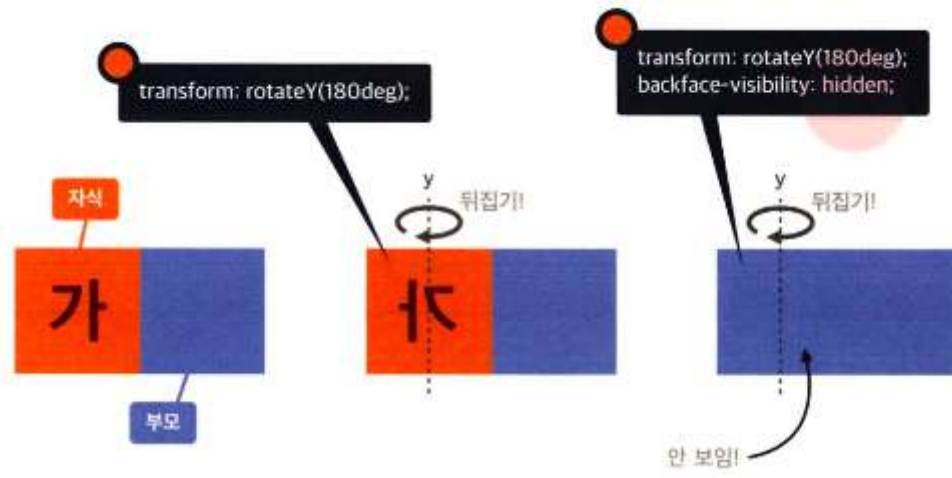
## ❖ Transform(변환)

### ✓ transform-style

- flat: 자손의 3차원 속성을 무시
- preserve-3d: 자손의 3차원 속성을 유지

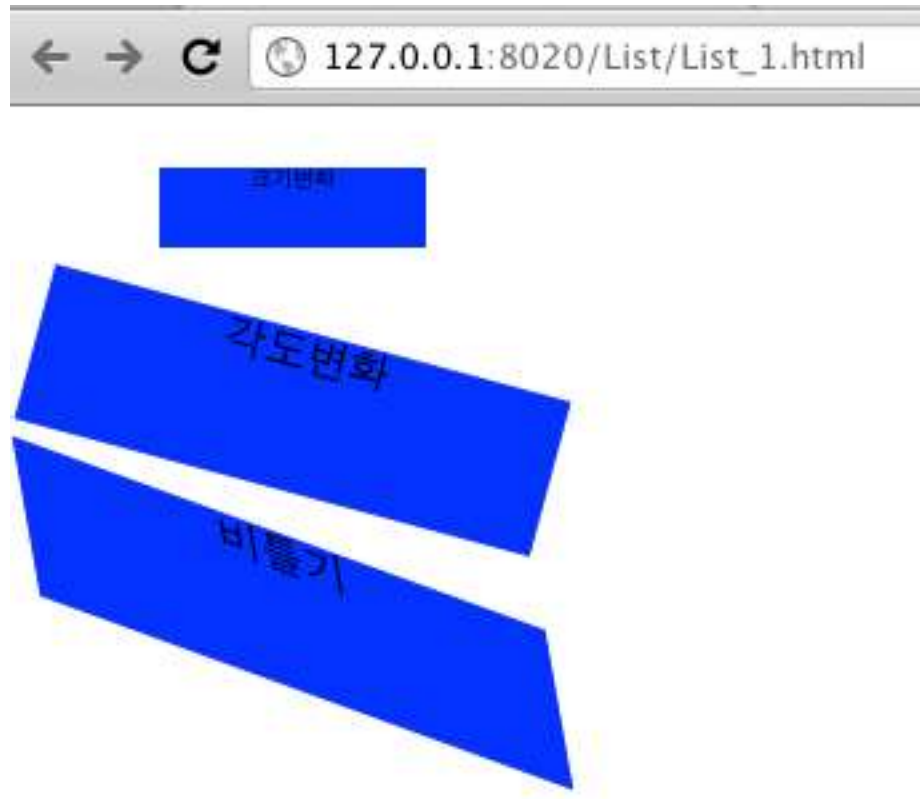
### ✓ backface-visibility

- visible: 뒷면을 출력
- hidden: 뒷면을 보이지 않게



# 변환 과 애니메이션

## ❖ Transform(변환)





# 변환 과 애니메이션

## ❖ transform2d.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Transform</title>
<style type="text/css">
 div {
 background-color: blue;
 width: 200px;
 height: 60px;
 text-align: center;
 }
 #div1 {
 transform: scale(0.5);
 }
 #div2 {
 transform: rotate(15deg);
 }
 #div3 {
 transform: skew(10deg, 20deg);
 }
</style>
</head>
```

# 변환 과 애니메이션

## ❖ transform2d.html

```
<body>
 <div id="div1">크기변화</div>
 <p></p>
 <div id="div2">각도변화</div>
 <p></p>
 <div id="div3">비틀기</div>
</body>
</html>
```

# 변환 과 애니메이션

## ❖ Transition Animation

- ✓ Transition: 하나의 스타일에서 다른 스타일로 부드럽게 변경해주는 기능
- ✓ 중간에 프레임을 채워서 상태가 갑자기 변하지 않고 시간이 지남에 따라 부드럽게 변하게 해 줌
- ✓ 애니메이션 설정
  - transition-property; => 적용되는 속성으로 all
  - transition-duration; => 적용 시간
  - transition-timing-function; => 인터폴레이션 효과
    - ◆ ease: 기본형
    - ◆ linear: 선형, 처음부터 끝까지 같은 속도로 움직임
    - ◆ ease-in: 시작 속도가 상대적으로 천천히 움직임
    - ◆ ease-out: 끝 속도가 상대적으로 천천히 움직임
    - ◆ ease-in-out: 시작과 끝이 상대적으로 천천히 움직임
  - transition-timing-delay; => 지연 시간
  - transition:속성 시간 인터폴레이션 지연시간; 을 순서대로 설정해도 가능

# 변환 과 애니메이션

## ❖ Transition Animation

### ✓ 사용 가능 속성

- top, left, bottom, right: 위치 속성
- width, height: 크기 속성
- margin, padding: 박스 속성
- border-width, border-radius, border-color: 테두리 속성
- color, background-color: 색상 속성
- opacity
- transform

# 변환 과 애니메이션

❖ Transition Animation

## Moving Mainmenu

COMPANY

PRODUCT

SERVICE

CUSTOMER SUPPORT

COMMUNITY

# 변환 과 애니메이션

## ❖ movingmenu.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Transition</title>
<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 body {
 font: 20px "Arial", Gulim;
 margin: 20px;
 }

 h1, ul {
 padding: 10px;
 }
```

# 변환 과 애니메이션

## ❖ movingmenu.html

```
.menu li a {
 display: block;
 width: 300px;
 height: 20px;
 padding: 10px;
 margin-bottom: 2px;
 background: #abc;
 text-indent: 10px;
 transition: width 1.2s;
 color: #123456;
 text-decoration: none;
}

.menu li a:hover {
 width: 450px;
 letter-spacing: 3px;
 color: #c00;
}
</style>
</head>
```

# 변환 과 애니메이션

## ❖ movingmenu.html

```
<body>
 <h1>Moving Mainmenu</h1>
 <ul class="menu">
 COMPANY
 PRODUCT
 SERVICE
 CUSTOMER SUPPORT
 COMMUNITY

</body>
</html>
```



# 변환 과 애니메이션

## ❖ Transition Animation



# 변환 과 애니메이션

## ❖ buttonmenu.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Transition Menu</title>
<style>
 body {
 font-family: Tahoma, Verdana, Geneva, sans-serif;
 margin: 6em;
 font-weight: bold;
 }
 ul {
 list-style-type:none;
 margin: 0;
 padding: 0;
 }
 li {
 float: left;
 }
}
```

# 변환 과 애니메이션

## ❖ buttonmenu.html

```
a {
 display: block;
 width: 6em;
 margin: 0 5px;
 padding: .5em 1em;
 text-decoration: none;
 text-align: center;
 color: #fff;
 background-color: rgb(190, 190, 190);
 border: 2px solid rgb(175,175,175);
 border-radius: 6px;
 text-shadow: #666 .1em .1em .1em;
 box-shadow: 0 5px 3px rgba(0,0,0,.5);
 position: relative;
 transition: background-color 0.2s, border-color 0.2s, top .2s, box-shadow 0.2s;
}
```

# 변환 과 애니메이션

## ❖ buttonmenu.html

```
a:hover, a:focus {
 background-color: #fdca00;
 border-color: #fda700;
}
a:active {
 top: 3px;
 box-shadow: 0 1px 2px rgba(0,0,0,.5);
}
</style>
</head>
```

# 변환 과 애니메이션

## ❖ buttonmenu.html

```
<body>

 야구
 축구
 농구
 배구
 일반

</body>
</html>
```

# 변환 과 애니메이션

## ❖ Transition Animation




# 변환 과 애니메이션

## ❖ imagetransition.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Image Transform</title>
<style>
 body {
 font-family: Tahoma, Verdana, Geneva, sans-serif;
 margin: 6em 6em;
 font-weight: bold;
 }
 ul {
 list-style-type: none;
 padding: 0;
 margin: 0;
 }
 li {
 float: left;
 margin-right: 10px;
 }
 li:last-of-type {
 margin-right: 0;
 }

```



# 변환 과 애니메이션

## ❖ imagetransition.html

```
a {
 display: block;
}
```

```
img {
 width: 200px;
 height: 150px;
 box-shadow: 2px 2px 2px rgba(0,0,0,.4);
 transition: transform .3s ease-in-out;
}
```



# 변환 과 애니메이션

## ❖ imagetransition.html

```
a:hover img {
 box-shadow: 6px 6px 6px rgba(0,0,0,.3);
}
```

```
a:hover #img1, a:focus #img1 {
 transform: scale(1.5) rotate(-3deg);
}
```

```
a:hover #img2, a:focus #img2 {
 transform: scale(1.5) rotate(5deg);
}
```

```
a:hover #img3, a:focus #img3 {
 transform: scale(1.5) rotate(-7deg);
}
```

```
a:hover #img4, a:focus #img4 {
 transform: scale(1.5) rotate(2deg);
}
```

```
</style>
</head>
```

# 변환 과 애니메이션

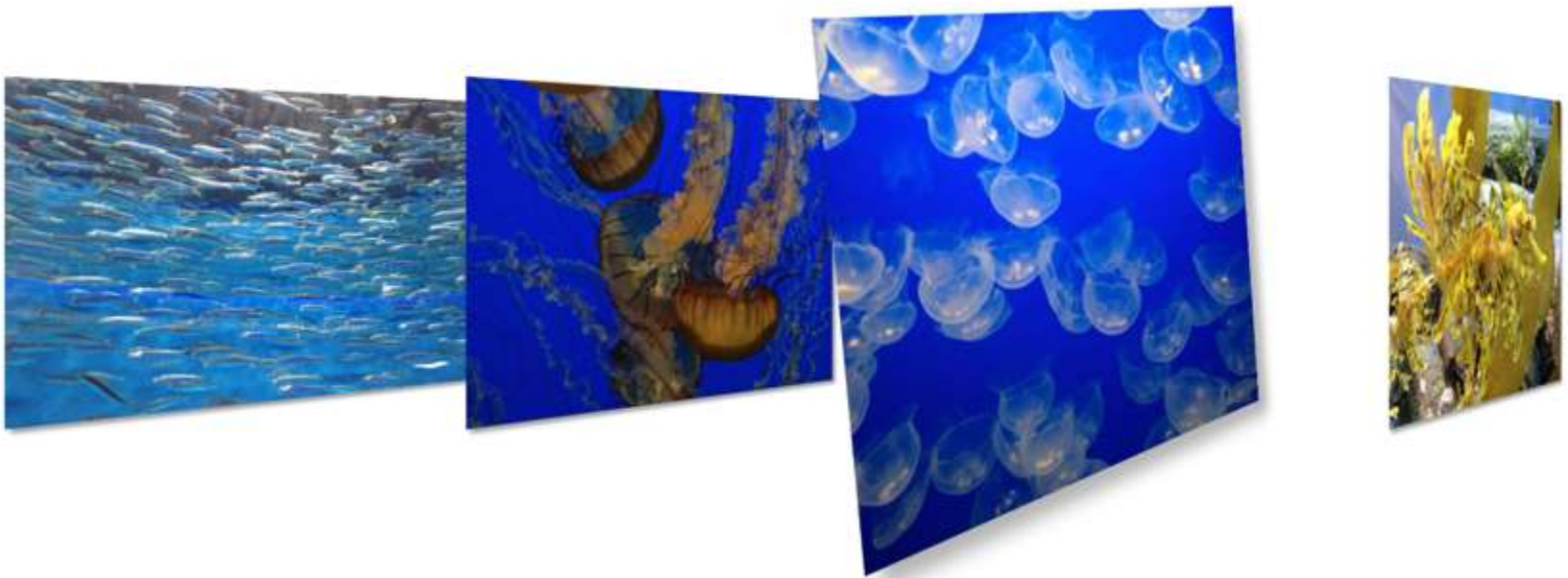
## ❖ imagetransition.html

```
<body>

</body>
</html>
```

# 변환 과 애니메이션

## ❖ Transition Animation



# 변환 과 애니메이션

## ❖ imagetransform.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Image Transform</title>
<style>
 body {
 font-family: Tahoma, Verdana, Geneva, sans-serif;
 margin: 6em 6em;
 font-weight: bold;
 }
 ul {
 list-style-type: none;
 padding: 0;
 margin: 0;

 width:1000px;
 height:100px;
 perspective:600;
 }
```

# 변환 과 애니메이션

## ❖ imagetransform.html

```
li {
 float: left;
 margin-right: 10px;

 transform: rotateY(45deg);
}
li:last-of-type {
 margin-right: 0;
}

a {
 display: block;
}

img {
 width: 200px;
 height: 150px;
 box-shadow: 2px 2px 2px rgba(0,0,0,.4);
 transition: -webkit-transform .3s ease-in-out;
}
```

# 변환 과 애니메이션

## ❖ imagetransform.html

```
a:hover img {
 box-shadow: 6px 6px 6px rgba(0,0,0,.3);
}

a:hover #img1, a:focus #img1 {
 transform: scale(1.5) rotate(-3deg);
}
a:hover #img2, a:focus #img2 {
 transform: scale(1.5) rotate(5deg);
}
a:hover #img3, a:focus #img3 {
 transform: scale(1.5) rotate(-7deg);
}
a:hover #img4, a:focus #img4 {
 transform: scale(1.5) rotate(2deg);
}
</style>
</head>
```

# 변환 과 애니메이션

## ❖ imagetransform.html

```
<body>
```

```

```

```

```

```

```

```

```

```

```

```

```

```
</body>
```

```
</html>
```

# 변환 과 애니메이션

## ❖ Keyframe Animation

✓ 프레임들의 상태를 지정해서 동작을 좀 더 세밀하게 제어할 수 있도록 하는 애니메이션

✓ 애니메이션 만들기

@-webkit-keyframes 애니메이션이름

{

from {시작값 }

to {종료값}

}

from 과 to 대신에 백분율 이용해서 설정도 가능

✓ 애니메이션 적용

- -webkit-animation-name: 애니메이션 이름

- -webkit-animation-timing-function: 인터폴레이션 효과

- -webkit-animation-iteration-count: 반복횟수(infinite이면 무한대)

- -webkit-animation-duration: 시간

- -webkit-animation-play-state: 애니메이션 상태

- -webkit-animation-direction: 애니메이션 반복 형태(alternate, normal)

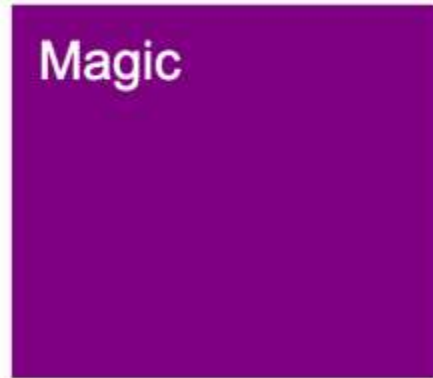


# 변환 과 애니메이션

## ❖ Keyframe Animation

### Animation

Magic



# 변환 과 애니메이션

## ❖ keyframe\_color.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Keyframe Animation</title>
<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }
 body {
 font: 12px "Arial", Gulim;
 margin: 20px;
 }
 li {
 list-style-type: none;
 }

```

# 변환 과 애니메이션

## ❖ keyframe\_color.html

```
p {
 width: 140px;
 height: 120px;
 padding: 10px;
 background: #c00;
 color: #fff;
 font-size: 20px;
 animation: keyanimatio 5s infinite;
}
@-webkit-keyframes keyanimatio{
 0% { background:red; }
 25% { background:orange; }
 40% { background:yellow; }
 60% { background:green; }
 80% { background:blue; }
 100% { background:purple;}
}
</style>
</head>
```

# 변환 과 애니메이션

## ❖ keyframe\_color.html

```
<body>
 <h1>Animation</h1>
 <p>
 Magic
 </p>
</body>
</html>
```



# 변환 과 애니메이션

## ❖ Keyframe Animation



# 변환 과 애니메이션

## ❖ keyframe\_3d.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>3D 애니메이션 Test</title>
<style type="text/css">
 @-webkit-keyframes spinner {
 from { -webkit-transform: rotateY(0deg);}
 to {-webkit-transform: rotateY(-360deg);
 }

}
#stage {
 margin: 1em auto;
 -webkit-perspective: 1200px;
}
```

# 변환 과 애니메이션

## ❖ keyframe\_3d.html

```
#spinner {
 -webkit-animation-name: spinner;
 -webkit-animation-timing-function: linear;
 -webkit-animation-iteration-count: infinite;
 -webkit-animation-duration: 6s;
 -webkit-transform-style: preserve-3d;
}

spinner:hover {
 -webkit-animation-play-state: paused;
}
</style>
</head>
```

# 변환 과 애니메이션

## ❖ keyframe\_3d.html

```
<body>
 <div id="stage" style="background: rgba(0, 0, 0, 0.5);">
 <p id="spinner"
 style="background: rgba(0, 0, 0, 0.5); text-align: center; color: #fff;">
 3D 애니메이션</p>
 </div>
 </body>
</html>
```



# 기타

## ❖ CSS Reset

- ✓ 모든 사용자 에이전트 스타일 규칙을 무효로 하고 CSS 스타일을 초기화된 상태로 시작할 수 있게 해주는 스타일 규칙의 컬렉션
- ✓ 야후의 리셋 관련 링크

<link rel="stylesheet" type="text/css"

href="<http://yui.yahooapis.com/3.5.1/build/cssreset/cssreset-min.css>">

# 기타

## ❖ Image Sprite

- ✓ 이미지를 나누어서 사용하는 기술
- ✓ 이미지 파일 여러 개를 사용하는 것보다는 하나의 파일을 잘라서 사용하는 것이 입출력 효율 면에서 유리
- ✓ 이미지 스프라이트 사용 예

<https://www.smashingmagazine.com/2009/04/the-mystery-of-css-sprites-techniques-tools-and-tutorials/>



# 기타

## ❖ Image Sprite

### ✓ 사용법

#### ● 마크업

```

 Twitter
 Facebook

```

#### ● 스타일

```
.hide{
 text-indent:100%;
 white-space:nowrap;
 overflow:hidden;
}
li a{
 display:block;
 width:29px;
 height:18px;
 background-image:url(social.png);
}
li a.twitter{background-position:0 0;}
li a.facebook{background-position:0 -20;}
```

# 기타

## ❖ Hack

- ✓ 크로스 브라우징 환경에서 CSS 지원여부 또는 해석의 차이로 인해 발생하는 버그를 바로잡기 위한 트릭을 핵(hack) 또는 필터(filter) 라고 함

```
<!--[if IE 6]>
```

```
<link rel="stylesheet" type="text/css" ref="css/ie6.css">
```

```
<![endif]-->
```

# 기타

❖ formstyle.html

## 회원가입

email

admin

중복확인

비밀번호

....

하고싶은 이야기

300자 이내로 작성

학력

중졸

▼

성별

☐ 남자

☐ 여자

메시지 수신설정

☐ 메일 수신 허용

☐ SMS 수신 허용

회원가입

작성취소

# 기타

## ❖ formstyle.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>폼 스타일</title>
<style>
 ul {
 list-style-type: none;
 }

 ul li {
 clear: both;
 }

 form {
 width: 50em;
 border: 1px solid #666;
 border-radius: 10px;
 box-shadow: .2em .2em .5em #999;
 background-color: #d0e9f6;
 padding: 1em;
 overflow: hidden;
 }
```

# 기타

## ❖ formstyle.html

```
label {
 display: block;
 float: left;
 width: 10em;
 text-align: right;
 margin-right: .5em;
 color: #04699d;
}

input.textinput {
 width: 30em;
 height: 2em;
 border: 1px solid #666;
 margin-bottom: 1em;
}
```



# 기타

## ❖ formstyle.html

```
textarea {
 display: block;
 width:30em;
 height:5em;
 border:1px solid #666;
 margin-bottom:1em;
 line-height:1.25;
 overflow:auto;
 resize:none;
}
input.textinput, textarea{
 font-family:Georgia, "Times New Roman", Times, serif;
 font-size:.875em;s
}

fieldset{
 margin:0;
 padding:0;
 border:none;
}
```



# 기타

## ❖ formstyle.html

```
legend{
 display:block;
 width:10em;
 float:left;
 margin-right:.5em;
 text-align:right;
 color:#04699d;
}

#genders label, #message label{
 color:#000;
 display:inline;
 float:none;
 text-align:inherit;
 width:auto;
 font-weight:normal;
 background-color:inherit;
}

#genders ul li{
 display:inline;
 margin-bottom:0;
}
```

# 기타

## ❖ formstyle.html

```
#message ul{
 margin-left:.11em;
}
```

```
#message ul li{
 display:inline;
 clear:none;
}
```

```
#genders, #message{
 margin-top:1em;
}
```



# 기타

## ❖ formstyle.html

```
input[type="submit"], input[type="reset"]{
 display:block;
 width:10em;
 height:2em;
 float:left;
 background:white;
 font-size:inherit;
 border: 1px solid #04699d;
 border-radius:4px;
 box-shadow: 2px 2px 3px rgba(0,0,0,.5);
 margin-top:1em;
}
input[type="button"]{
 display:block;
 width:10em;
 height:2em;
 float:right;
 background:white;
 font-size:inherit;
 border: 1px solid #04699d;
 border-radius:4px;
 box-shadow: 2px 2px 3px rgba(0,0,0,.5);
}
```

# 기타

## ❖ formstyle.html

```
input[type="submit"]{
 margin-left:10.5em;
 margin-right: 1em;
 color:#C00;
}
</style>
</head>
```



# 기타

## ❖ formstyle.html

```
<body>
 <form method="post" action="server.jsp">
 <h2>회원가입</h2>

 <label for="email">email</label> <input type="text"
 id="email" name="email" class="textinput" /> <input
type="button"
 alt="중복확인" value="중복확인" />

 <label for="userPw">비밀번호</label> <input type="password"
 id="userpw" name="userpw" class="textinput" />
 <label for="yourstory">하고싶은 이야기</label> <textarea rows="3"
 cols="30" placeholder="300자 이내로 작성" id="story"
name="story"> </textarea>

 <label for="education">학력</label> <select name="education">
 <option value="1">중졸</option>
 <option value="2">고졸</option>
 <option value="3">대졸</option>
 <option value="4">대학원졸 이상</option>
 <option value="5">대학원졸 이상</option>
 </select>
```

# 기타

## ❖ formstyle.html

```

 <fieldset id="genders">
 <legend>성별</legend>

 <label for="man"><input type="radio"
name="gender" id="man"
 value="male" /> 남자
</label> <label> <input type="radio" name="gender" id="woman"
 value="female" /> 여자</label>

 </fieldset>

 <fieldset id="message">
 <legend>메시지 수신설정</legend>

 <label><input type="checkbox"
name="accept"
 id="acceptEmail" />
메일 수신 허용</label>
```

# 기타

## ❖ formstyle.html

```
name="accept"
/>SMS 수신 허용</label>

</fieldset>

<li class="buttons"><input type="submit" value="회원가입" /> <input
type="reset" value="작성취소" />

</form>
</body>
</html>
```

<li> <label> <input type="checkbox" id="acceptSMS"

# 레이아웃

## ❖ 레이아웃 종류

- ✓ 고정형 레이아웃: 브라우저 창이나 텍스트 크기와 상관없이 특정 픽셀 폭에 고정
- ✓ 유동형 레이아웃: 브라우저 창의 크기에 비례해 크기가 고정
- ✓ 탄력적 레이아웃: 텍스트 크기에 비례해 크기가 고정
- ✓ 하이브리드 레이아웃: 고정된 영역과 비율을 조정할 수 있는 영역을 함께 사용





# 레이아웃

## ❖ 고정형 레이아웃

- ✓ 인쇄물처럼 디자이너가 페이지 요소, 맞춤, 줄 길이 등을 제어
- ✓ 특정 크기의 디스플레이를 타겟으로 하는 경우 사용
- ✓ 일반적으로 오른쪽에 남은 공간을 두고 왼쪽 모서리에서 시작
- ✓ 장점
  - 어떤 레이아웃이 될 지 예상할 수 있고 글줄 길이를 제어할 수 있음
  - 디자인과 제작이 쉬움
- ✓ 단점
  - 브라우저 창이 페이지보다 작으면 오른쪽 모서리에 있는 콘텐츠가 보이지 않음
  - 화면이 큰 스크린에서는 남은 공간이 이상해 보임
  - 텍스트의 크기가 매우 크면 글줄 길이가 이상하게 짧아질 수 있음
- ✓ 만드는 방법
  - 전체 페이지의 콘텐츠를 div 요소에 넣고 특정한 픽셀로 설정하고 class 나 id를 content, container, wrapper, page 등으로 부여

# 레이아웃

## ❖ fixedlayout.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>고정 폭 2단 레이아웃</title>
<style>
 #wrapper{
 width:960px;

 }
 #main{
 float:left;
 width:650px;
 margin:0 20px;
 background-color:blue;
 }
 #extras{
 float:left;
 width:250px;
 margin:0 20px 0 0;
 background-color:red;
 }
}
```

# 레이아웃

## ❖ fixedlayout.html

```
#header{
 background-color:gray;
}
#footer{
 clear:left;
 background-color:gray;
}
</style>
</head>
```



# 레이아웃

## ❖ fixedlayout.html

```
<body>
 <div id="wrapper">
 <div id="header">Headline</div>
 <div id="main">Main Article</div>
 <div id="extras">Linkes And News</div>
 <div id="footer">Copylight Park</div>
 </div>
</body>
</html>
```

# 레이아웃

## ❖ 유동형 레이아웃

- ✓ 어떠한 기기에서도 동일한 레이아웃을 가지면서 기기 특성에 맞게 웹 문서가 표현될 수 있으면 문서 안의 각 요소의 너비를 픽셀 값과 같은 고정 폭이 아니라 백분율과 같은 가변 폭으로 지정하면 되는데 이렇게 사이트 레이아웃을 백분율로 지정하는 것이 유동형 레이아웃
- ✓ 유동형 레이아웃을 사용할 때는 콘텐츠가 여러 단으로 구성되어 있을 경우 너비 값이 줄어들면 실제 콘텐츠를 확인하기 불편하므로 되도록 간결한 디자인을 사용하는 것이 좋습니다.
- ✓ 장점
  - 텍스트가 창을 채우므로 남는 공간을 없앨 수 있음
  - 사용자가 창과 콘텐츠의 폭을 제어할 수 있음
  - 수평 스크롤 막대를 없앨 수 있음
- ✓ 단점
  - 큰 모니터에서는 글줄 길이가 길어져서 읽기가 어려울 수 있음
  - 예측이 어려움
  - 여백을 만들기 어려움
  - 크기를 계산하기 어려움
- ✓ 만드는 방법
  - px 단위로 되어 있는 것은 백분율로 변경하고 글자 크기는 em 단위로 변경
    - ◆  $em = \text{글자크기(px)} / 16px$
  - width는 생략

# 레이아웃

## ❖ 유동형 레이아웃



# 레이아웃

## ❖ 유동형 레이아웃

헤더(header) 960px

아티클  
(article)

아티클  
(article)

아티클  
(article)

사이드바(aside) 900px

푸터/footer) 960px

# 레이아웃

## ❖ liquidlayout.html

```
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width">
<title>스마트폰 </title>
<style>
 #wrapper {
 width: 96%;
 margin: 0 auto;
 }

 header {
 background-color: #069;
 padding: 10px;
 overflow: hidden;
 }

 section {
 border: 5px solid #333;
 overflow: hidden;
 }
```



# 레이아웃

## ❖ liquidlayout.html

```
footer {
 padding: 10px;
 background-color: #333;
 color: white;
 text-align: center;
}
```

```
header h1 {
 width: 40%;
 float: left;
 font-size: 2.5em;
 font-family: "돋움", "굴림";
 color: #fff;
}
```

```
header nav {
 width: 60%;
 float: right;
}
```

# 레이아웃

## ❖ liquidlayout.html

```
nav ul {
 list-style-type: none;
 margin-top: 20px;
}
```

```
nav ul li {
 display: inline;
 float: left;
 margin-top: 10px;
 margin-right: 20px;
}
```

```
nav ul li a, nav ul li a:visited {
 font-family: "돋움", "굴림";
 font-size: 1em;
 color: white;
 text-decoration: none;
}
```

# 레이아웃

## ❖ liquidlayout.html

```
article {
 width: 29.17%;
 height: auto;
 float: left;
 margin-right: 1.5625%;
 margin-bottom: 20px;
 margin-top: 20px;
 padding: 0.5%;
 border: 1px dotted #333;
 border-radius: 10px;
}
```

```
article:first-child {
 margin-left: 2.083%;
}
```

```
article h3 {
 background: url(note.png) no-repeat left center;
 padding-left: 30px;
}
```

# 레이아웃

## ❖ liquidlayout.html

```
dl {
 margin: 10px;
}

dt {
 font-weight: bold;
}

aside {
 clear: both;
 width: 93.75%;
 margin: 10px auto;
 padding: 10px;
 background-color: #99cc33;
}
</style>
</head>
```

# 레이아웃

## ❖ liquidlayout.html

```
<body>
 <header>
 <h1>스마트폰 운영체제 </h1>
 <nav>

 아이폰
 안드로이드
 기타

 </nav>
 </header>
```

# 레이아웃

## ❖ liquidlayout.html

<section>

<article>

<h3>아이폰 </h3>

<dl>

<dt>아이폰5</dt>

<dd>세계 최초로 ITO 터치 센서 전극을 디스플레이 내부에 증착한  
인셀방식의 디스플레이를 적용</dd>

<dt>아이폰6</dt>

<dd>아이폰 6는 기존 기종인 아이폰5s의 두께인 7.6mm에서  
6.9mm로 0.7mm 더욱 얇아졌다.카메라는

시네마틱 손떨림 방지 기능이 추가되었다.</dd>

<dt>아이폰6 플러스</dt>

<dd>애플 아이폰 6 플러스(Apple iPhone 6 plus)는 페가수스와 폭  
스콘이 제조하고, 애플이

판매하는 iOS 패블릿폰으로, 2014년 9월 9일 아이폰 6  
와 함께 미국 캘리포니아 샌프란시스코 모스코니 센터에서  
발표했다. 크기가 5.5인치로 커진 것이 특징이다.</dd>

</dl>

</article>

# 레이아웃

❖ liquidlayout.html

```
<article>
 <h3>안드로이드</h3>
 <dl>
 <dt>타이젠</dt>
 <dd>삼성 갤럭시 S5(Samsung GALAXY S5)는 삼성전자에서 제조,
판매하는 안드로이드 스마트폰으로,
 삼성 갤럭시 S4의 후속 제품이다.</dd>
 <dt>G3</dt>
 <dd>LG 전자의 최신 스마트폰</dd>
 <dt>샤오미</dt>
 <dd>샤오미 폰(Xiaomi Phone, 중국어: 小米手機)은 중국의 커스텀
롬 개발팀인 MIUI가 설계하고
 샤오미 테크가 제작한 안드로이드 기반의 MIUI 레퍼런
스 스마트폰이다.</dd>
 </dl>
</article>
```

# 레이아웃

❖ liquidlayout.html

```
<article>
 <h3>Web 기반 스마트폰</h3>
 <dl>
 <dt>타이젠</dt>
 <dd>타이젠(Tizen)은 휴대 전화를 비롯한 휴대용 장치를 주로 하며,
TV, 냉장고와 같은 모든 전자기기에
 포함을 목적으로 하는 오픈 소스 모바일 운영 체제이
 다.</dd>
 <dt>WebOS</dt>
 <dd>웹OS(webOS)는 리눅스 커널에서 구동되는 모바일 운영 체제
 이다. 팜 OS를 계승하는 소프트웨어로서
 팜에서 개발을 하고 휴렛 팩커드가 후에 인수했고, 또
 그 뒤에 LG가 인수했다.</dd>
 <dt>블랙베리</dt>
 <dd>리서치 인 모션(Research In Motion Limited, RIM)이 블랙베
 리 스마트폰을 위해 만든
 소프트웨어 플랫폼이다.</dd>
 </dl>
</article>
```



# 레이아웃

## ❖ liquidlayout.html

```
<aside>
 <h3>좋은 스마트폰 </h3>

 사용하기 편리해야 한다.
 배터리 소모량이 적어야 한다.
 다양한 애플리케이션이 제공되어야 한다.
 화면 크기가 적당해야 한다.

</aside>
</section>
<footer>
 <p>Copyright 2014 choongang</p>
</footer>

</body>
</html>
```

# 레이아웃

## ❖ Grid Layout

- ✓ 브라우저 화면을 다양한 크기의 사각 영역(그리드)으로 나눈 후 브라우저 창의 너비에 따라 그리드의 배치를 다르게 만드는 방법
- ✓ 그리드는 너비 값과 높이 값을 가지고 있고 가로로 나열할 수 도 있고 세로로 나열할 수 있도록 CSS의 미디어 쿼리와 유동형 레이아웃을 함께 사용해서 다양한 모습과 위치를 지정
- ✓ 사각 영역은 시맨틱 태그를 사용하거나 서로 다른 <div>를 태그에 넣어두고 id를 지정해서 설정
- ✓ 시맨틱 태그를 이용하는 경우에도 동일한 태그가 여러 개 사용된다면 id를 이용해서 구분하는 것을 권장

# 탄력적 레이아웃

- ❖ 크기를 조절할 수 있는 텍스트와 예상 가능한 글줄 길이를 결합한 것
- ❖ 사용자가 텍스트의 크기를 크게 하면 텍스트를 표현한 상자는 비례적으로 늘어남
- ❖ 브라우저의 확대/축소 기능이 탄력적 레이아웃의 일부를 가져온 것
- ❖ 장점
  - ✓ 텍스트 크기를 조정할 수 있고 사용자에게 일관된 레이아웃 경험을 제공
  - ✓ 유동형 레이아웃과 고정형 레이아웃보다 글줄 길이를 좀 더 엄격히 제어
- ❖ 단점
  - ✓ 이미지와 비디오는 텍스트와 레이아웃의 나머지 부분에 따라 자동으로 비율이 조정되지 않음
  - ✓ 가장 큰 텍스트 크기에서는 레이아웃의 폭이 브라우저 창의 폭을 넘어갈 수 있음
  - ✓ 다양한 장치와 브라우저 창의 크기를 처리하는데 유용하지 않음
  - ✓ 복잡함
- ❖ 만드는 방법
  - ✓ body 요소의 텍스트 크기를 16픽셀로 설정하고 page의 폭을 40em으로 설정
- ❖ <http://www.csszengarden.com/063/>

# 기타 레이아웃

## ❖ 하이브리드 레이아웃

- ✓ 픽셀, 퍼센트, em 을 혼합해서 사용하는 레이아웃
- ✓ 고정된 콘텐츠 영역과 비율이 조정되는 콘텐츠 영역을 섞어서 사용

## ❖ Flexible Box Layout

- ✓ 그리드 레이아웃을 기본으로 해서 플렉스 박스를 원하는 위치에 배치하는 형태
- ✓ 플렉스 박스는 여유 공간에 따라 너비나 높이 및 위치를 자유롭게 변경
- ✓ 하면 크기에 따라 레이아웃의 배치나 크기를 조절해야 할 때 편리하게 사용



# 반응형 웹 디자인

## ❖ 반응형 웹 디자인

- ✓ 초창기에는 모바일 웹 사이트와 PC용 모니터로 보는 웹 사이트를 별도로 개발 – 지금도 그렇게 개발하는 곳이 있음
- ✓ 기존 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 개발하는 것
- ✓ 화면 요소들을 화면 크기에 맞게 재배치하고 요소의 표시 방법을 변경해서 구현
- ✓ <http://colly.com>

## ❖ 장점

- ✓ 모든 스마트 기기에서 콘텐츠 사용 가능
- ✓ 가로 모드에 맞추어 레이아웃 변경 가능
- ✓ 사이트 유지 및 관리가 용이 – 서버 쪽 코드 없이 구현 가능

# Media Query

## ❖ 미디어 쿼리

- ✓ 미디어 별로 스타일 시트를 다르게 적용
- ✓ 사이트에 접속하는 장치에 따라 특정한 CSS 모듈을 사용하도록 해주는 기능
- ✓ 사용방법

[ONLY | NOT] @media 미디어 유형 [AND 조건] \* [AND 조건]

## ✓ 미디어 유형

- ☐ screen : 컴퓨터 디스플레이
- ☐ aural: 음성 합성 장치
- ☐ braille: 점자 표시 장치
- ☐ embossed: 점자 프린터
- ☐ tty: 컴퓨터 터미널
- ☐ tv
- ☐ projection
- ☐ handheld: PDA
- ☐ print: 프린터
- ☐ all: 모든 기기

# Media Query

- ❖ 미디어 쿼리
  - ✓ 브라우저 창의 가로 너비와 세로 높이
    - ☐ width, height
    - ☐ min-width, min-height
    - ☐ max-width, max-height
  - ✓ 단말기의 가로 너비와 세로 높이
    - ☐ device-width, device-height
    - ☐ device-min-width, device-min-height
    - ☐ device-max-width, device-max-height
  - ✓ 화면 회전 속성
    - ☐ orientation:portrait
    - ☐ orientation:landscape

# Media Query

## ❖ 미디어 쿼리

### ✓ 화면 비율

- ☐ aspect-ratio
- ☐ min-aspect-ratio
- ☐ max-aspect-ratio

### ✓ 단말기 화면 비율

- ☐ device-aspect-ratio
- ☐ device-min-aspect-ratio
- ☐ device-max-aspect-ratio

### ✓ 색상 당 비트 수

- ☐ color
- ☐ min-color
- ☐ max-color



# Media Query

## ❖ 적용 방법

- ✓ 인쇄용 프린터에 적용: `<link href="print.css" rel="stylesheet" type="text/css" media="print">`
- ✓ 스마트폰에 적용: `<link href="smartphone.css" rel="stylesheet" type="text/css" media="screen and (max-device-width:480px) and (min-device-width:320px)">`
- ✓ `@import url(css 파일의 경로) 미디어쿼리 조건 의 형태 가능`
- ✓ `<style media="미디어쿼리 조건">`의 형태 가능

# viewport meta

- ❖ 모바일 웹 페이지는 화면과 관련된 정보를 viewport meta 태그를 이용해서 설정합니다.
- ❖ viewport 메타 태그 속성
  - ✓ width : width=240, width=device-width
  - ✓ height : height=800, height=device-height
  - ✓ initial-scale : initial-scale=2.0
  - ✓ user-scalable : user-scalable
  - ✓ minimum-scale : minimum-scale=1.0
  - ✓ maximum-scale : maximum-scale=2.0
  - ✓ marget-densitydpi : target-densitydpi=medium-dpi
  - ✓ 일부 브라우저에서는 target-densitydpi 속성은 제공하지 않습니다.
  - ✓ 안드로이드: <http://developer.android.com/guide/webapps/targeting.html>
  - ✓ 아이폰:  
[http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/Safari\\_iWebContent/UsingtheViewport/UsingtheViewport.html#//apple\\_ref/doc/uid/TP40006509-SW1](http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/Safari_iWebContent/UsingtheViewport/UsingtheViewport.html#//apple_ref/doc/uid/TP40006509-SW1)
  - ✓ '모바일 웹 사이트에서 가장 일반적인 설정'  
`<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>`

# Media Query



# css46.html

```
<!doctype html>
<html lang="ko">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width">
<title>미디어쿼리</title>
<style>
 body {
 background-color: gray;
 }

 #logo img {
 width: 100%;
 margin-top: 150px;
 }
```

# css46.html

```
@media screen and (max-width: 960px) {
 body {
 background-color: green;
 }
}

@media screen and (max-width:500px) {
 body {
 background-color: yellow;
 }
}

@media screen and (max-width:320px) {
 body {
 background-color: orange;
 }
}

</style>
</head>
```

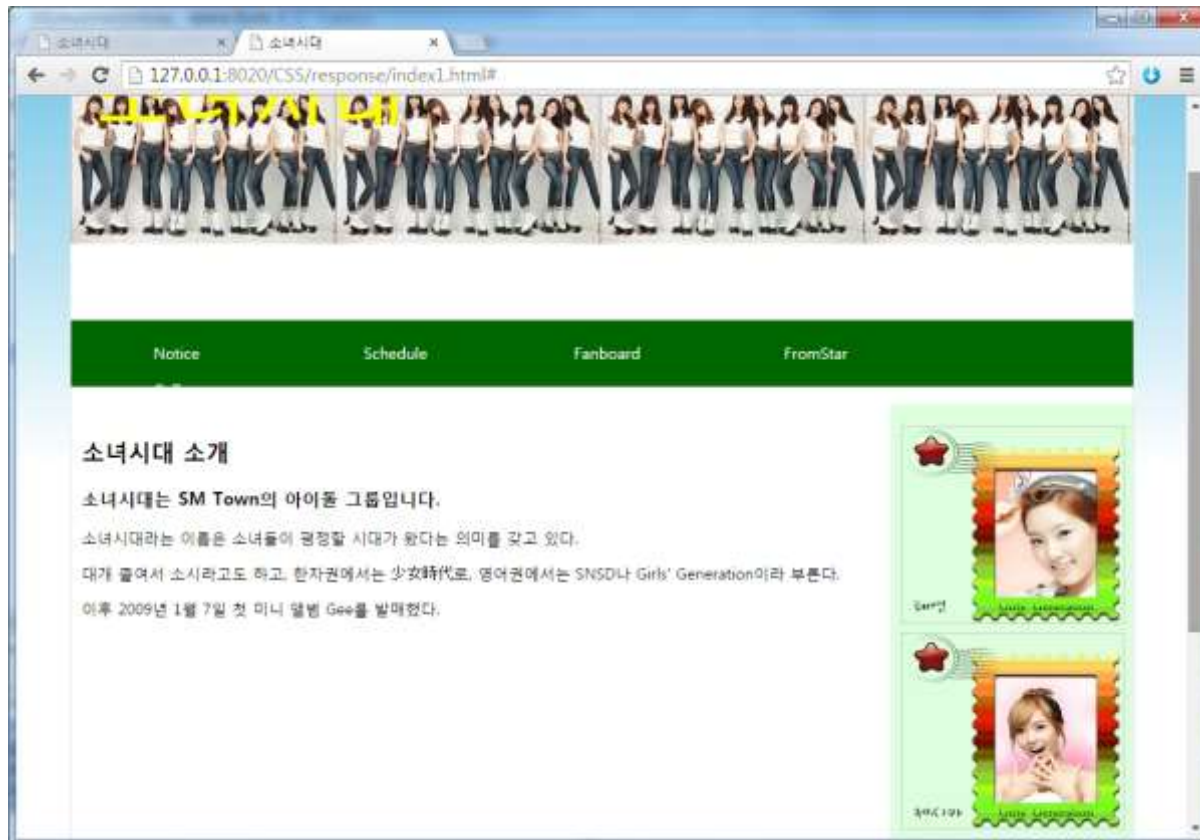
# css46.html

```
<body>
 <div id="logo">

 </div>
</body>
</html>
```



# 미디어 쿼리 이용

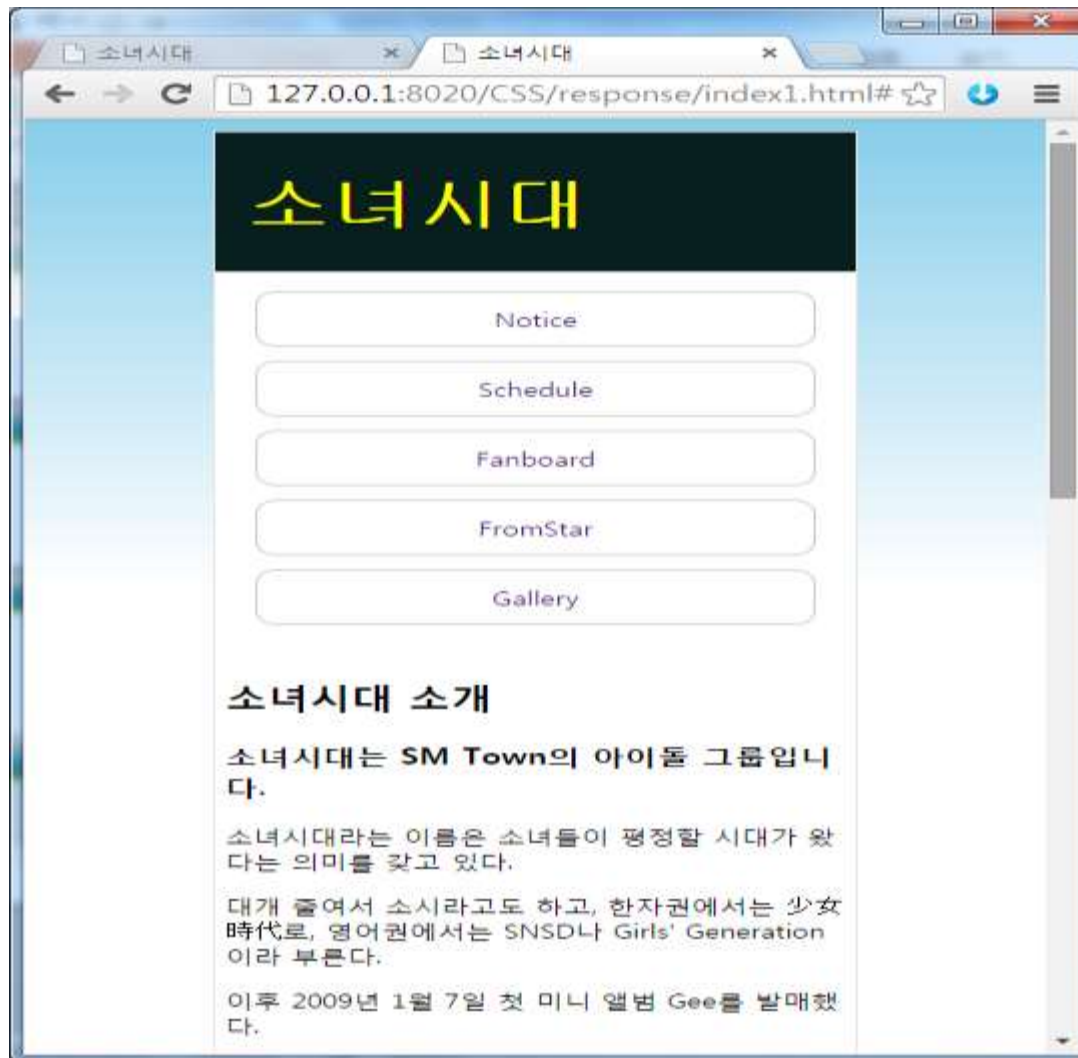


# Media Query





# Media Query



# css47.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width">
<title>소녀시대</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
 <div id="container">
 <header id="mainHeader">
 <h1 class="header1">소녀시대</h1>
 </header>
 <nav id="mainNav">

 Notice
 Schedule
 Fanboard
 FromStar
 Gallery

 </nav>
```

# css47.html

```
<section id="content">
 <h2>소녀시대 소개</h2>
 <article>
 <h3>소녀시대는 SM Town의 아이돌 그룹입니다.</h3>
 <p>소녀시대(少女時代, 영어: Girls' Generation 걸스 제너레이션[*])
 는 대한민국의 9인조 걸그룹으로, 소속사는 SM 엔터테인먼트이다.</p>
 <p>소녀시대라는 이름은 소녀들이 평정할 시대가 왔다
 는 의미를 갖고 있다.</p>
 <p>대개 줄여서 소시라고도 하고, 한자권에서는 少女時
 代로, 영어권에서는 SNSD나 Girls'
 Generation이라 부른다.</p>
 <p>소녀시대는 2007년 8월 2일 데뷔 싱글 <다시 만난
 세계>를 발매하고 8월 5일 SBS 《인기가요》를 통해
 정식 데뷔했다.</p>
 <p>같은 해 11월 1일 한국 첫 정규 앨범 《소녀시대》를
 발매했으며, <Kissing You>와 <Baby
 Baby> 등으로 활동하며 1위에 올랐다.</p>
 <p>이후 2009년 1월 7일 첫 미니 앨범 Gee를 발매했
 다.</p>
 <p>2009년 12월 29일부터는 Into the New World라는
 첫 아시아 투어도 개최했다.</p>
 </article>
</section>
```

# css47.html

```
<aside id="sidebar">

</aside>
<footer id="footer">
 <p>항상 응원해 주시는 소원 여러분~ 너무 고마워요!</p>
 <address>Copyright©2012 S.M.Entertainment Co., Ltd. All
 rights reserved.</address>
</footer>
</div>
</body>
</html>
```

# style.css

```
@charset "utf-8";
body {
 font: 14px "맑은 고딕";
 background: linear-gradient(top, skyblue, white 50%);
 background: -webkit-linear-gradient(top, skyblue, white 50%);
 background: -moz-linear-gradient(top, skyblue, white 50%);
 background-attachment: fixed;
}
#container {
 margin: 0 auto;
 background-color: #fff;
 border: 1px solid #E7E7E7
}
#content {
 background-color: #ffffff;
 padding-top: 5px;
 padding-left: 10px;
}
#sidebar img:hover {
 border: 1px solid #000;
 opacity: 1.0;
}
```

# style.css

```
#footer {
 padding: 10px 0;
 background-color: #061f1f;
 color: #fff;
 text-align: center;
 clear: both;
}
header, section, footer, aside, article, figure {
 display: block;
}
```



# style.css

```
@media screen and (max-width: 559px) {
 #container {
 width: 320px;
 }
 #mainHeader {
 height: 100px;
 background: #061f1f;
 }
 #mainHeader h1 {
 position: absolute;
 margin-left: 15px;
 }
 .header1 {
 font-family: Arial;
 font-size: 3em;
 color: #FF0;
 text-shadow: 0px 1px 1px #000;
 }
 #mainNav ul {
 list-style: none;
 }
 #mainNav ul li {
 width: 280px;
 margin-bottom: 10px;
 margin-left: -20px;
 }
}
```

# style.css

```
#mainNav a, #mainNav a:visited {
 display: block;
 border: 1px solid #ccc;
 border-radius: 10px;
 padding: 10px 5px;
 text-align: center;
 text-decoration: none;
}
#content {
 width: 310px;
 padding: 5px;
}
#sidebar {
 width: 320px;
 margin-top: 5px;
 text-align: center;
}
#sidebar img {
 border: 1px solid #CCC;
 width: 90%;
}
}
```



# style.css

```
@media screen and (min-width: 600px) and (max-width: 1023px) {
 #container {
 width: 580px;
 }
 #mainHeader {
 height: 200px;
 margin-top: 0;
 background-image: url(bg.jpg);
 background-repeat: repeat-x;
 background-position: left top;
 }
 #mainHeader h1 {
 position: absolute;
 margin-top: 35px;
 margin-left: 20px;
 }
 .header1 {
 font-family: Arial;
 font-size: 4em;
 color: #ff0;
 text-shadow: 0px 1px 1px 000;
 }
}
```

# style.css

```
#mainNav {
 background: #060;
 margin-top: -15px;
 width: 580px;
 height: 60px;
}
#mainNav ul {
 list-style: none;
 height: 40px;
 padding-top: 10px;
 padding-bottom: 5px;
}
#mainNav ul li {
 display: inline;
 float: left;
}
#mainNav a, #mainNav a:visited {
 padding: 10px 5px 5px 5px;
 display: block;
 color: #fff;
 width: 100px;
 text-decoration: none;
}
```

# style.css

```
#content {
 width: 550px;
 padding: 10px;
}
#sidebar {
 width: 550px;
 height: 200px;
 background-color: #dfd;
 margin-top: 5px;
 margin-left: 5px;
}
#sidebar img {
 border: 1px solid #ccc;
 text-align: center;
 width: 32%;
}
}
```

# Flexible Box Layout

- ❖ flex라는 디렉토리를 생성하고 필요한 이미지 복사



# Flexible Box Layout

## ❖ index.html 파일을 추가하고 작성

```
<!DOCTYPE html>
<html lang="ko">
<head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>솔로의 식탁</title>
 <link rel="stylesheet" href="flex/style.css">
</head>
```

# Flexible Box Layout

❖ index.html 파일을 추가하고 작성

```
<body>
 <div id="container">
 <header>
 <h1>솔로의 식탁</h1>
 </header>
 <section id="menus">
 <div id="menu1">
 <h2>밥/죽</h2>
 </div>

 <div id="menu2">
 <h2>국/찌개</h2>
 </div>

 <div id="menu3">
 <h2>반찬</h2>
 </div>
 <div id="menu4">
 <h2>일품요리</h2>
 </div>

 <div id="menu5">
 <h2>음료/커피</h2>
 </div>
 </section>
```

# Flexible Box Layout

- ❖ index.html 파일을 추가하고 작성

```
<section id="notice">
</section>

<footer>
 <p>솔로의 식탁</p>
</footer>

</div>
</body>
</html>
```



# Flexible Box Layout

- ❖ 모바일에 맞는 페이지 생성





# Flexible Box Layout

- ❖ flex라는 디렉토리에 style.css 파일을 생성하고 작성

```
* {
 margin:0;
 padding:0;
 box-sizing:border-box;
}
#container {
 width:100%;
}
header {
 width:100%;
}
header h1 {
 font-size:3em;
 text-align: center;
}
```

# Flexible Box Layout

- ❖ flex라는 디렉토리에 style.css 파일을 생성하고 작성

```
#menus {
 width:100%;
}
#menus > div {
 position:relative;
 height:400px;
 border:1px solid black;
 margin-bottom:15px;
}
#menu1, #menu2, #menu3, #menu4, #menu5 {
 width:100%;
}
#menus h2 {
 position:absolute;
 padding:5px;
 font-size:2em;
 color:white;
 text-shadow:3px 3px 5px black;
 right:3%;
 bottom:10px;
}
```

# Flexible Box Layout

- ❖ flex라는 디렉토리에 style.css 파일을 생성하고 작성

```
#menu1 {
 background:url("dish1.jpg") no-repeat center;
 background-size:cover;
}
#menu2 {
 background:url("dish2.jpg") no-repeat center;
 background-size:cover;
}
#menu3 {
 background:url("dish3.jpg") no-repeat center;
 background-size:cover;
}
#menu4 {
 background:url("dish4.jpg") no-repeat center;
 background-size:cover;
}
#menu5 {
 background:url("dish5.jpg") no-repeat center;
 background-size:cover;
}
```

# Flexible Box Layout

- ❖ flex라는 디렉토리에 style.css 파일을 생성하고 작성

```
footer {
 width:100%;
 background:#373737;
 height:200px;
}
footer p {
 font-size:1.2em;
 color:#eee;
 text-align:center;
 line-height:200px;
}
```

# Flexible Box Layout

❖ 태블릿에 맞는 페이지 생성

## 솔로의 식탁



# Flexible Box Layout

## ❖ style.css 파일에 추가

```
@media all and (min-width:768px) {
 #menus {
 display: flex; /* 플렉서블 박스 레이아웃 적용 */
 -ms-flex-wrap: wrap;
 -webkit-flex-wrap: wrap;
 flex-wrap: wrap; /* 여러 줄에 걸쳐 표시 */
 justify-content: space-between; /* 화면 양쪽에 바짝 붙이기 */
 }
 #menu1, #menu2, #menu3, #menu4 {
 width: 49%;
 }
 #menu5 {
 width: 100%;
 }
}
```

# Flexible Box Layout

❖ 모니터에 맞는 페이지 생성

## 솔로의 식탁



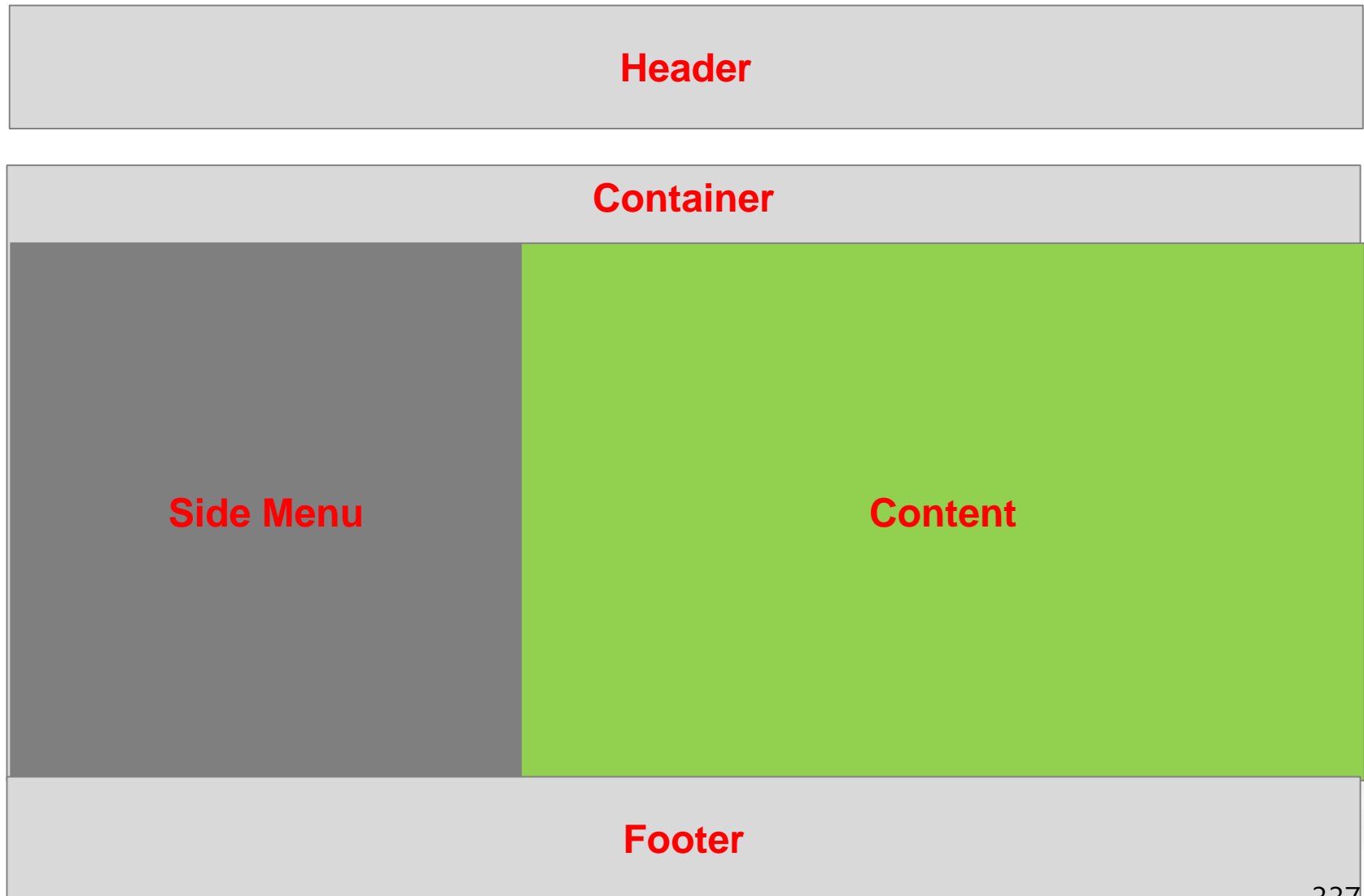
# Flexible Box Layout

## ❖ style.css 파일에 추가

```
@media all and (min-width:992px) {
 #menu1, #menu2, #menu3, #menu4, #menu5 {
 width:33%;
 }
 #menu5 {
 margin-left: 0.5%;
 -ms-flex: 2 2 0;
 -webkit-flex: 2 2 0;
 flex: 2 2 0;
 }
}
```



# 레이아웃 - 2단 레이아웃



# 레이아웃 - 2단 레이아웃

## ❖ 2단 형식의 레이아웃을 만드는 방법

- ✓ float 이용: sidemenu는 float을 left로 설정하고 content는 right로 설정해서 만들 수 있는데 float의 해제가 필요
- ✓ position 이용: sidemenu는 position을 absolute로 하고 content는 position을 relative로 설정



## 호랑이를 주제로 한 민화

그림의 작가

그림의 시대

그림의 재료

그림의 주인공

그림의 활용

### 승운홍호도

사람 이름 같기도 한 이 그림을 처음 보았을 때 티셔츠에 새길 그림을 일러스트레이터로 작업 중이었다. 비록 그림을 보며 원본대로 그려내는 작업이었지만 즐거웠다. 용과 겨루는 황금빛 호랑이의 모습이 웅맹하기도 했지만 내게는 약간 귀여워 보였고 무엇보다 순수해 보였다. 어둠 속에서 나타난 비겁한 용보다는 맑은 물을 뿜어 신성한 기운을 드러내는 우리 호랑이에서 무한한 애정을 느꼈기 때문이다.

### 길상백호

호랑이 그림은 예로부터 아래로 내려오는 것보다 위로 올라가는 형상이 더 좋다고 하였다. 하지만 이 그림은 달랐다. 아래로 내려오고 있기는 하지만 용이나는 등의 털과 그윽하게 쳐다보는 두 눈은 백호의 여유로움을 충분히 공감할 수 있게 해주었기 때문이다. 채도가 내려앉아 버린 백호의 그림이지만 건조해 보이기보다는 은빛으로 축축해 보이는 호랑이의 자태에 반했다고 할까? 알록달록함이 주는 아기자기함은 없지만 세련된 무채색에 다부지게 밟아 짙은 앞발에서 오히려 나를 지켜주는 든든함까지 느껴진다. 호랑이가 밟은 풀잎들에서도 우리를 한 공간 안에 무사히 존재하게 하는 안정감이 있는 듯하다. 그림을 감상할 때 그 시대적 배경과 소재의 재질 등을 함께 감상하여 아는 즐거움을 늘려야 하겠지만 나는 아무 부수적인 요소들을 제하고 오로지 내 눈과 그림이 서로 교감하며 빚어내는 상상의 공간을 좋아한다. 시대를 알고 의도를 아는 것도 중요하겠지만 작가가 우리에게 오로지 늘 같은 그것만 바라지는 않았을 것 같다. 심지어 바랬다고 해도 그냥 미안하게 생각하고 나만의 감상포인트를 놓치지 말아야겠다. 이것도 정체성을 잃지 않으려는 관객의 필사적인 노력으로 보아주기를 바라본다.

Copyright © Allright Reserved.

# twolayout.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
<title>CSS</title>
<meta charset="utf-8">

<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 img, fieldset {
 border: none;
 }

 li img, dd img {
 vertical-align: top;
 }
```

# twolayout.html

```
body {
 background: #666;
 font: 13px/1.5 "굴림", Gulim;
 color: #333;
}

#wrapper {
 width: 630px;
 margin: 0 auto;
}

#header {
 background: #eee;
}

h1 {
 font: bold 30px "휴먼모음T", Gungseo;
 color: #cc0;
 background: #123456;
 padding: 10px;
 text-align: center;
}
```

# twolayout.html

```
h1 strong {
 font-size: 300%;
 color: #ffd;
}

h1 span {
 font-size: 150%;
}

h1 em {
 font-style: normal;
 font-size: 200%;
 color: #ccc;
}

#container {
 position: relative;
 width: 100%;
 background: #ccc;
}
```

# twolayout.html

```
#menu {
 position: absolute;
 left: 0;
 top: 0;
 width: 150px;
 padding: 20px;
}

#menu li {
 padding: 5px 0;
}

#menu li a {
 color: #000;
}

#menu li a:hover {
 color: #662;
 border-bottom: 1px dashed #662;
}
```

# twolayout.html

```
#content {
 position: relative;
 left: 130px;
 width: 500px;
 background: #fff;
}
h2 {
 padding: 10px 20px;
 font: bold 25px "궁서", Gungseo;
 background: #800;
 color: #ee0;
}
p {
 padding: 20px;
 text-align: justify;
}
#footer {
 background: #aaa;
}
address {
 font-style: normal;
 padding: 10px 20px;
 color: #000;
}

</style>
</head>
```



# twolayout.html

<body>

<div id="wrapper">

<div id="header">

<h1>

<strong>호랑이</strong>를 <span>주제</span>로 한 <em>민화</em>

</h1>

</div>

<div id="container">

<ul id="menu">

<li><a href="#">그림의 작가</a></li>

<li><a href="#">그림의 시대</a></li>

<li><a href="#">그림의 재료</a></li>

<li><a href="#">그림의 주인공</a></li>

<li><a href="#">그림의 활용</a></li>

</ul>

<div id="content">

<h2>승운옹호도</h2>

<p>사람 이름 같기도 한 이 그림을 처음 보았을 때 티

셔츠에 새길 그림을 일러스트레이터로 작업 중이었다. 비록

그림을 보며 원본대로 그려내는 작업이었지만 즐거웠다. 용과 겨루는 황금빛 호랑이의 모습이 용맹하기도 했지만 내게는 약간

귀여워 보였고 무엇보다 순수해 보였다. 어둠 속에서 나타난 비겁한 용보다는 맑은 물을 튀며 신성한 기운을 드러내는 우리

호랑이에서 무한한 애정을 느꼈기 때문이

다.</p>

# twolayout.html

<h2>길상백호</h2>

<p>호랑이 그림은 예로부터 아래로 내려오는 것보다 위로 올라가는 형상이 더 좋다고 하였다. 하지만 이 그림은 달랐다. 아래로 내려오고 있기는 하지만 윤기나는 등의 털과 그윽하게 쳐다보는 두 눈은 백호의 여유로움을 충분히 공감할 수 있게 해주었기 때문이다. 채도가 내려앉아 버린 백호의 그림이지만 건조해 보이기보다는 은빛으로 축축해 보이는 호랑이의 자태에 반했다고 할까? 알록달록함이 주는 아기자기함은 없지만 세련된 무채색에 다부지게 밟아 짚은 앞발에서 오히려 나를 지켜주는 든든함까지 느껴진다. 호랑이가 밟은 풀잎들에서도 우리를 한 공간 안에 무사히 존재하게 하는 안정감이 있는 듯하다. 그림을 감상할 때 그 시대적 배경과 소재의 재질 등을 함께 감상하여 아는 즐거움을 늘려야 하겠지만 나는 아무 부수적인 요소들을 제하고 오로지 내 눈과 그림이 서로 교감하여 빚어내는 상상의 공간을 좋아한다. 시대를 알고 의도를 아는 것도 중요하겠지만 작가가 우리에게 오로지 늘 같은 그것만 바라지는 않았을 것 같다. 심지어 바랬다고 해도 그냥 미안하게 생각하고 나만의 감상포인트를 놓치지 말아야겠다. 이것도 정체성을 잃지 않으려는 관객의 필사적인 노력으로 보아주기를 바라본다.</p>

</div>

</div>

<div id="footer">

<address>Copyright &copy; Allright Reserved.</address>

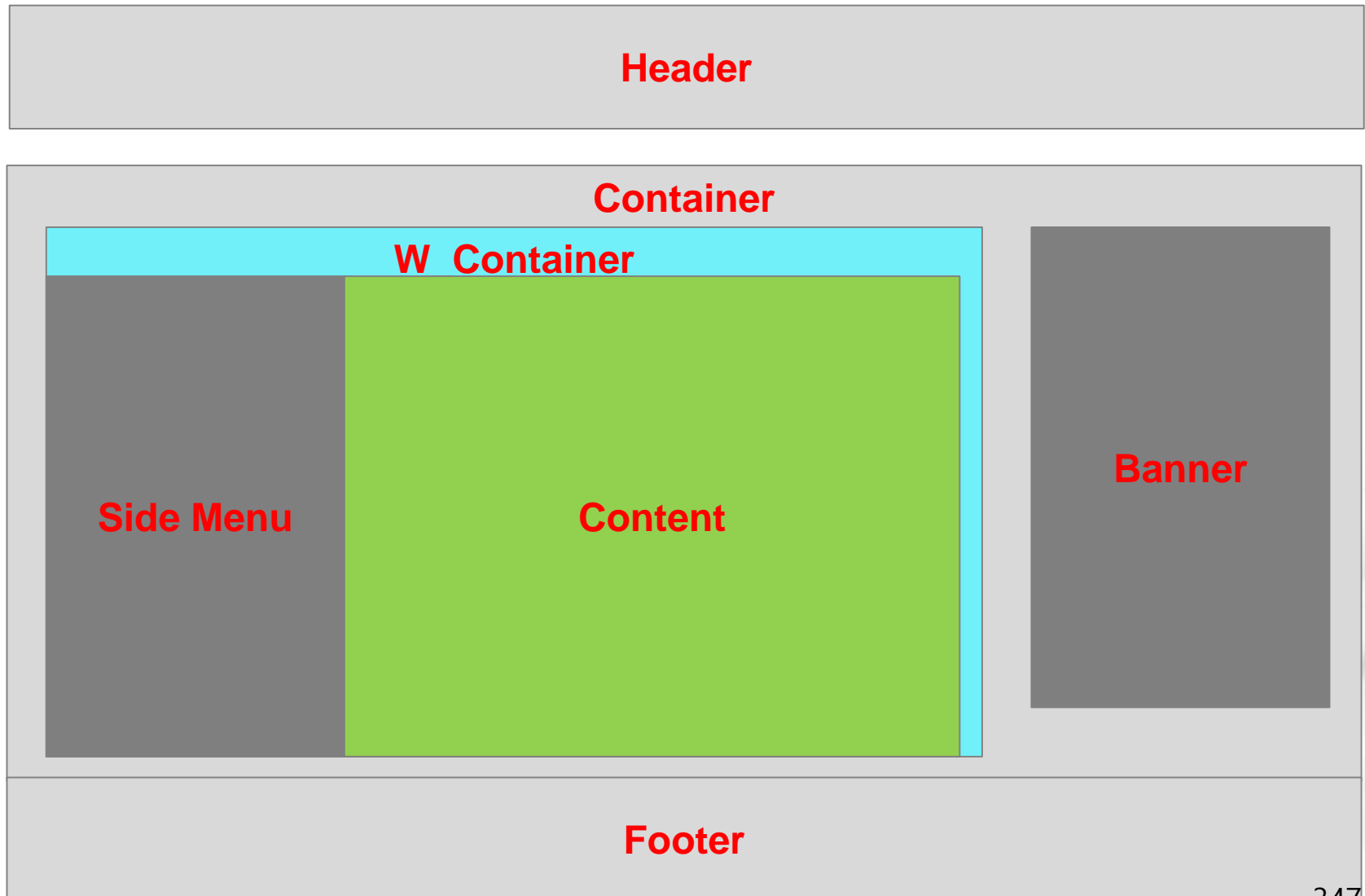
</div>

</div>

</body>

</html>

# 3단 레이아웃 만들기



# Musée d'Orsay

Musée d'Orsay Musée d'Orsay Musée d'Orsay Musée d'Orsay Musée d'Orsay

Royal Academy of Arts, THE REAL VAN GOGH THE ARTIST AND HIS LETTERS

**ARTIST**  
[작가의 작품](#)  
[작가의 시대](#)  
[작가의 일생](#)

**피리부는 소년**

에두아르 마네  
1866 캔버스에 유채, 161\*97cm, 파리 오르세 Edouard MANET, Le Filin, H.T, 161\*97cm

**고흐의 방**

빈센트 반 고흐  
1889 캔버스에 유채, 57.5\*74cm, 파리 오르세 Vincent Van GOGH, La Chambre de Van Gogh, H/T, 57.5\*74cm

**황색 그리스도가 있는 화가의 자화상**

폴 고갱  
1890-1891 캔버스에 유채, 38\*46cm, 파리 오르세 Paul Gauguin, Portrait de l'artiste au Paul Gauguin, H/T, 38\*46cm

**오페라좌의 관현악단**

에드가 드가  
1868-1869 캔버스에 유채, 32\*46cm, 파리 오르세 Edgar Degas, L'orchestre de l'Opera H/T, 38\*46cm

**전세계인들에게  
제일 사랑받는  
서양미술사의 시대**

**Workshop Go**  
Life drawing workshop

**Summer exhibition**  
A-level Summer Exhibition  
Online 2010

**RA collection**  
RA Collections

348

# threelayout.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
<title>CSS</title>
<meta charset="utf-8">

<style type="text/css">
 * {
 margin: 0;
 padding: 0;
 }

 li {
 list-style-type: none;
 }

 img, fieldset {
 border: none;
 }

 li img, dd img {
 vertical-align: top;
 }
```

# threelayout.html

```
@font-face {
 font-family: 'NGothic';
 src: url('font/NanumGothic.woff') format('woff');
}

@font-face {
 font-family: 'NGothicB';
 src: url('font/NanumGothicBold.woff') format('woff');
}

@font-face {
 font-family: 'NGothicEB';
 src: url('font/NanumGothicExtraBold.woff') format('woff');
}
```

# threelayout.html

```
body {
 font: 13px "NGothic", Arial;
 background: #aaa;
}

/* Layout */
#wrapper {
 width: 800px;
 margin: 0 auto;
 background: #fff;
}

#header {
 text-align: center;
 background: #fa0;
 color: #fff;
}

#container {
 float: left;
 width: 100%;
}
```

# threelayout.html

```
.cover {
 float: left;
 width: 600px;
}

.content {
 float: right;
 width: 400px;
 background: #ddd;
 color: #123456;
}

.sidemenu {
 float: left;
 width: 200px;
}

.aside {
 float: right;
 width: 200px;
}

#footer {
 clear: both;
}
```



# threelayout.html

```
/* Detail */
a {
 font-weight: bold;
}

a:hover {
 border-bottom: 1px dashed;
}

h1 {
 font-size: 800%;
}

h1 span {
 display: block;
 font-size: 20px;
 padding-bottom: 10px;
}

.stit {
 background: #ccc;
 color: #333;
 padding: 5px;
}
```

# threelayout.html

```
.content dt a {
 display: block;
 color: #ddd;
 font: bold 130% "N Gothic", Arial;
 padding: 5px;
 text-decoration: none;
 background: #123456;
}
```

```
.content dt+dd {
 font-weight: bold;
 font-size: 120%;
}
```

```
.sidemenu dt {
 font: bold 130% "N Gothic", Arial;
 background: #900;
 color: #fff;
}
```

```
.sidemenu a {
 color: #000;
}
```

# threelayout.html

```
.aside dt {
 font: bold 170% "NGothic", Arial;
 color: #999;
}

.aside dt .all {
 text-decoration: underline;
}

.aside dt .love {
 color: #fa0;
 border-bottom: 1px solid #fa0;
}

.aside dt strong {
 color: #333; /*border:none;*/
}

.aside a {
 color: #000;
}

.aside span {
 display: block;
}
```

# threelayout.html

```
dt, dd {
 padding: 10px;
}

address {
 font: bold normal 100% "NGothic", Arial;
 padding: 10px;
 background: #123456;
 color: #eee;
 text-align: center;
}

address a {
 color: #eee;
}

</style>
</head>
```

# threelayout.html

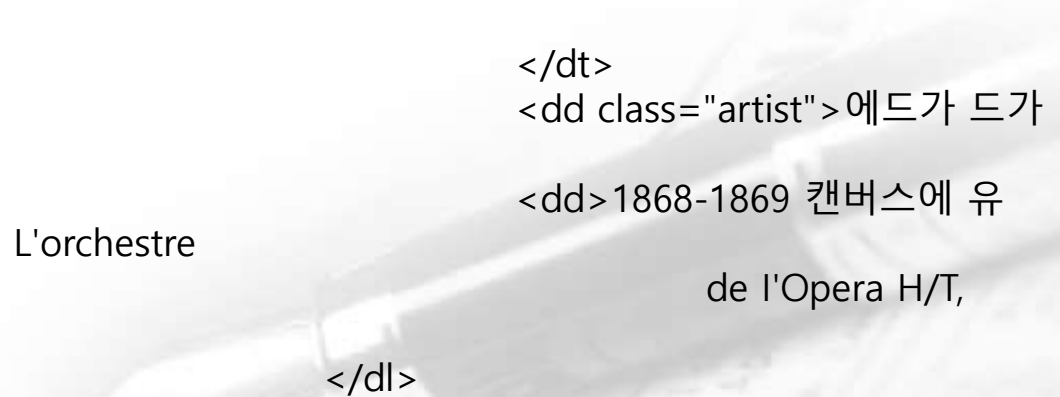
```
<body>
 <div id="wrapper">
 <div id="header">
 <h1>
 Musée d'OrsayMusée d'Orsay Musée d'Orsay
 Musée d'Orsay Musée d'Orsay
 </h1>
 <p class="stit">
 Royal Academy of Arts, THE REAL VANGOGH
 HIS LETTERS
 THE ARTIST AND
 </p>
 </div>
```

# threelayout.html

```
<div id="container">
 <div class="cover">
 <div class="content">
 <dl>
 <dt>
 피리부는 소년
 </dt>
 <dd>에두아르마네 </dd>
 <dd>1866 캔버스에 유채,
 161*97cm, 파리 오르세 Edouard MANET, Le Fifre,
 H.T, 161*97cm </dd>
 </dd>
 </dl>
 <dl>
 <dt>
 고흐의
 방
 </dt>
 <dd class="artist">빈센트반고흐 </dd>
 <dd>1889 캔버스에 유
 채,57.5*74cm,파리 오르세 Vincent Van GOGH,La Chambre
 de Van Gogh, H/T,
 57.5*74cm </dd>
 </dd>
 </dl>
 </div>
 </div>
</div>
```

# threelayout.html

```
<dl>
 <dt>
 황색 그리스도가 있는 화가의 자화상
 </dt>
 <dd class="artist">폴 고갱</dd>
 <dd>1890-1891 캔버스에 유
채,38*46cm,파리 오르세 Paul Gauguin Portrait de
Gauguin, H/T, 38*46cm</dd>
 </dd>
 <dt>
 오페라
좌의 관현악단
 </dt>
 <dd class="artist">에드가 드가
 <dd>1868-1869 캔버스에 유
채,32*46cm,파리 오르세 Edgar Degas, L'orchestre
38*46cm</dd>
 </dd>
</dl>
```



# threelayout.html

```

밀레</dd>
채,55.5*66cm,파리 오르세 Jean-Franqois Millet,
55.5*66cm</dd>

장식을 단 베르트모리조
네</dd>
Berthe Morisot au bouquet de H/T,

</div>
<dl>
<dt>
만종
</dt>
<dd class="artist">장 프랑수아
<dd>1857-1859 캔버스에 유
L'Angelus H/T,

</dl>
<dl>
<dt>
제비꽃
</dt>
<dd class="artist">에두아르마
<dd>1872 Eduouard Manet
55*38cm</dd>

</dl>
```



# threelayout.html

```
<dl class="sidemenu">
 <dt>ARTIST</dt>
 <dd>
 작가의 작품
 </dd>
 <dd>
 작가의 시대
 </dd>
 <dd>
 작가의 일생
 </dd>
</dl>
</div>
```

# threelayout.html

```
<dl class="aside">
 <dt>
 전세계인들에게
 제일 사랑받는 서양미술사의
 시대
 </dt>
 <dd>
 Workshop Go
 Life drowing workshop
 </dd>
 <dd>
 Summer exhibition
 A-level Summer
 Exhibition Online 2010
 </dd>
 <dd>
 RA collection RA
 Collections
 </dd>
</dl>
</div>
```

# threelayout.html

```
<div id="footer">
 <address>
 © 2012 Les Amis du Musée d'Orsay. All rights
reserved. amis@amis-musee-orsay.org
 Legal status
 </address>
</div>
</body>
</html>
```



# 드롭다운 네비게이션 메뉴

처음으로

회사소개

사업분야

게시판

사이트 맵

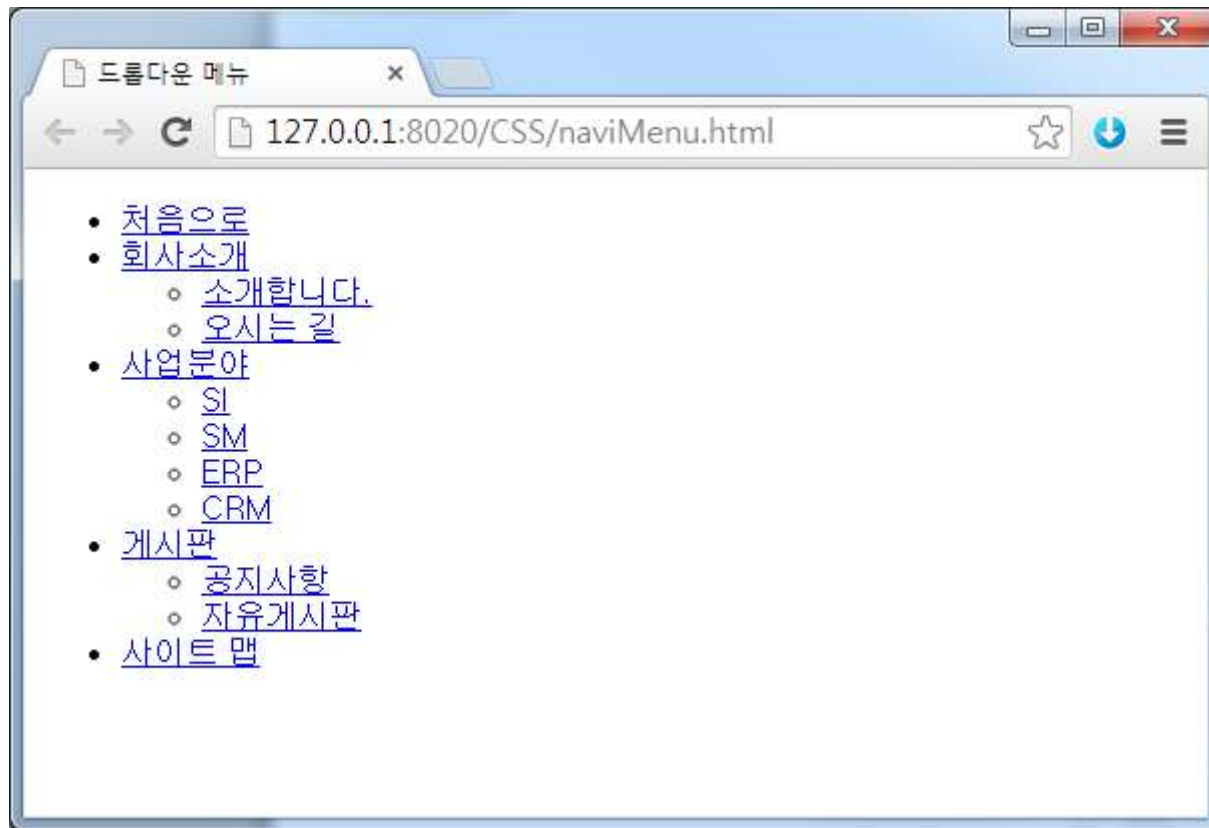
SI

SM

ERP

CRM

# 드롭다운 네비게이션 메뉴



# navigationmenu.html

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>드롭다운 메뉴</title>

</head>

<body>
 <ul id="nav">
 처음으로

 회사소개

 소개합니다.
 오시는 길

 사업분야

 SI
 SM
 ERP
 CRM

```

# navigationmenu.html

```
 게시판

 공지사항
 자유게시판

사이트 맵

</body>
</html>
```



# 스타일 추가

<style>

```
#nav {
 /* 메뉴 전체 영역에 대한 스타일 */
 list-style-type: none; /* 불릿 기호 없음 */
 height: 45px; /* 내비게이션 높이 */
 padding: 6px 7px; /* 위아래 6px, 좌우 7px씩 안여백 */
 margin: 0; /* 바깥 여백 없음 */
 background: #7d7d7d; /* 내비게이션 전체 배경색 */
 border-radius: 1.5em; /* 모서리 둥글게 */
}

#nav li { /* 메뉴 각 항목에 대한 스타일 */
 float: left; /* 왼쪽부터 표시 */
 position: relative; /* 순서대로 표시 */
 margin: 5px 10px; /* 위아래 5px, 좌우 10px씩 바깥 여백 */
 padding: 0; /* 안여백 없음 */
}
```



# 스타일 추가

```
#nav li a { /* 메뉴 항목에서도 링크에 대한 스타일 */
 display: block; /* 영역을 만듦 */
 font-family: "굴림"; /* 글꼴 */
 font-weight: 600; /* 진하게 */
 font-size: 1em; /* 글자크기 */
 padding: 10px 20px; /* 안여백 */
 color: #e7e5e5; /* 링크 글자색 */
 text-decoration: none; /* 밑줄 없앴 */
 margin: 0; /* 바깥 여백 없음 */
 border-radius: 1.4em; /* 링크의 모서리를 둥글게 */
 text-shadow: 0 1px 1px rgba(0, 0, 0, .3); /* 텍스트 그림자 */
}

#nav li: hover>a { /* 메뉴 항목 위로 마우스 올렸을 때 스타일 */
 background: #ebebeb; /* 배경색 */
 color: #444; /* 글자색 */
 text-shadow: 0 1px 1px rgba(255, 255, 255, 1); /* 글자 그림자 */
 border-radius: 1.4em; /* 모서리 둥글게 */
 -webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, 0.2); /* 메뉴 항목에 그림자 */
 -moz-box-shadow: 0 1px 1px rgba(0, 0, 0, 0.2); /* 메뉴 항목에 그림자 */
}
```

# 스타일 추가

```
#nav ul { /* 서브 메뉴 영역에 대한 스타일 */
 margin: 0; /* 서브 메뉴 바깥 여백 없음 */
 padding: 0; /* 서브 메뉴 안여백 없음 */
 list-style: none; /* 목록의 불릿 없앰 */
 position: absolute; /* 고정 위치에 표시 */
 left: 0; /* 부모 요소와 left 좌표 값은 같게 */
 top: 45px; /* 부모 요소로부터 아래로 45px 밑에 표시 */
 width: 150px; /* 서브 메뉴 너비 */
 background: #ddd; /* 서브 메뉴 배경색 */
 border: 1px solid #b4b4b4; /* 서브 메뉴 테두리를 그림 */
 border-radius: 8px; /* 모서리를 둥글게 처리 */
 -webkit-box-shadow: 0 1px 3px rgba(0, 0, 0, .3);
 /* 서브 메뉴 전체에 그림자 효과 추가 */
 -moz-box-shadow: 0 1px 3px rgba(0, 0, 0, .3);
 box-shadow: 0 1px 3px rgba(0, 0, 0, .3);
 opacity: 0; /* 투명하게 표시 - 결국 평소엔 안보임 */
}

#nav li:hover ul { /* 메인 메뉴 위로 마우스 오버했을 때 서브 메뉴 스타일 */
 opacity: 1; /* 서브 메뉴 불투명해짐 -> 화면에 보임 */
}
```

# 스타일 추가

```
#nav ul li { /* 서브 메뉴 각 항목의 스타일 */
 float: none; /* 세로로 표시 */
 margin: 0; /* 바깥 여백 없음 */
 padding: 0; /* 안 여백 없음 */
}
```

```
#nav ul a { /* 서브 메뉴 항목의 링크 스타일 */
 font-weight: normal; /* 진하기 보통으로 */
 text-shadow: 0 1px 0 #fff; /* 글자 그림자 */
 color: #333; /* 글자색 */
}
```

```
#nav ul li:hover a { /* 서브 메뉴 항목 위로 마우스 오버했을 때의 스타일 */
 background: #0078ff; /* 배경색 */
 color: #fff; /* 글자색 */
 text-shadow: 0 1px 1px rgba(0, 0, 0, 0.1); /* 글자 그림자 */
 border-radius: 0; /* 모서리 둥글게 처리 안함 */
}
```

# 스타일 추가

```
#nav ul li:first-child>a { /* 서브 메뉴 첫번째 항목의 스타일 */
 -webkit-border-top-left-radius: 9px; /* 왼쪽 윗부분 둥글게 */
 -moz-border-radius-topleft: 9px; /* 오른쪽 윗부분 둥글게 */
 -webkit-border-top-right-radius: 9px;
 -moz-border-radius-topright: 9px;
}

#nav ul li:last-child>a { /* 서브 메뉴 마지막 항목의 스타일 */
 -webkit-border-bottom-left-radius: 9px; /* 왼쪽 아랫부분 둥글게 */
 -moz-border-radius-bottomleft: 9px; /* 오른쪽 아랫부분 둥글게 */
 -webkit-border-bottom-right-radius: 9px;
 -moz-border-radius-bottomright: 9px;
}
</style>
```