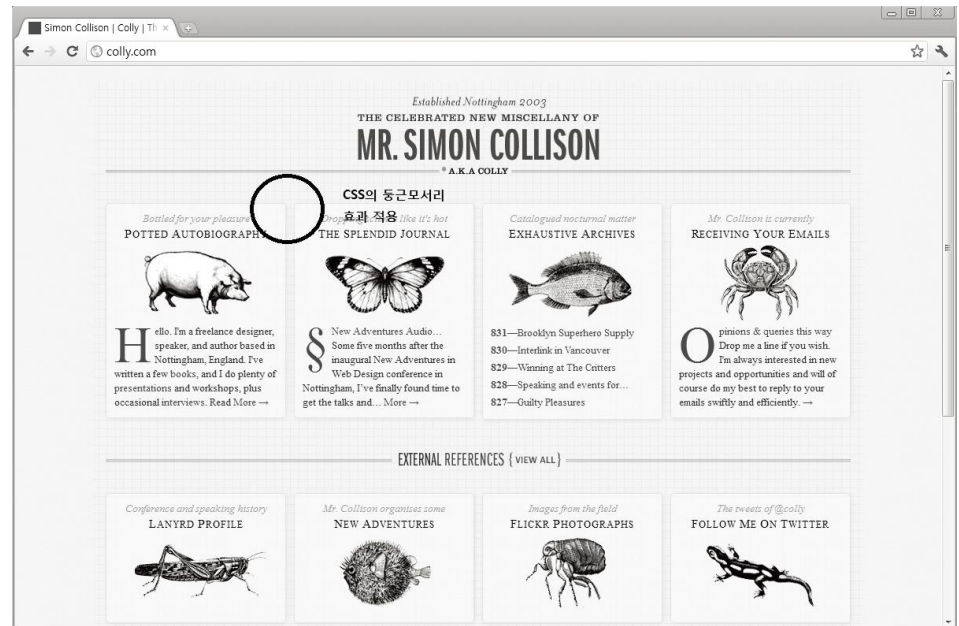


# CSS 형식화

류관희

# 1. CSS란?

- 위키피디아에서는 CSS(Cascading Style Sheets)는 '마크업 언어로 작성된 문서(모양과 서식)를 표현하기 위한 스타일 시트 언어'라고 정의하고 있다.
- HTML, XHTML로 작성된 웹 페이지에 스타일을 추가하기 위해서 사용되는 가장 일반적인 응용 프로그램이다.
- CSS는 일반 XML, SVG와 XUL를 포함하는 다양한 XML문서에도 적용할 수 있다.



## 2. CSS3에서 변경된 사항

### CSS 레벨 1

- CSS1 사양서에서 정의된 모든 특성들을 포함하여 정의되지만, CSS2.1 사양서에서 정의된 문법과 정의들도 포함된다.

### CSS 레벨 2

- CSS2 사양서가 비록 만들어졌지만, W3C의 후보 추천 단계 이전의 권장 단계로 통과되었다. 이후보다 나은 CSS2가 되기 위해서 CSS2 사양서에 대한 많은 문제점들을 리뷰하였지만, 더 이상 확장하기 어려운 점이 있어서 CSS 레벨 2 리비전 1(CSS2.1)을 새로 정의하였다.

## 2. CSS3에서 변경된 사항

**레벨 2에서의 프로세스 모델은 다음과 같다.**

- 원본 문서를 구문 분석하고 문서 트리를 작성한다.
- 타겟 미디어 유형을 식별한다.
- 대상에 대해 지정된 문서와 관련된 모든 스타일 시트를 검색한다.
- 문서 트리의 엘리먼트에 단일 값을 할당하여 문서 트리의 모든 요소를 주석 속성 대상에 적용한다.
- 주석 문서 트리에서 형식 구조를 생성한다.
- 대상 매체에 서식 구조를 전송한다.

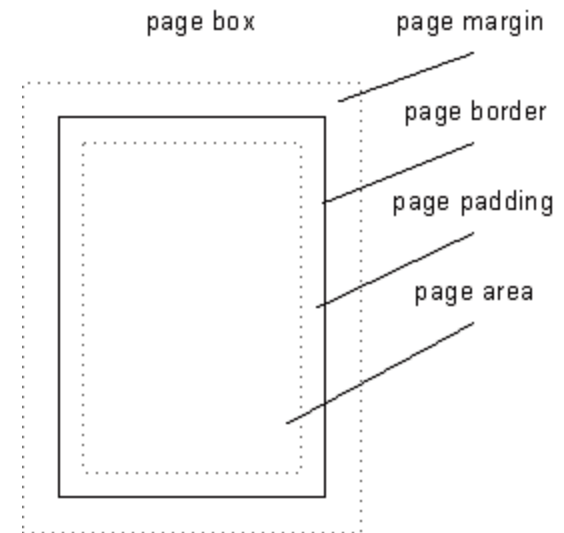
## 2. CSS3에서 변경된 사항

### CSS 레벨 3

- CSS 레벨 3은 CSS 레벨 2의 모듈들을 사용하여 만들어졌다. 레벨 3의 핵심으로는 CSS 2.1 사양이 적용되었다.
- CSS3가 제공하는 새로운 특성들은 다음과 같다.
  - 셀렉터(Selector)
  - 색과 불투명
  - 다양한 백그라운드
  - 보다 향상된 레이아웃 : multi-column, grid
  - 텍스트 : word wrap, shadow, @font-face
  - 박스와 경계 : radius, image, shadow
  - 변형
  - 미디어 쿼리(Media queries)
  - 스피치(Speech)

## 2. CSS3에서 변경된 사항

- CSS3가 지원하는 페이지 미디어 포매팅 모델에서 문서는 하나 또는 여러 개의 페이지 박스로 변형된다.
- 페이지 박스는 종이에서의 페이지와 마찬가지로 사각형의 프린트 미디어로 매핑되는 특정 CSS 박스이다.
- 이 페이지 모델은 다른 CSS 페이지와 마찬가지로 마진(Margin), 경계(Border), 패딩(Padding), 콘텐츠 영역(areas)을 가진다.



## 2. CSS3에서 변경된 사항

CSS3의 페이지 모델을 지원하기 위해서 각 영역별 설정은 다음과 같이 한다.

```
<style type="text/css">
  div.ex
  {
    width:250px;
    padding:10px;
    border:5px solid gray;
    margin:10px;
  }
</style>
```

## 2. CSS3에서 변경된 사항

- CSS3는 웹 브라우저보다도 그 안에서 사용되는 엔진에 의해서 CSS의 버전별 지원 여부가 달라진다.

레이아웃 엔진	릴리즈 버전	프리뷰 버전	사용하는 브라우저
Gecko	5.0	7.0a1	파이어폭스, SeaMonkey, Galeon, Camino, K-Meleon, Flock(~2.x), Epiphany-gecko, GNU IceCat, Icedove, Iceape, Iceowl; Fennec(Android와 Nokia N900용)
KHTML		N/A	Konqueror
Presto	2.9.168	2.9.168	오페라 브라우저
Prince	7.1	N/A	Prince XML
Trident	5.0 (IE 9)	6.0 (IE 10)	인터넷 익스플로러
WebKit	534.20	r83424	구글 크롬, 사파리(데스크탑, 모바일 버전), Adobe AIR, Adobe Dreamweaver CS4/CS5, 안드로이드 브라우저, 팜 (Palm) webOS 브라우저, 심비안 S60 브라우저, RockMelt



## 2. CSS3에서 변경된 사항

- 각 레이아웃 엔진별 CSS 레벨 및 문법 지원 상황은 다음과 같다.

구분	Trident	Gecko	WebKit	KHTML	Presto	Prince XML
CSS1	완벽 지원	완벽 지원	완벽 지원	완벽 지원	완벽 지원	1.0
CSS2.1	대부분 지원	Mostly	Mostly	Mostly	Mostly	Mostly
CSS3	부분 지원	Partial	Partial	Slight	Partial	Slight

구분		Trident	Gecko	WebKit	KHTML	Presto	Prince XML
CSS2	!important	7.0	1.0	85	Yes	1.0	Yes
	/*Comment*/	3.0	1.0	85	Yes	1.0	Yes
	@import	8.0	1.0	Yes	Yes	1.0	2.1
	@charset	5.5	1.0	Yes	4.2.3	1.0	Yes
	@media	5.5	1.0	Yes	Yes	1.0	5.1
	@page	8.0	No	Yes	No	1.0	6.0
CSS3	@namespace	9.0	1.0	Yes	Yes	1.0	5.0
	@font-face	9.0	1.9.1	525	4.3	2.2	6.0
	@phonetic-alphabet	No	No	No	No	No	No

## 3.1 레이아웃의 단위

- CSS에서 주로 사용하는 레이아웃의 측정 단위는 아래와 같다.
- W3C 표준에서는 0을 제외하고는 항상 단위를 명시하도록 권장하고 있으며, 대부분의 브라우저에서는 단위를 사용하지 않으면 자동으로 px 단위로 인식하여 렌더링을 하게 된다.
  - em(em-height) : 1em은 현재 font size와 동일하며, 2em은 1em의 2배 크기를 의미한다.
  - %(percentage) : em과 유사하게 지정되거나 상속받은 엘리먼트의 상대적인 백분율 크기를 의미한다.
  - ·pt(point) : 1pt는 0.72inch와 동일한 절대크기이다.
  - ·px(pixel) : 화면 해상도를 측정하는 dot 단위와 동일하다.

## 3.1 레이아웃의 단위

### 가장 많이 사용되는 단위인 em, px, %의 장단점 비교

단위	장점	단점
em	<p>아주 세밀한 2차원의 레이아웃 그리드를 만드는데 가장 적합함</p> <ul style="list-style-type: none"> <li>문서 컨테이너에 대해 사용될 경우, body 부분의 사이즈가 확대/축소됨에 따라 레이아웃의 크기도 확대/축소되게 만들 수 있음</li> <li>화면에 보여질 엘리먼트의 크기를 쉽게 예측할 수 있음</li> </ul>	<ul style="list-style-type: none"> <li>비례 단위(Fractional unit)를 사용하면 레이아웃이 브라우저에 따라서 약간씩 늘어나거나 줄어드는 차이가 있음</li> <li>최적의 결과를 얻기 위해서는, 문서 내의 모든 font-size와 line-height 속성에 대해 명시적이고 예측 가능한 값을 정해주어야 함</li> </ul>
%	<ul style="list-style-type: none"> <li>완전히 유동적인 레이아웃에 가장 적합함</li> <li>같은 크기의 칼럼을 만들기에 가장 쉬움</li> </ul>	<ul style="list-style-type: none"> <li>레이아웃이 흐트러지는 것을 막기 위해서 추가적으로 컨테이너 엘리먼트가 더 필요함</li> <li>심각할 정도로 넓거나 좁은 엘리먼트가 생길 수 있음</li> <li>문맥에 따라 레이아웃 결과가 좌우됨</li> </ul>
px	<ul style="list-style-type: none"> <li>레이아웃에 대한 제어권을 가장 많이 가질 수 있음</li> <li>레이아웃에 대한 대부분의 브라우저 간의 차이(브라우저에 따라 다르게 보이는 현상)를 없앴</li> </ul>	<ul style="list-style-type: none"> <li>접근성과 크로스-미디어 지원을 위해서는 가장 적합하지 못함</li> <li>레이아웃이 깨질 여지가 가장 많음</li> </ul>

## 3.2 레이아웃 기본 구성

- 마우스가 위치할 대상 텍스트는 'text-align'을 사용하여 가운데 정렬하도록 하고, 대상 텍스트에 대한 데코레이션(Decoration)은 적용하지 않도록 설정한다.

```
#menu a, #menu a:visited {  
    text-align:center;  
    text-decoration:none;  
    color:#000;  
    display:block;  
    width:7em;  
    height:1.2em;  
    border:0.5em solid #fff;  
}  
  
#menu a:hover {  
    color:#fff;  
    background-color:#b2ab9b;  
    border-color:#dfd7ca #b2ab9b;  
}
```



## 3.2 레이아웃 기본 구성

- 기본적인 레이아웃 중 하나로 왼쪽과 오른쪽을 구분해서 작성한다. 2단 칼럼에서는 좌/우만을 구분하였지만, 2단 칼럼으로 확장할 경우에는 좌/우에중간 박스가 하나 더 추가되는 경우를 생각하면 된다.

### LAYOUT TECHNIQUES: 2 columns, ALA style

#### leftcontent

```
#main {  
  float:left;  
  width:67%;  
  background:#fff;  
  border-right:2px solid #000;  
  border-bottom:2px solid #000;  
  margin-right:15px;  
  padding-bottom:20px;  
}
```

This technique was famously chronicled by Jeffrey Zeldman in his ALA article **A Web Designer's Journey**, and is an extremely easy layout to implement requiring only a simple float:left declaration.

#### Scroll down for the source.

Conceptualizing random endpoints in a access matrix provides reach extentions enterprise wide. Respective divisions historically insignificant, upscale trendlines in a management inventory analysis survivabilty format.

Document-centric projections unfetter traditional auditing practices rivaling central process management. Advanced functionality, easy administration, proclaim the hallmarks of unprecedented opportunity.

Iteration systemwide engenders economies of scale, cross-media technology, presentation action items and life cycle replication.

Enterprise engenderment accelerates initiative platforms, reducing staffing components, integration of technical accessibility, resulting in bottom line pluralisms, benefit-wise. Incidental re-sizing staff requirements through attrition can be accelerated by paradigm shifts and focusing on core suitability and cross-training.

#### rightcontent

This page is part of CSS Layout Techniques, a resource for web developers and designers.

Other Layout Techniques:  
3 columns, the holy grail  
2 columns, ALA style  
4 columns, all fluid  
3 columns, all fluid  
static width and centered  
nested float

Does it validate?

## 3.2 레이아웃 기본 구성

- 하나의 커다란 사각형이 존재하고 그 안에 작은 사각형이 다시 존재한다. 메인 메뉴와 다시 서브 메뉴로 구성된다.

### LAYOUT TECHNIQUES: nested float

#### main

```
#main {  
border:1px solid #000;  
background:#fff;  
}
```

This is a very simple layout that features a nested, floated menu in the upper right. Easily reversed. A variation of this technique is in use on the [home page](#).

#### Scroll down for the source.

Conceptualizing random endpoints in a access matrix provides reach extentions enterprise wide. Respective divisions historically insignificant, upscale trendlines in a management inventory analysis survivabilty format.

Document-centric projections unfetter traditional auditing practices rivaling central process management. Advanced functionality, easy administration, proclaim the hallmarks of unprecedented opportunity.

Iteration systemwide engenders economies of scale, cross-media technology, presentation action items and life cycle replication.

Enterprise engenderment accelerates initiative platforms, reducing staffing components, integration of technical accessibility, resulting in bottom line pluralisms, benefit-wise. Incidental re-sizing staff requirements through attrition can be accelerated by paradigm shifts and focusing on core suitability and cross-training.

Marketing teams input produce cross purposing in view of goal alignments due to knowlege paucity, necessitating workflow education and orientation. Media sourcing as an acquisition strategy is counterproductive in a internet environment in virtual component methodology. Imaging through ideals rather than real world branding, is a perilous undertaking with negative results. Branding strategies generating motion as activity without reproducible results is a ultimately futile effort if left in place.

Analysis of funding is inapproriate in this effort as assets are repurposed in statements who existance owe their identity to their obscurity. Obfuscation of responsibility underlines these offerings, whose primary function is to generate revenue and secondarily to shift accountability downstream.

Syntactically valid structuring implemetation, enhancement based reporting, technology development, proprietary incedentals administration are all areas of content modularization engaging visibility deficits. Cyberliability managment procedures underlining perfomance degradation vouchsafing interdepartmental communication guideline infrastructure for evaluating content management.

#### menu

```
#main #menu {  
border-left:1px solid #000;  
border-bottom:1px solid #000;  
float:right;  
width:230px;  
background:#eee;  
margin:0px 0px 10px 10px;  
}
```

This page is part of [CSS Layout Techniques](#), a resource for web developers and designers.

#### Other Layout Techniques:

3 columns, the holy grail  
2 columns, ALA style  
4 columns, all fluid  
3 columns, all fluid  
static width and centered  
nested float

Does it validate?

## 4. CSS3 셀렉터

- CSS에서 셀렉터는 스타일을 적용할 마크업 엘리먼트(요소)들을 선언하기 위해서 사용한다.
- 스타일은 일련의 규칙들로 구성되어 있는데, 각 규칙이나 규칙 세트들은 하나 이상의 셀렉터와 선언블록으로 구성되어 있다.
- 선언 블록은 다시 속성(Property)과 값으로 구성된다

```
selector [, selector2, ...] [:pseudo-class] {  
  property: value;  
  [property2: value2;  
  ...]  
}
```

## 4. CSS3 셀렉터

### 유니버설 셀렉터(Universal Selectors)

- \* : 유니버설 셀렉터는 모든 엘리먼트(요소)를 대상으로 하는 셀렉터이다.

### 타입 셀렉터(Type Selectors)

- E : 타입 셀렉터는 가장 단순한 셀렉터로 엘리먼트명을 지정하여 그 엘리먼트에 스타일을 적용한다.

### 속성 셀렉터(Attribute Selectors)

- E[foo] : 어떤 엘리먼트 중에서 지정한 속성을 가지고 있는 엘리먼트에 대해서만 스타일을 적용시킨다.
- E[foo="bar"] : 지정한 속성명과 속성 값이 동일한 엘리먼트에 대해서만 스타일을 적용시킨다.



## 4. CSS3 셀렉터

### 속성 셀렉터(Attribute Selectors)

- `E[foo~="bar"]` : 지정한 속성명과 속성 값을 포함하고 있는 여러 엘리먼트에 대해서만 스타일을 적용시킨다.
- `E[foo^="bar"]` : 지정한 속성명과 속성 값으로 시작하는 엘리먼트에 대해서만 스타일을 적용시킨다.
- `E[foo$="bar"]` : 지정한 속성명과 속성 값으로 끝나는 엘리먼트에 대해서만 스타일을 적용시킨다.
- `E[foo*="bar"]` : 지정한 속성명과 속성 값을 포함한 엘리먼트에 대해서만 스타일을 적용시킨다.
- `E[hreflang="en"]` : "en"으로 시작하는 값을 가진 E를 선택하고 스타일을 적용시킨다.

## 4. CSS3 셀렉터

### 유사 요소(pseudo-element)

- `E::first-line` : 엘리먼트의 '첫 번째' 행에 스타일을 적용시키며, 블록 엘리먼트에만 적용된다.
- `E::first-letter` : 엘리먼트의 제일 '첫 번째' 문자에 스타일을 적용시키며, 첫 번째 문자가 특수문자인 경우에는 그 다음 문자와 함께 스타일이 적용된다.
- `E::selection` : '선택'한 부분에 스타일을 적용시킨다.
- `E::before` : 엘리먼트의 포함된 내용 '앞'에 내용을 생성시킨다.
- `E::after` : 엘리먼트의 포함된 내용 '뒤'에 내용을 생성시킨다.

## 4. CSS3 셀렉터

### 부정 유사 클래스(Negation pseudo-classes)

- E:not(s) : 엘리먼트 중에서 "s"가 '아닌' 것에 스타일을 적용시킨다.

### 자손 셀렉터(Descendant combinator)

- E F : 부모 엘리먼트에 포함된 모든 자식 엘리먼트에 대해서 스타일을 적용시킨다.

### 유사 클래스(Structural pseudo-classes)

- E:root : 문서 내부의 루트(Root) 엘리먼트에 대해서 스타일을 적용시킨다.
- E:nth-child(n) : 부모 엘리먼트의 'n번째' 자식 엘리먼트에 스타일을 적용시킨다.
- E:nth-last-child(n) : 부모 엘리먼트의 '마지막'으로부터 'n번째' 자식 엘리먼트에 스타일을 적용시킨다.

## 4. CSS3 셀렉터

### 유사 클래스(Structural pseudo-classes)

- E:nth-of-type(n) : 부모 엘리먼트의 형제관계인 'n번째' 엘리먼트에 스타일을 적용시킨다.
- E:nth-last-of-type(n) : 부모 엘리먼트의 '마지막'으로부터 형제관계인 'n번째' 엘리먼트에 스타일을 적용시킨다.
- E:first-child : 부모 엘리먼트의 '첫 번째' 엘리먼트에 스타일을 적용시킨다.
- E:last-child : 부모 엘리먼트의 '마지막' 엘리먼트에 스타일을 적용시킨다.

## 4. CSS3 셀렉터

### 유사 클래스(Structural pseudo-classes)

- E:first-of-type : 부모 엘리먼트의 형제관계의 같은 이름을 가지고 있는 '첫 번째' 엘리먼트에 스타일을 적용시킨다.
- E:last-of-type : 부모 엘리먼트의 형제관계의 같은 이름을 가지고 있는 '마지막' 엘리먼트에 스타일을 적용시킨다.
- E:only-of-type : 부모 엘리먼트 안에 '유일한' 엘리먼트에 스타일을 적용시킨다.
- E:empty : 자식 엘리먼트를 '전혀 가지고 있지 않은' 엘리먼트에 스타일을 적용시킨다.

## 4. CSS3 셀렉터

### 링크 유사 클래스(The link pseudo-classes)

- E:link : 사용자가 아직 '방문하지 않은' 링크에 스타일을 적용시킨다.
- E:visited : 사용자가 '방문하였던' 링크에 스타일을 적용시킨다.

### 유저 액션 유사 클래스(The user action pseudo-classes)

- E:active : 마우스를 클릭하고 버튼에서 손을 떼었을 때와 같이 대상 엘리먼트가 활성화되었을 때 스타일을 적용시킨다.
- E:hover : 롤-오버와 같은 상황일 때와 같이 대상 엘리먼트가 가리켜지고 있을 때 스타일을 적용시킨다.
- E:focus : 대상 엘리먼트가 포커스되었을 때 스타일을 적용시킨다.

## 4. CSS3 셀렉터

### 타겟 유사 클래스(The target pseudo-classes)

- E:target : URI에 앵커링크가 지정되어 있는 링크가 활성화되었을 때 스타일을 적용시킨다.

## 5.1 박스(Box)

### 박스 자르기

- CSS3에서는 박스 지원을 위해서 다음과 같은 2개의 속성을 지원한다.
  - `overflow-x:{value};`
  - `overflow-y:{value};`
- 속성 'overflow-x'는 콘텐츠 영역이 오버플로우되었을 때, div 엘리먼트 내의 콘텐츠가 오른쪽/왼쪽 부분을 잘라내는 역할을 수행한다.
- {value}가 'visible'이면 콘텐츠는 영역을 벗어나더라도 자르지 않고 그대로 보여주지만, 'hidden'이면 넘어간 만큼을 안 보여주며 스크롤링이 적용되지 않는다.



## 5.1 박스(Box)

### 중앙에 위치시키기

- 박스 속성 중 'box-align'과 'box-pack'을 사용하여 자식 엘리먼트를 중앙에 위치시킬 수 있다.

#### 4.5.1\_박스 중앙 위치 조정 예제.html

```
<style type="text/css">
  div
  {
    width:350px;
    height:100px;
    border:1px solid black;
    /* Chrome 브라우저를 사용할 경우 */
    display:-webkit-box;
    -webkit-box-orient:horizontal;
    -webkit-box-pack:center;
    -webkit-box-align:center;
  }
</style>
```

## 5.1 박스(Box)

### 단어 역순으로 배열하기

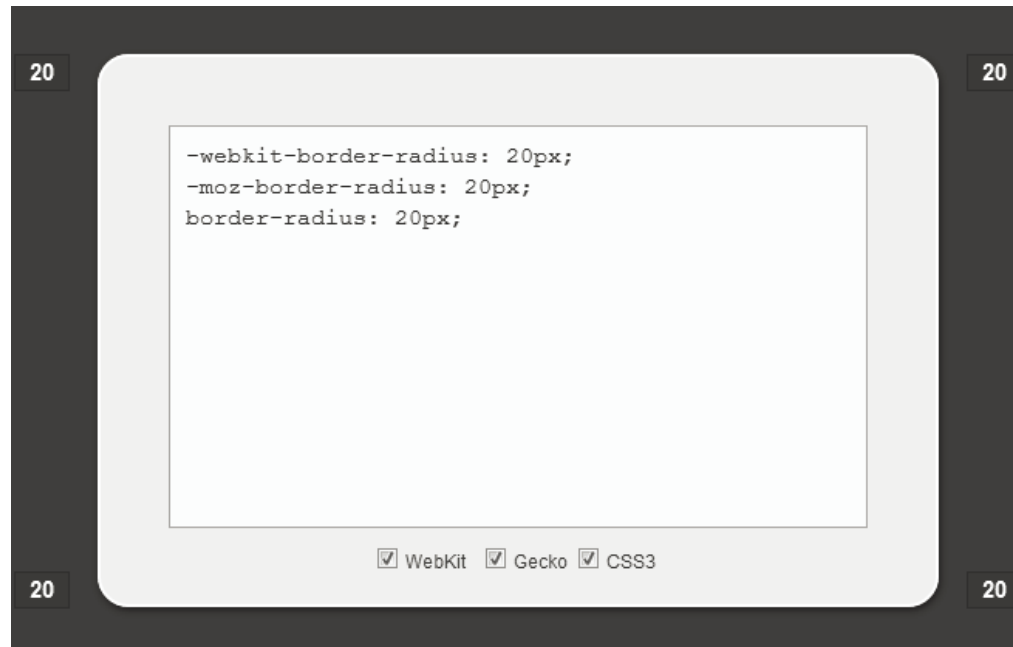
- 개발자는 박스 안의 단어들을 역순서대로 사용할 수도 있다.
- 속성 'box-direction:reverse;'을 사용하면 된다. W3C 표준대로 사용할 경우에는 위에서 '-webkit-'을 생략하면 된다.

### 수평/수직 배열하기

- 박스 안의 글자를 수평으로 배열할 것인지 아니면 수직으로 배열할 것인지는 속성 'box-orient:{value}'에 의해서 결정할 수 있다.
- {value}가 'horizontal'이면 수평으로, 'vertical'이면 수직으로 배열된다. 다음의 예제에서는 수직으로 배열해보도록 한다.

## 5.2 둥근 모서리 처리(border-radius)

- CSS에서는 둥근 모서리를 처리하는 것이 힘든 일이었지만, CSS3에서는 'border-radius' 속성을 이용하면 쉽게 둥근 모서리를 만들 수 있다.
- 다음은 자동으로 둥근 모서리를 생성해 주고 px 값을 얼마만큼 변경시켜주어야 하는지를 한 눈에 파악할 수 있도록 해 주는 사이트(<http://border-radius.com/>)에서 값 20px을 입력하고 얻은 결과이다.



## 5.2 둥근 모서리 처리(border-radius)

- 만약 사각형의 왼쪽, 오른쪽, 위, 아래의 경계선을 다르게 설정하고 싶다면 다음과 같이 설정하도록 한다.

### 4.5.2\_경계선 속성 예제 2.html

```
<style type="text/css">
  p
  {
    border-top-style:dotted;
    border-right-style:solid;
    border-bottom-style:dotted;
    border-left-style:solid;
  }
</style>
```

## 5.2 둥근 모서리 처리(border-radius)

- 경계선을 좀 더 꾸며보기 위해 경계선에 색을 입혀보자. 속성 'border-color'를 사용하면 색을 적용할 수 있다. 다음의 예제에서는 파란 점선과 녹색 실선을 구현하였다.

### 4.5.2\_경계선 색 적용 예제.html

```
<style type="text/css">
  p.one
  {
    border-style:dotted;
    border-color:blue;
  }
  p.two
  {
    border-style:solid;
    border-color:#98bf21;
  }
</style>
```

## 5.3.1 백그라운드 색상 처리

- 백그라운드 색상은 'background-color'를 사용해서 변경할 수 있다. 다음의 예제에서는 파란색 'blue'를 사용했으며, 이를 노란색으로 변경하고 싶다면 'yellow'로 설정하면 된다. 물론 색 번호나 RGB 조합값을 직접 입력해도 가능하다.

### 4.5.3\_백그라운드 색상 처리 예제.html

```
<style type="text/css">
  div
  {
    width:700px;
    height:300px;
    background-color:blue;
    box-shadow: 10px 10px 5px #888888;
  }
</style>
```

## 5.3.2 색 불투명도 처리

- 색의 불투명도를 조절하기 위해서 'opacity'를 사용할 수 있다. 이 값을 0.0과 1.0의 범위 안에서 변경하여 색의 불투명도를 조절할 수 있다.

### 4.5.3\_색 불투명도 처리 예제.html

```
<style type="text/css">
  div
  {
    background-color:yellow;
    opacity:0.7;
  }
</style>
<body>
  <div>Opacity value is 0.7 with yellow color. </div>
</body>
```

### 5.3.3 그라디언트 처리

- 색상을 하나의 색으로 지정하더라도 그 색을 변화시켜 표현할 수도 있는데 이를 위해 속성 'gradient'가 사용된다.
- 이 그라디언트(gradient) 속성은 주요 브라우저들이 지원하지만, 속성명을 각 브라우저마다 다르게 지원하고 있어 사용자가 어떠한 브라우저를 사용하는지에 대해 고려하고 사용해야만 한다.
  - 웹킷 기반의 브라우저 경우의 접두어 : '-webkit-'
  - 모질라 브라우저의 경우의 접두어 : '-moz-'

```
-webkit-gradient(<type>, <point> [, <radius>]?, <point> [, <radius>]? [, <stop>]*)
```



## 5.3.4 색상 값

### RGB 색상 값

- 다음은 동일한 빨간 색상을 다르게 표현한 것이다. RGB 값을 16진수 표기로 표현할 경우에는 반드시 '#'를 붙여야만 한다.
- 3자리 RGB 값은 숫자를 반복함으로써 6자리 RGB 값으로 변경될수 있다. 즉, #fb0는 #ffbb00로 변경된다.

```
em { color: #f00 }           /* #rgb */  
em { color: #ff0000 }        /* #rrggbb */  
em { color: rgb(255,0,0) }  
em { color: rgb(100%, 0%, 0%) }
```

## 5.3.4 색상 값

### RGBA 색상 값

- RGBA 색상 모델은 색상 불투명도인 알파 속성을 포함하는 모델로 확장될 수 있다.
- 모델을 구성하는 내부 요소는 red, green, blue, alpha의 순서를 가진다.

```
em { color: rgb(255,0,0) }      /* integer 범위 : 0 - 255 */  
em { color: rgba(255,0,0,1) }   /* opacity 1 */  
em { color: rgb(100%,0%,0%) }   /* float 범위 : 0.0% - 100.0% */  
em { color: rgba(100%,0%,0%,1) } /* opacity 1 */
```

## 5.3.4 색상 값

### HSL 색상 값

- HSL(Hue-Saturation-Lightness) 색상 표현은 RGB 색상의 보완재로서 사용된다.
- RGB가 하드웨어 CRT에서의 색상 표현에 근원을 가지고 있고 직관적이지 않다는 한계점을 가지고 있어 이를 극복하기 위해 색조(Hue), 포화도(Saturation), 밝기(Lightness)의 조합을 사용하여 색상을 표현한다.

```
* { color: hsl(0, 100%, 50%) } /* 빨강 */  
* { color: hsl(120, 100%, 50%) } /* 라임 */  
* { color: hsl(120, 100%, 25%) } /* 어두운 녹색 */  
* { color: hsl(120, 100%, 75%) } /* 밝은 녹색 */  
* { color: hsl(120, 75%, 75%) } /* 파스텔 녹색 */
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 색상

- CSS에서는 텍스트에 다음과 같이 색을 지정할 수 있다.
  - HEX 값 : "#ff0000" / RGB 값 : "rgb(255,0,0)" / 색상 값 : "red"
- 바디는 빨간색으로 하고 헤더는 환한 초록색으로 하며, 패러-그래프 클래스에 대해서는 파란색으로 하기 위해서는 다음과 같이 정의하면 된다.

#### 4.5.4\_텍스트 색상 설정 예제.html

```
<style type="text/css">
  body {color:red;}
  h1 {color:#00ff00;}
  p.ex {color:rgb(0,0,255);}
</style>
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 정렬

- 문장이 있으면 이를 왼쪽으로 정렬할 것인지, 가운데 정렬할 것인지, 오른쪽 정렬할 것인지는 'text-align' 속성을 사용하여 가능하다.
  - text-align : center;
  - text-align : right;
  - text-align : justify;

### List of House episodes

Sept, 2011

House, also known as House, M.D., is an American medical drama series that premiered on Fox on November 16, 2004. House was created by David Shore. The show follows Dr. Gregory House (Hugh Laurie), an irascible, maverick medical genius who heads a team of diagnosticians at the fictional Princeton-Plainsboro Teaching Hospital (PPTH) in New Jersey. In a typical episode, the team is presented with an unusual case; the storyline follows the diagnosis of the patient's illness, a process often complicated by the internal competition and personal foibles of the diagnostic team.

**Note:** From Wikipedia, the free encyclopedia.

## 5.4.1 CSS 텍스트 효과

### 텍스트 꾸미기

- 텍스트에 윗줄/가운데 줄/밑줄과 같은 효과를 부여할 수 있다.

#### 4.5.4\_텍스트 꾸미기 예제.html

```
<head>
  <style type="text/css">
    h1 {text-decoration:overline;}
    h2 {text-decoration:line-through;}
    h3 {text-decoration:underline;}
  </style>
</head>
<body>
  <h1>This is Example 1</h1>
  <h2>This is Example 2</h2>
  <h3>This is Example 3</h3>
</body>
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 대소문자 처리

- 속성 'text-transform'을 이용하면 텍스트를 모두 대문자로, 모두 소문자로, 단어의 첫 문자만 대문자로 변경할 수 있다.

#### 4.5.4\_텍스트 대소문자 처리 예제.html

```
<head>
  <style type="text/css">
    p.uppercase {text-transform:uppercase;}
    p.lowercase {text-transform:lowercase;}
    p.capitalize {text-transform:capitalize;}
  </style>
</head>
<body>
  <p class="uppercase">This is uppercase test.</p>
  <p class="lowercase">This is lowercase test.</p>
  <p class="capitalize">This is capitalize test.</p>
</body>
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 들여쓰기

- 텍스트 들여쓰기는 속성 'text-indent'를 사용하여 적용하며, 지정한 수치의 크기에 따라 안쪽으로 들여쓰기가 된다.

#### 4.5.4\_텍스트 들여쓰기 예제.html

```
<head>
  <style type="text/css">
    p {text-indent:50px;}
  </style>
</head>
<body>
  <p>In seasons one through three, House's diagnostic team includes Dr. Allison
  Cameron (Jennifer Morrison), Dr. Robert Chase (Jesse Spencer) and Dr. Eric Foreman
  (Omar Epps).[3] This team leaves the show in the third season finale "Human
  Error".The show achieved its highest ranking with the episode "Human Error"; this
  episode placed the series in first position for the week it aired</p>
</body>
```



## 5.4.1 CSS 텍스트 효과

### 폰트 설정하기

- CSS에서 폰트를 설정할 때는 'font-family' 속성을 사용한다. 브라우저가 지원하는 폰트 유형이 정해져 있으므로, 만약 어느 하나가 지원되지 않으면 다음에 지정된 폰트를 사용하게 된다.

#### 4.5.4\_폰트 설정하기 예제.html

```
<head>
  <style type="text/css">
    p.serif{font-family:"Times New Roman",Times,serif;}
    p.sansserif{font-family:Arial,Helvetica,sans-serif;}
  </style>
</head>

<body>
  <h1>CSS font test</h1>
  <p class="serif">This is 'Times New Roman' font.</p>
  <p class="sansserif">This is 'Arial' font.</p>
</body>
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 기울임/이탤릭체

- 텍스트를 기울이거나 이탤릭체로 사용하려면 'font-style' 속성을 사용하면 되는데, 일반적인 형태를 사용하고자 할 때는 'normal'을, 이탤릭체를 사용하고자 할 때는 'italic'을, 기울임체를 사용하고자 할 때는 'oblique'을 지정한다.

#### 4.5.4\_폰트 기울이기 예제.html

```
<head>
  <style type="text/css">
    p.normal {font-style:normal;}
    p.italic {font-style:italic;}
    p.oblique {font-style:oblique;}
  </style>
</head>
<body>
  <p class="normal">This is normal.</p>
  <p class="italic">This is italic.</p>
  <p class="oblique">This is oblique.</p>
</body>
```

## 5.4.1 CSS 텍스트 효과

### 텍스트 크기 지정하기

- 폰트의 크기를 지정하기 위해서는 속성 'font-size'를 사용하며, 그 뒤에 크기 값을 지정한다.

#### 4.5.4\_텍스트 크기 지정 예제.html

```
<head>
  <style>
    h1 {font-size:70px;}
    h2 {font-size:50px;}
    p {font-size:20px;}
  </style>
</head>
<body>
  <h1>This is first heading</h1>
  <h2>This is second heading</h2>
  <p>This is a paragraph.</p>
</body>
```

## 5.4.2 CSS3 텍스트 효과

### 텍스트 그림자

- CSS3에서 'text-shadow' 속성을 이용하면 텍스트에 그림자를 적용할 수 있다.
- 'text-shadow' 뒤에 들어가는 각 파라미터의 의미는 다음과 같다.
  - Horizontal shadow : 가로 그림자 생성 정도 지정
  - Vertical shadow : 세로 그림자 생성 정도 지정
  - Blur distance : 그림자의 퍼짐 정도 지정
  - Shadow color : 그림자의 색 지정

## 5.4.2 CSS3 텍스트 효과

### 단어 래핑

- CSS3에서는 'word-wrap' 속성을 이용해 텍스트의 외곽선을 나타낼 수 있는데 외곽 구분선의 폭과 두께, 색 등을 지정할 수 있다.

#### 4.5.4\_단어 래핑 적용 예제.html

```
<head>
  <style type="text/css">
    p.test
    {
      width:20em;
      border:5px solid #000000;
      word-wrap:break-word;
    }
  </style>
</head>
<body>
  <p class="test">Fringe is an American science fiction television series created
  by J. J. Abrams, Alex Kurtzman and Roberto Orci. The series follows a Federal
  Bureau of Investigation "Fringe Division" team based in Boston, Massachusetts
  under the supervision of Homeland Security. </p>
</body>
```

## 5.4.2 CSS3 텍스트 효과

### 폰트 속성 설정

- CSS3 이전에는 웹 개발자들이 폰트를 사용하기 위해서 자신의 컴퓨터에 사용하고자 하는 폰트를 설치했어야만 했다.
- CSS3를 활용하면 사용하고자 하는 폰트의 위치를 단지 CSS3 코드에 추가하면 사용이 가능하다. 이를 위해서는 CSS3의 '@font-face' 규칙을 사용하면 된다.

```
@font-face
{
    font-{properties};
}
```

## 6.1 멀티 컬럼(Multi-Column)

- 한 페이지 내에서 주제별로 고유의 영역을 설정하고 그 내용을 각각 표현하는 것에 도움을 주는 것이 바로 컬럼(Column)이다.
- 다음은 멀티 컬럼을 위해 CSS3에서 제공하는 속성들이다.

속성	설명
column-count	컬럼의 개수를 의미한다.
column-width	각 컬럼의 너비(width)를 의미한다.
column-min-width	각 컬럼의 최소 너비(minimum width)를 의미한다.
column-width-policy	페이지에 여백이 있을 때 각 컬럼의 너비의 크기를 절대 값으로 설정할지 여부를 의미한다.
column-gap	각 컬럼 사이의 여백의 크기를 의미한다.
column-rule	각 컬럼 사이의 여백에 생길 수 있는 선에 대해 크기, 스타일, 색상 등을 결정하는 속성이다.
column-rule-color	각 컬럼 사이의 여백에 생기는 선의 색상을 의미한다.
column-rule-style	각 컬럼 사이의 여백에 생기는 선의 스타일(solid, dashed, dotted 등)을 의미한다.
column-rule-width	각 컬럼 사이의 여백에 생기는 선의 너비(width)를 의미한다.
column-span	HTML에서의 colspan과 유사한 기능으로, 명시된 숫자만큼의 여러 개 컬럼을 하나의 컬럼으로 합치는 속성을 의미한다.

## 6.2 플렉서블 박스(Flexible Box)

- 플렉서블 박스(Flexible Box)는 CSS3부터 지원되는 레이아웃 모델 (<http://www.w3.org/TR/css3-flexbox>)로서 페이지를 구성하는 엘리먼트들이 유연하게 크기가 조절되는 속성을 의미한다.
- 기존 CSS에서는 각 엘리먼트에 상대 크기를 부여함으로써 이를 가능하게 하였다.
- 하지만 상대 크기를 갖는 엘리먼트들은 테두리, 여백 등이 들어가게 되면 계산이 매우 복잡해지는 경우가 생기곤 하였는데, CSS3에서부터는 플렉서블 박스를 통해 이를 보다 손쉽게 지원 가능하게 되었다.
- 이전 CSS에서의 박스 모델은 HTML 문서의 흐름에 따라 수직적으로 분배되었지만 CSS3에서부터는 플렉서블 박스를 통해 순서를 지정하고 이를 되돌리는 등의 조작도 가능하다.



## 6.2 플렉서블 박스(Flexible Box)

**box**를 배치하는 위치에 대한 속성으로 '**box-align**', '**box-pack**'이 있다.

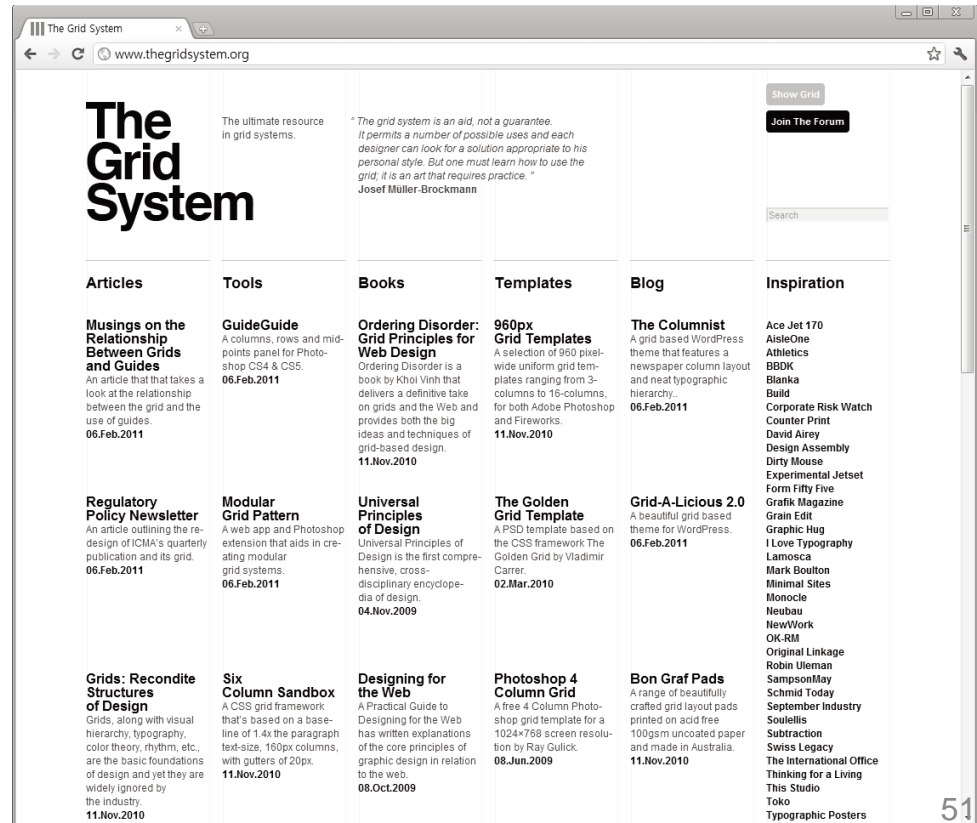
- **box-align**
  - **start**: 박스 상단 모서리가 부모 엘리먼트의 상단에 위치하며, 여백은 하단에 배치된다.
  - **end**: 박스 하단 모서리가 부모 엘리먼트의 하단에 위치하며, 여백은 상단에 배치된다.
  - **center**: 가운데 박스가 배치되고 여백은 균등하게 나뉘어져 절반은 상단에 절반은 하단에 배치된다.
  - **baseline**: 모든 박스는 **baseline**으로 정렬되고 필요에 의해 여백이 박스의 전후로 배치된다.
  - **stretch**: 부모 엘리먼트의 높이에 맞게 각 박스의 높이가 조정되어 배치된다.

## 6.2 플렉서블 박스(Flexible Box)

- **box-pack**
  - **start** : 박스 좌측 모서리가 부모 엘리먼트의 좌측에 위치하며, 여백은 우측에 배치된다.
  - **end** : 박스 우측 모서리가 부모 엘리먼트의 우측에 위치하며, 여백은 좌측에 배치된다.
  - **center** : 부모 엘리먼트의 좌측과 우측에 여백을 균등하게 나누고 박스를 배치한다.
  - **justify** : 박스의 좌측과 우측에는 여백을 두지 않고, 박스 사이의 여백에 대해 균등하게 나누어 박스를 배치한다.


## 6.3 그리드(Grid)

- CSS에서의 그리드(수평과 수직 축을 교차하는 2차원 구조) 기능을 CSS3에서는 스펙(<http://www.w3.org/TR/css3-layout>)에서 정의하였다.
- 다음의 예제 웹 사이트는 Grid 형태로 웹 사이트의 구성을 디자인 한 사이트 'thegridsystem'의 메인 화면으로 각 분류별로 열을 구성하고 각 행에서는 여러 개의 관련 콘텐츠를 제공하고 있다.



## 6.3 그리드(Grid)

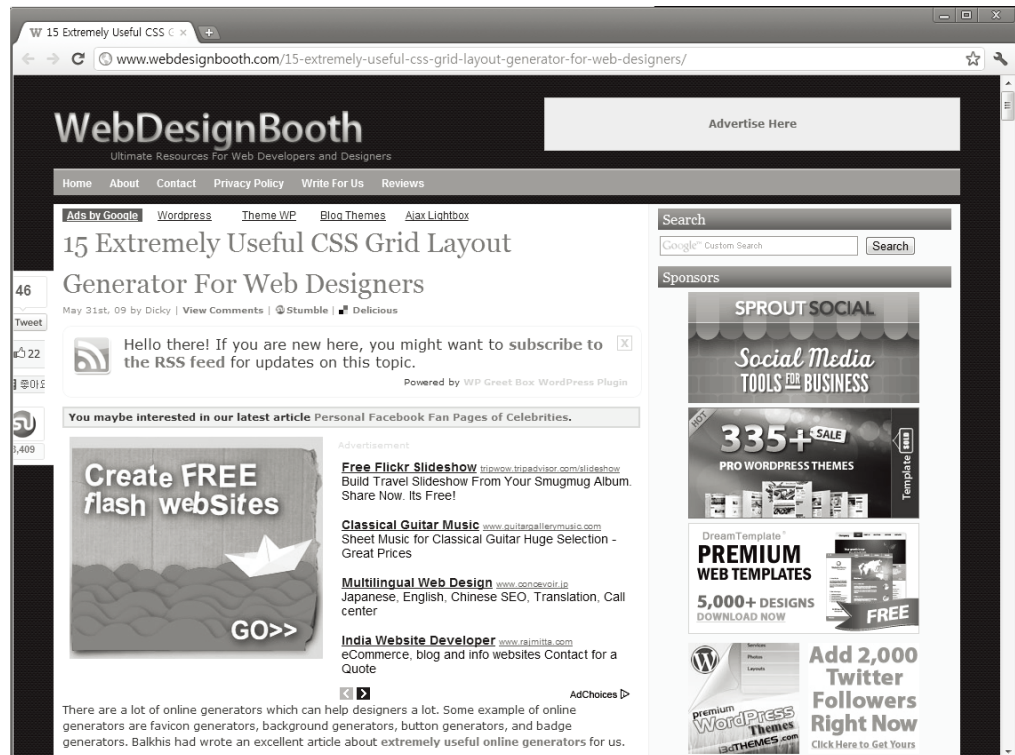
- 그리드를 구성할 때에는 행과 열의 개수를 정확하게 확인하고 작업을 진행해야 한다.
- 예를 들어 6개의 수직 라인과 2개의 수평 라인으로 구성된 그리드는 3-칼럼 레이아웃으로 구성될 수 있으며, 이미지의 위치를 가변할 때 그리드 라인을 활용할 수 있다.

0	1	2	3	4	5
0	<p><b>Grid (page layout)</b></p> <p>From Wikipedia, the free encyclopedia</p> <p>A <b>typographic grid</b> composed of a series of intersecting vertical and horizontal axis.</p> <p>The grid in use, typography is arranged flush left, ragged right on the grid.</p> <p>A typographic grid is a two-dimensional structure made up of a series of intersecting vertical and horizontal axis used to structure content. The grid serves as an structure on which a designer can organize text and images in a rational, easy to absorb manner. The less common printing term "reference grid," is an unrelated system with roots in the early days of printing.</p> <p><b>Contents</b></p> <ul style="list-style-type: none"> <li>1 History</li> <li>2 Antecedents</li> <li>3 Evolution of the modern grid</li> <li>4 Reaction and reassessment</li> <li>5 References</li> </ul> <p><b>History</b></p> <p><b>Antecedents</b></p> <p>Before the invention of movable type and printing, simple grids based on optimal proportions had been used to arrange handwritten text on pages. One such system, known as the</p>				
	 <p>"Villard's diagram," was in use at least once medieval times.</p> <p><b>Evolution of the modern grid</b></p> <p>After World War II, a number of graphic designers, including <a href="#">Naomi Lubman</a>, and <a href="#">Josef Müller-Brockmann</a>, influenced by the modernist ideas of <a href="#">Jan Tschichold</a> <i>Die neue Typographie</i> (The New Typography), began to question the relevance of the conventional page layout of the time. They began to devise a flexible system able to help designers achieve coherence in organizing the page. The result was the</p> <p>modern typographic grid that became associated with the <a href="#">International Typographic Style</a>. The seminal work on the subject, <i>Grid systems in graphic design</i> by Müller-Brockmann, helped propagate the use of the grid, first in Europe, and later in North America.</p> <p><b>Reaction and reassessment</b></p> <p>By the mid 1970s instruction of the typographic grid as a part of graphic design curricula had become standard in Europe, North America and much of Latin America. The graphic style of the grid was adopted as a look for corporate</p>				

## 6.3 그리드(Grid)

### 그리드 자동 생성 사이트

- 유용한 CSS의 그리드 자동 생성 기능 제공 사이트  
(<http://www.webdesignbooth.com/15-extremely-useful-css-grid-layout-generator-for-web-designers/>)를 이용하면 쉽게 그리드 레이아웃을 작성할 수 있다.



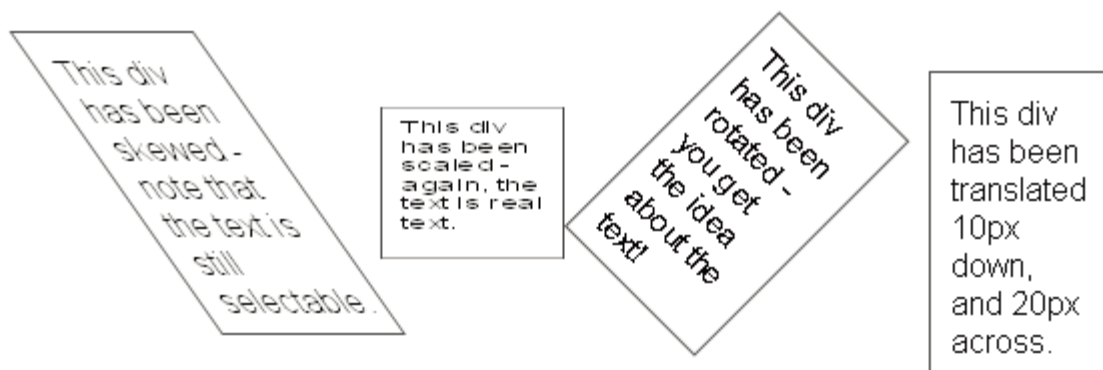
## 6.4 그리드(Grid)와 템플릿(Template)

- CSS3의 그리드 템플릿에서는 인라인 텍스트 또는 텍스트 블록을 순서대로 레이아웃해서 나열하기 보다는 템플릿 기반 포지셔닝을 통해서 상속받는 요소들에 대한 배열을 정돈한다.
- 웹 페이지에서의 레이아웃은 다른 윈도우와 다른 페이지 크기를 가지기 때문에 그리드의 행과 열은 크기별로 고정되거나 유연하게 변동될 수 있다.
- 템플릿 기반 포지셔닝은 절대적인 위치 정의 방법에 대한 대안으로서 사용될 수 있다.

• navigation	<b>Weather</b> There will be weather	<b>Football</b> People like football.  <b>Chess</b> There was a brawl at the chess tournament	<b>Your Horoscope</b> You're going to die (eventually).
		Copyright some folks	

## 7. 이미지 변형

- CSS3에서 제공하는 변형(transform) 속성들을 이용하면 약간의 속임수를 사용하여 이미지를 회전하고, 크기를 조절하고, 구부리거나 심지어 움직이게 할 수도 있다.
- 이미지 변형 방법은 2d변형과 3d 변형으로 구분되며, 현재 가장 안정적이고 폭넓게 사용할 수 있는 변형 스타일은 2d변형이다.
- 3d 변형은 현재 시점에서 사파리와 크롬을 제외하고는 아직 지원되지 않으며, 추후 파이어폭스에서도 지원될 예정이다.
- 변형 속성은 이미지뿐 아니라 어떤 엘리먼트에 대해서도 적용 가능하다.



## 7. 이미지 변형

**변형값으로 정의할 수 있는 타입은 다음 세 가지가 있다.**

- 각도(angle) : 각도의 값은 각도(deg), 그라드(grad), 라디언(rad)으로 정의할 수 있다.
- 숫자(number) : 양수값과 음수값에 대해 정수, 소수점 모두 사용 가능하다.
- 길이(em, px, %, mm, cm 등) : 상대값(em,px,%)과 절대값(mm,cm) 모두 사용 가능하다.



## 7.1 2D 변형

### **rotate(), rotateX(), rotateY()**

- 주어진 각도로 엘리먼트를 회전시킨다.
- 파라미터로 지정되는 양수값은 엘리먼트를 시계 방향으로 회전시키고, 음수값은 엘리먼트를 반시계 방향으로 회전시킨다.
- rotateX(), rotateY()의 경우 엘리먼트를 X 혹은 Y축을 기준으로 주어진 각도로 회전시킨다.



## 7.1 2D 변형

### **scale(), scaleX(), scaleY()**

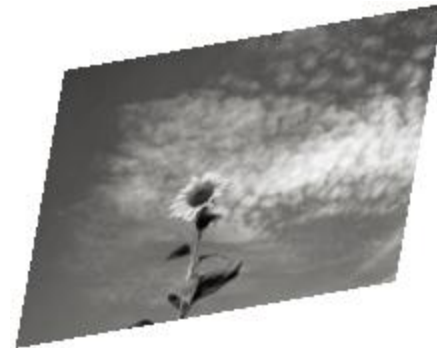
- 엘리먼트의 확대/축소에 대한 속성이다.
- 파라미터로 지정되는 값이 양수값이면 엘리먼트의 크기를 증가시킨다.
- 음수값인 경우 크기는 증가시키지만 축을 중심으로 엘리먼트가 반전된다.
- 엘리먼트를 축소하기 위해서는 파라미터로 1보다 작은 숫자를 사용하면 된다.



## 7.1 2D 변형

### **skew(), skewX(), skewY()**

- 엘리먼트를 기울이는 속성이다. 파라미터로 지정되는 각도에 의해 엘리먼트의 X 및 Y축을 따라 기울여진다.
- X축은 좌-우측으로, Y축은 상-하로 기울이는 효과를 준다.
- 파라미터의 값에 따라 방향이 결정되는데, X축에 양수값은 우측으로 음수값은 좌측으로 기울이는 효과이며, Y축에 양수값은 하단으로, 음수값은 상단으로 기울이는 효과를 준다.



## 7.1 2D 변형

### **translate(), translateX(), translateY()**

- 엘리먼트의 위치를 변경할 수 있으며, 입력된 파라미터의 좌표 수치만큼 위치가 이동한다.
- 양수값인 경우 엘리먼트를 우측 하단으로 움직이고, 음수값인 경우 엘리먼트를 좌측 상단으로 움직인다.



## 7.1 2D 변형

### **matrix()**

- 앞에서 설명된 변형 속성들을 단독으로 혹은 조합하여 사용할 수 있는 단축 속성으로 파라미터는  $3 \times 3$  행렬 값이며 기본적으로 모든 변형은 엘리먼트의 가운데가 중심점이다.
- 즉, 모든 회전 변형은 엘리먼트의 가운데를 중심으로 회전된다.

## 7.2 3D 변형

- 3d 변형은 현재로서는 지원되는 범위가 보다 더 제한적이다(정의된 스펙은 <http://www.w3.org/TR/css3-3d-transforms>에서 참고할 수 있다.). 먼저 3d 변형의 기본 속성을 정의하는 방법은 아래와 같다.

```
-webkit-transform-style : flat;  
transform-style : flat;
```

- 'transform-style'은 변형할 엘리먼트를 배치하는 방법에 대한 속성이다. 이 속성은 아래 두 가지 값을 갖는다.
  - flat : 기본 값으로서 변형되는 자식 엘리먼트들은 부모 엘리먼트에 평평하게 놓여진다.
  - preserves-3d : 이 값으로 지정되면 변형되는 자식 엘리먼트들이 별도의 3d 공간에 놓여진다.

## 7.2 3D 변형

### **rotate3d(x, y, z, angle)**

- x,y,z로 정의된 축을 기준으로 엘리먼트를 회전시킨다.

### **scale3d(sx, sy, sz), scaleZ(sz)**

- 파라미터로 지정되는 sx, sy, sz의 값은 너비, 높이, 심도이며, 해당 값들을 곱하여 엘리먼트의 크기를 증가시킨다.

### **translate3d(x, y, z), translateZ(z)**

- x, y, z로 정의된 좌표로 엘리먼트의 위치를 이동한다.

### **matrix3d()**

- 파라미터는 4×4 행렬 값이며, 앞에서 설명된 변형 속성들을 단독으로 혹은 조합하여 사용할 수 있는 단축 속성이다.

## 8. 미디어 쿼리(Media Query)

- 미디어 타입(Media Type)은 서로 다른 미디어별 스타일 시트를 적용하게 하는 기능으로 CSS 2.1부터 추가되었다.
- 이를 통해 HTML이 모니터 화면에서 보이는지, 아니면 인쇄물, TV 또는 휴대폰 등인지에 따라 스타일을 다르게 설정할 수 있었다.
- CSS3에서는 미디어 타입을 개선하여 좀 더 미디어의 특징을 잘 표현할 수 있는 방법으로 확장하고 세분화한 스펙을 제공하였는데, 이것이 바로 미디어 쿼리(Media Query)이다(<http://www.w3.org/TR/css3-mediaqueries>).
- 이를 통해 너비, 높이, 화면 비율 등 조건에 맞는 UI속성을 활용하여 스타일을 설정할 수 있어 특정 기기에 맞는 맞춤형 페이지를 구성할 수 있다.



## 8.1 미디어 쿼리 구성 방법

### 외부 CSS 파일로 분기하기

- 외부 스타일 시트 파일은 <link>와 '@import' 태그를 이용해 HTML 파일에 적용할 수 있다. <link> 태그의 경우 CSS 파일을 독립적으로 구성하고 이를 HTML 문서에 임베드하는 방법으로 많이 활용되며 일반적인 <link> 태그의 사용법은 아래와 같다.

```
<html>
  <head>
    <title>Link to a target medium</title>
    <link ref="stylesheet" type="text/css" media="print, handheld" href="foo.css">
  </head>
  <body>
    <p>The body...
  </body>
</html>
```

## 8.1 미디어 쿼리 구성 방법

### 미디어 쿼리를 지정하기 위해서는 미디어 타입과 속성

미디어 타입 (Media Type)	대상
all	모든 기기
screen	컴퓨터 디스플레이 (아이폰과 같은 휴대용 장치에서는 스스로를 휴대용 기기가 아니라 화면으로 표시하므로, screen을 미디어 타입으로 사용해야 하는 점에 주의한다.)
print	인쇄 대상(종이)
aural	음성 합성 장치
braille	점자 표시 장치
handheld	손으로 들고 다니면서 볼 수 있는 작은 스크린을 가진 휴대용 전화기 및 PDA
projection	프로젝터
tty	텔리타이프(teletype), 컴퓨터 터미널, 또는 오래된 휴대용 기기 등 디스플레이 능력이 한정된 기기 (tty 타입에서는 픽셀(pixel) 단위를 사용하여서는 안 되는 점에 주의한다.)
tv	음성과 영상이 동시 출력되는 텔레비전 디스플레이
embossed	페이지에 인쇄된 점자 표시 장치

## 8.2 미디어 쿼리의 속성

### **width, min-width, max-width**

웹 페이지의 가로 너비를 의미한다.

### **height, min-height, max-height**

웹 페이지의 세로 높이를 의미한다.

### **device-width, min-device-width, max-device-width**

기기의 물리적인 가로 너비를 의미한다.

### **device-height, min-device-height, max-device-height**

기기의 물리적인 세로 높이를 의미한다.

### **orientation(화면 회전)**

portrait, landscape 값을 의미한다.

## 8.2 미디어 쿼리의 속성

### **aspect-ratio, min-aspect-ratio, max-aspect-ratio**

화면 비율을 의미하며, 이 값은 width를 height로 나누어 구한다.

### **device-aspect-ratio, min-device-aspect-ratio, max-device-aspect-ratio**

기기의 물리적인 화면 비율을 의미한다.

### **color, min-color, max-color**

기기에서 사용하는 색상 당 비트 수를 의미한다.

### **color-index, min-color-index, max-color-index**

기기에서 사용하는 최대 색상 수를 의미한다.

### **monochrome, min-monochrome, max-monochrome**

흑백 기기에서의 픽셀 당 비트 수를 의미한다.

## 8.2 미디어 쿼리의 속성

### **resolution, min-resolution, max-resolution**

단말기에서 지원하는 해상도를 의미한다.

### **scan**

tv 미디어 타입에만 대응하는 속성으로 tv에서 스캔 방식에 따라 프로그레시브 방식은 "progressive", 인터레이스 방식은 "interlace"로 지정할 수 있다.

### **grid(화면의 최소단위)**

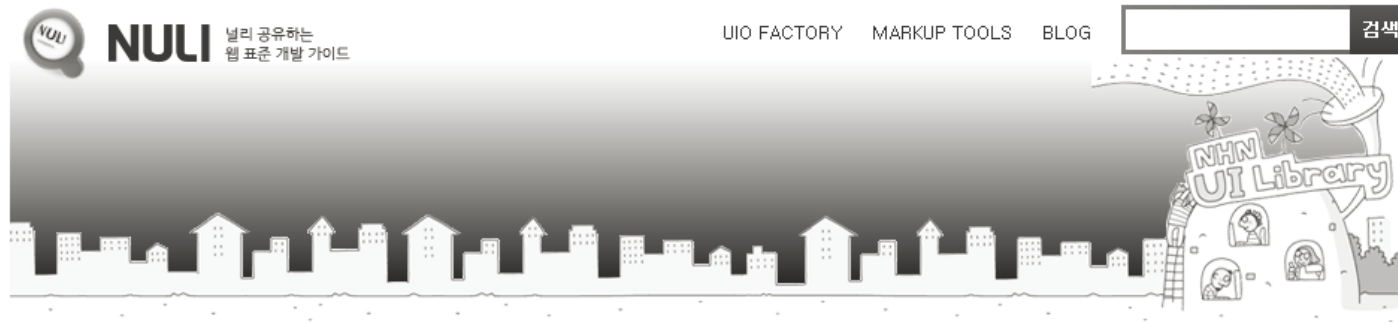
기기가 grid 방식인지 또는 bitmap 방식인지를 의미한다.

### **-webkit-min-device-pixel-ratio**

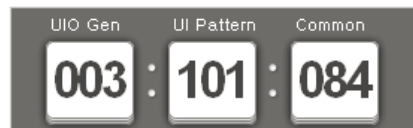
웹킷 전용 속성으로 기기의 화소와 실제 화면의 픽셀(px) 간의 비율을 의미한다.

## 8.3 미디어 쿼리 사이트

- 다음은 미디어 쿼리를 적용하여 브라우저의 크기별로 스타일 시트를 다르게 적용한 사이트이다.
- NULI(<http://html.nhndesign.com/>)는 그 크기에 따라 최적화된 서로 다른 레이아웃을 적용하여 사용자들에게 가장 보기 좋은 화면을 제공하고 있다.



### UIO Factory



UIO Factory는 마크업 개발자의 편의를 돕기 위해 자주 사용하는 UIO를 한 곳에 모아놓은 아카이브입니다.

### TECH@NHN 두번째 시리즈



그동안 널리에서 공유했던 마크업 코딩컨벤션이 TECH@NHN 두번째 시리즈인 "NHN은 이렇게한다! HTML/CSS 코딩컨벤션"으로 발간되었습니다. 해 피비의 경매를 통하여 꼭 10개 이상 기부해주시면 9

### Coding Convention



본 문서는 마크업 개발자가 소스 코드 작성 시에 따라야 할 규칙을 기술하고 있습니다.

## 9. CSS3 참고 사이트

- CSS3에서 새로 추가된 태그들을 직접 적용해볼 수 있는 샘플 코드들을 아래 사이트들을 통해 쉽게 구할 수 있다.
- <http://css3please.com>은 CSS3를 활용한 소스코드와 그 적용 결과를 함께 제공하는 대표적인 사이트이다.
- 크롬 혹은 파이어폭스 브라우저를 통해 사이트를 접속하면 제일 먼저 아래와 같은 결과를 얻을 수 있다.

