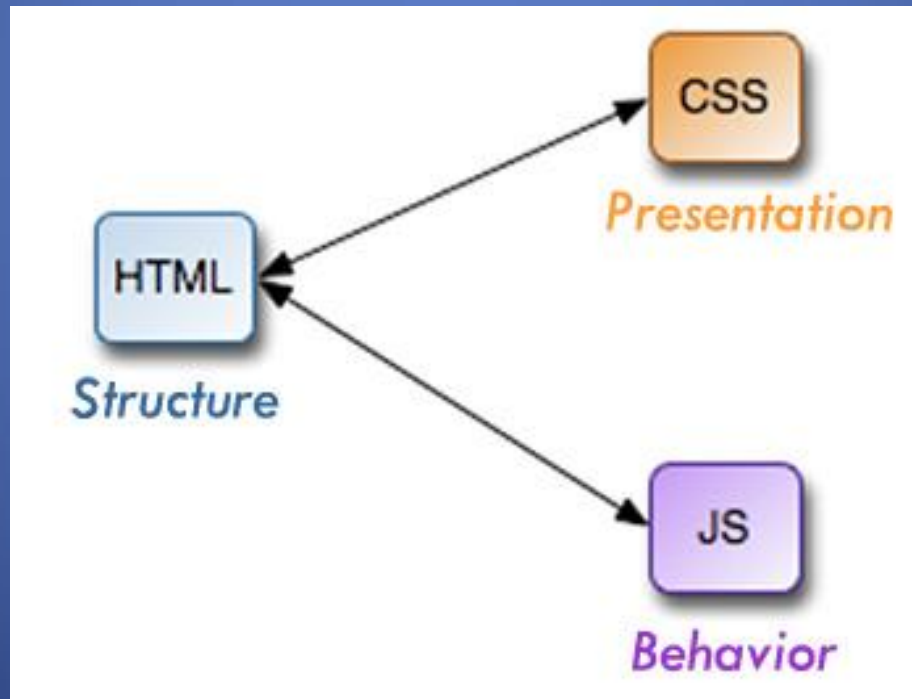


CSS 기초 문법

스타일시트(styleSheet)

: 종속형 스타일 시트(Cascading Style Sheets, CSS)

확장 스타일시트 언어(Extensible Stylesheet Language, XSL)



CSS는 무엇인가?

1. 내부 스타일 시트, 외부 스타일 시트, 인라인 스타일 시트

- 우선순위는 **인라인 스타일 시트**로 각 태그에 직접적으로 적용

문법규칙

2. 내부 스타일 시트

```
<head>  
  <style type="text/css">  
    p{color:#ff0000;}  
  </style>  
</head>
```

문법규칙

3. 외부 스타일 시트

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

```
<head>  
  <style type="text/css">  
    @import url("css/style.css")  
    p{color:#ff0000;}  
  </style>  
</head>
```

문법규칙

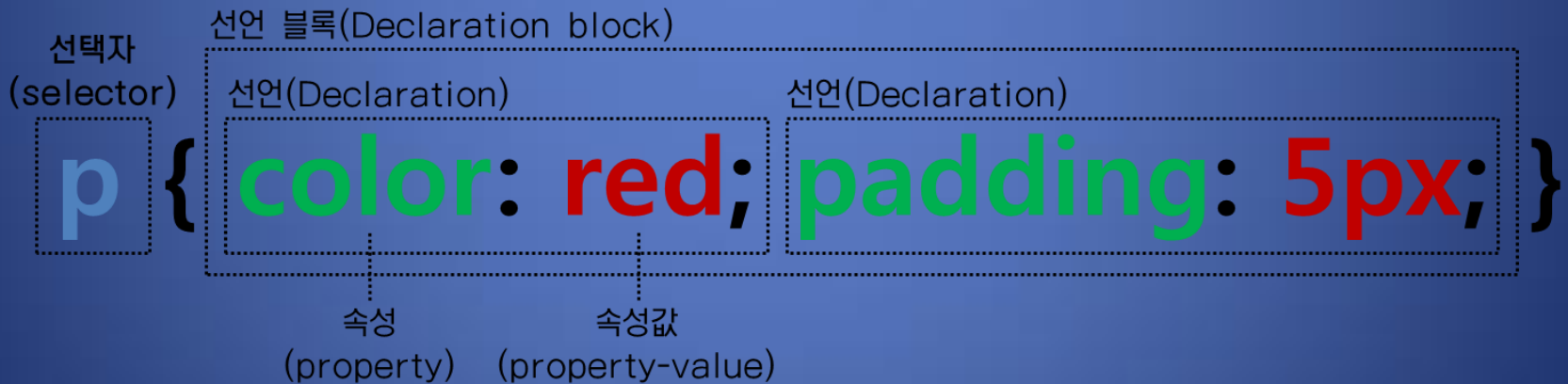
4. 인라인 스타일 시트

<p style="color:#ff0000;"> 인라인 스타일 시트 적용</p>

선택자

1. 개요

- CSS에서 선택자는 다양하고 많은 태그들 사이에서 각각의 스타일을 줄 수 있음.



선택자

2. 전체 선택자

- HTML페이지 내부의 모든 태그를 선택
- 예시 - * { margin: 0; text-decoration: none; }

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    * { color : blue; }
  </style>
</head>
<body>
  <h1>당신에게 지금 가장 필요한 것은?</h1>
  <ul>
    <li>옷</li>
    <li>신발</li>
    <li>안경</li>
    <li>장갑</li>
  </ul>
</body>
<Product by 안재욱>
```


선택자

3. 태그 선택자

- HTML 페이지 내부의 지정 태그를 선택
- 예시 - `p{color:#ff0000;}`

```
<head>
  <title> CSS3 </title>
  <meta charset="utf-8">
  <style type = "text/css">
    p { color : blue; }      /* p 태그 안의 글자들을 파란색으로 처리함.*/
  </style>
</head>
<body>
  <p>기분이 좋아지는 음식</p>
  <h1>딸기 케이크</h1>
  <h2>초콜렛</h2>
</body>
```

선택자

4. 클래스 선택자

- HTML 페이지에서 클래스의 속성값을 가진 요소만 선택
- 클래스의 속성값은 복수(2개 이상)가 가능
- 클래스의 속성값에 첫 글자는 숫자 표기 금지(인식 안됨)
- 예시 - `.register-docu-001{color:#ff0000;}`

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    .ctxt { color : blue; }
  </style>
</head>
<body>
  <p class = "ctxt">여행지에서 만나는 추억</p>
</body>
```

선택자

4. 클래스 선택자 - 실습

- 아래의 코드를 작성하고 세번째 <p> 태그 내부의 문장을 **폰트크기 24px**로 변경하시오.
- 네번째 <p> 태그 내부의 문장의 색상을 **green**으로 변경하시오.

```
<body>
    <p class = "txt_01">나만의 여행을 즐기는 법</p>
    <p class = "txt_02">가족과 함께 가면 좋은 여행지 추천</p>
    <p class = "txt_03">나홀로 여행의 장점</p>
    <p class = "txt_04">맞춤형 여행 패키지</p>
</body>
```

선택자

4. 클래스 선택자 - 실습

- 아래와 같이 동일한 코드를 4개 작성하고 전체 이미지의 가로 사이
즈를 150px로 구성하고,
- 두번째 태그 내부의 이미지의 가로 사이즈를 200px로 조정하시오.
- 세번째 태그 내부의 이미지는 세로 사이즈를 300px로 조정하시오.

```
<body>
```

```
  
```

```
  
```

```
  
```

```
  
```

```
</body>
```

선택자

5. 아이디 선택자

- HTML 페이지에서 아이디(id)의 속성값을 가진 요소만 선택
- 아이디의 속성값에는 단 한 개만의 속성값을 작성. 복수 속성값 불가
- 단 하나만 존재해야 하기 때문에 유연성이 없고 재활용 불가!
- 예시 - `#register{color:#ff0000;}`

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type="text/css">
    # tea-style { color : blue; }
  </style>
</head>
<body>
  <p id="tea-style">건강에 좋은 차</p>
  <p>건강에 좋은 차</p>
</body>
```

선택자

6. 자식 선택자

- 기준 태그보다 하위에 있는 태그를 선택할 때 사용
- 예시 - `ul>li{list-style:none; float:left, color:#ff0000;}`
- `.main-menu>.sub-menu{list-style:none; float:left, color:#ff0000;}`

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    li > p { color : blue; }
  </style>
</head>
<body>
  <ul>
    <li> <p>7월의 여행지 </p> </li>
    <li> <p>8월의 여행지 </p> </li>
  </ul>
  <p>내년의 여행지 </p>
</body>
```

선택자

7. 인접 선택자

- 요소 주변에 있는 부분을 지정하여 스타일을 적용할 때 사용
- 예시 - P + h1{list-style:none; font-weight:bold; color:#ff0000;}

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    h1 + p { color : blue; }
  </style>
</head>
<body>
  <h1>엑스포 안내</h1>
  <p>책자를 배부하고 있습니다.</p>
  <p>안내데스크를 활용하시기 바랍니다.</p>
</body>
```

선택자

8. 병렬 선택자

- 다양한 요소를 선택하여 스타일을 적용할 때 사용
- 예시 – `P, h1{font-weight:bold; color:#ff00ff;}`

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    h1, p { color : blue; }
  </style>
</head>
<body>
  <h1>건강한 삶</h1>
  <h2>하루에 물 2리터 마시기</h2>
  <p>물을 마시는 것은 노폐물 배출에 따른 몸의 정화작용을
불러올 수 있다.</p>
</body>
```


선택자

9. 방문(visited) 및 링크(link) 선택자

- a:link { color: red; } – 클릭 전 상태를 대상으로 선택
- a:visited { color: purple; } – 클릭 후 또는 방문했던 페이지

11. 이전(before) 및 이후(after) 선택자

- a:before { background-color: #000; } – 선택자 앞에서 오는 가상 요소
- a:after { background-color: #aaa; } – 선택자 뒤에서 오는 가상 요소
- CSS3 선택자 명세서상, 가상요소는 두 개의 콜론(::)으로 표현하지만 한 개의 콜론(:)도 허용

12. 마우스오버(hover) 선택자

- a:hover { border-bottom: 1px solid black; }
- 사용자가 요소 위에 커서를 올릴 때 특정한 스타일을 적용
- 익스플로러(IE) 하위 버전(6)에서 반응하지 않음.

선택자

```
<head>
  <link type="text/css" rel="stylesheet" href="style.css"/>
  <title> </title>
</head>
<body>
  <div>
    <a href="http://www.naver.com">Naver</a>
    <a href="http://www.google.com">Google</a>
    <a href="https://www.wikipedia.org">Wikipedia</a>
  </div>
</body>
```

[css]

```
a:link { text-decoration: none; color: #008B45; }
a:hover { color: #00FF00; }
a:visited { color: #EE9A00; }
```

선택자

13. 포커스(focus) 선택자

- `a:focus { color: red; }` – 요소에 초점이 맞춰진 상태
- 요소에 포커스가 머물러 있는 동안 해당 요소를 선택

14. 활성(active) 선택자

- `a:active { background-color: #000; }` – 마우스를 클릭할 때
- 요소에 마우스 클릭 또는 키보드 Enter가 눌린 동안 해당 요소를 선택

선택자

15. 속성(Attribute) 선택자

- `a[href] {color:#ffff00;}` / `a[href="one.html"] {color:red}`
- `제품소개`
- 태그 내부에 있는 속성을 가리킴

selector[attribute]	지정된 어트리뷰트를 갖는 모든 요소를 선택한다.
selector[attribute="값"]	지정된 어트리뷰트를 가지며 지정된 값과 어트리뷰트의 값이 일치하는 모든 요소를 선택한다.
selector[attribute~="값"]	지정된 어트리뷰트의 값이 지정된 값을 (공백으로 분리된) 단어로 포함하는 요소를 선택한다.
selector[attribute ="값"]	지정된 어트리뷰트의 값과 일치하거나 지정 어트리뷰트 값 뒤 연이은 하이픈("값-")으로 시작하는 요소를 선택한다.
selector[attribute^="값"]	지정된 어트리뷰트 값으로 시작하는 요소를 선택한다.
selector[attribute\$="값"]	지정된 어트리뷰트 값으로 끝나는 요소를 선택한다.
selector[attribute*="값"]	지정된 어트리뷰트 값을 포함하는 요소를 선택한다.

선택자

15. 속성(Attribute) 선택자

- `a[href] {color:#ffff00;}` / `a[href="one.html"] {color:red}`
- `제품소개`
- 태그 내부에 있는 속성을 가리킴

selector[attribute]	지정된 어트리뷰트를 갖는 모든 요소를 선택한다.
selector[attribute="값"]	지정된 어트리뷰트를 가지며 지정된 값과 어트리뷰트의 값이 일치하는 모든 요소를 선택한다.
selector[attribute~="값"]	지정된 어트리뷰트의 값이 지정된 값을 (공백으로 분리된) 단어로 포함하는 요소를 선택한다.
selector[attribute ="값"]	지정된 어트리뷰트의 값과 일치하거나 지정 어트리뷰트 값 뒤 연이은 하이픈("값-")으로 시작하는 요소를 선택한다.
selector[attribute^="값"]	지정된 어트리뷰트 값으로 시작하는 요소를 선택한다.
selector[attribute\$="값"]	지정된 어트리뷰트 값으로 끝나는 요소를 선택한다.
selector[attribute*="값"]	지정된 어트리뷰트 값을 포함하는 요소를 선택한다.

선택자

15. 속성(Attribute) 선택자

(돌발퀴즈)

1. 다음 중 빨간(red) 글씨로 표현되는 것은 무엇일까요?

```
<style>div[class*="test"] { color: red; }</style>
```

```
<div class="test">1.The first div element.</div>
```

```
<div class="test2">2.The second div element.</div>
```

```
<div class="test3">3.The third div element.</div>
```

```
<div class="test4">4.The fourth div element.</div>
```

2. 다음 중 빨간(red) 글씨로 표현되는 것은 무엇일까요?

```
<style>div[class$="test"] { color: red; }</style>
```

```
<div class="test-01">1.The first div element.</div>
```

```
<div class="02-test">2.The second div element.</div>
```

```
<div class="test-test">3.The third div element.</div>
```

```
<div class="t-test-fourth">4.The fourth div element.</div>
```

3. 다음 중 빨간(red) 글씨로 표현되는 것은 무엇일까요?

```
<style>div[class^="test"] { color: red; }</style>
```

```
<div class="test-01">1.The first div element.</div>
```

```
<div class="02-test">2.The second div element.</div>
```

```
<div class="test-test">3.The third div element.</div>
```

```
<div class="fourth-test">4.The fourth div element.</div>
```

선택자

15. 속성(Attribute) 선택자 – 실습

- 속성 선택자를 이용하여 아래 항목 중 세 번째 글자 색상을 RED로 변경하시오

```
<body>  
  <ul>  
    <li><a href="index.html">메인화면</a></li>  
    <li><a href="project.html">프로젝트</a></li>  
    <li><a href="about.html">본인 소개</a></li>  
  </ul>  
</body>
```

선택자

15. 속성(Attribute) 선택자 – 실습

- 전체 이미지의 가로 사이즈를 100px로 구성하시오.
- 속성 선택자를 이용하여 아래 항목 중 두 번째 이미지의 가로 사이즈를 250px로 변경하시오.

```
<body>
```

```
  <p>몸에 좋은 그린푸드</p>
```

```
  
```

```
  
```

```
  
```

```
</body>
```


선택자

16. 구조적 가상 선택자

- 동일한 형제 요소가 나열된 경우, 몇 번째를 선택자로 지정할 것인지에 따라 적용

li:nth-child(n)	위에서부터 몇 번째 자식 요소인지를 선택 (n)의 표기 방식 - 일반적인 숫자 부여 가능 - 짝수일 때 표기 : even, 2n - 홀수일 때 표기 : odd, 2n-1
li:nth-last-child(n)	아래에서부터 몇 번째 자식 요소인지를 선택 (n)의 표기방식은 상동
li:first-child	첫 번째 자식 요소만 선택
li:last-child	마지막 자식 요소만 선택
li:nth-of-type(n)	해당 요소 중 앞으로부터 순서가 일치하는 해당 요소를 선택 (해당 요소의 순서만 계산에 포함)
li:nth-last-of-type(n)	해당 요소 중 뒤로부터 순서가 일치하는 해당 요소를 선택 (해당 요소의 순서만 계산에 포함)
li:first-of-type	해당 요소 중 첫 번째를 선택
li:last-of-type	해당 요소 중 마지막을 선택

선택자

16. 구조적 가상 선택자

- 구조적 가상 선택자를 이용하여 아래 항목 중 두 번째 글자("**Project**") 색상을 red로 변경하시오.
- 동일한 방식으로 마지막 태그 내의 <a>태그의 폰트 크기를 24px로 변경하시오.

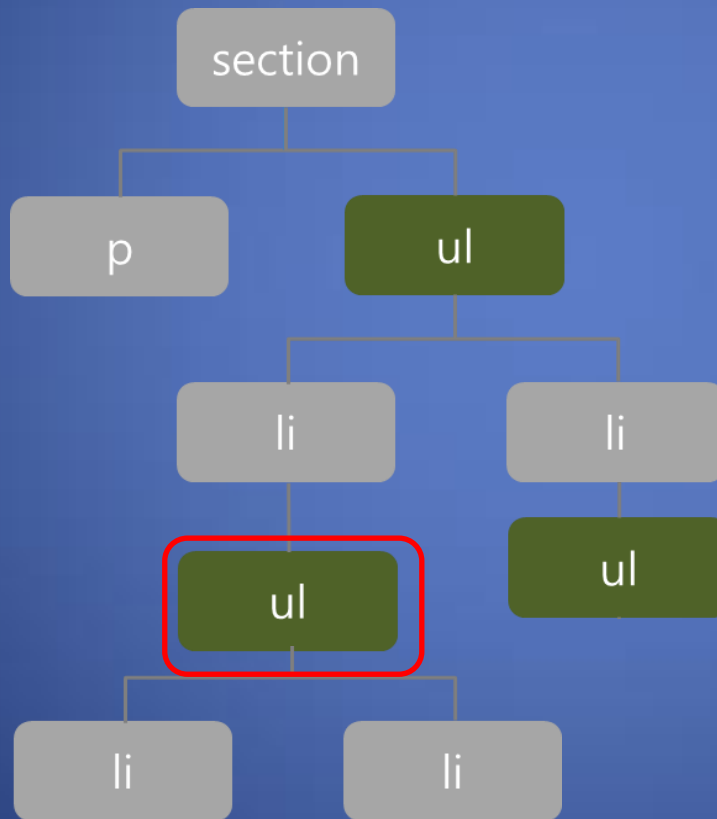
```
<body>
  <ul>
    <li><a href="index.html">Main</a></li>
    <li><a href="project.html">Project</a></li>
    <li><a href="about.html">About Me</a></li>
  </ul>
</body>
```

선택자

16. 구조적 가상 클래스 선택자

(돌발퀴즈)

- 앞의 구조적 가상 클래스 선택자를 이용하여 **복합 선택자 방식** (**a>b>c>d{속성:속성값}**)으로 빨간색 박스 영역을 선택자로 구성하세요.



선택자

17. 부정(not) 선택자

- 예시 : p:not(a){color:#fff;}
- a가 아닌 p 요소를 선택

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    p:not(a){color:#777;}
  </style>
</head>
<body>
  <p>물을 마시는 것은 노폐물 배출에 따른 몸의 <a href="">정화작용</a>을 불러올 수 있다.</p>
</body>
```

선택자

18. 종속 선택자

- 예시 : p.main {color:#fff;}
- <p>태그 중에서 class의 속성값이 main인 요소만을 선택
- <p class="main"> 텍스트</p>

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style type = "text/css">
    p.txt{color:#777;}
  </style>
</head>
<body>
  <p class="txt">물을 마시는 것은 노폐물 배출에 따른 몸의 <a
href="">정화작용</a>을 불러올 수 있다.</p>
</body>
```

선택자

18. 선택자 우선순위

- 제작자(author) 원천 소스 : 웹 사이트 제작자가 지정하는 자신의 페이지 스타일
- 사용자(user) 원천 소스 : 사용자가 직접 정하는 자신이 사용할 스타일
- 사용자 도구(user agent) 원천 소스 : 웹 브라우저 자체에 지정된 기본 스타일

1) 원천 소스 우선 순위

- **! important 선언을 한 사용자 스타일 > ! important 선언을 한 제작자 스타일 > 제작자 스타일 > 사용자 스타일 > 사용자 도구 선언 (브라우저 자체의 선언)**

2) CSS 명시도(Specificity) 계산법

- **! important > id [100] > class [10] > tag [1] > * [0]**

- ① ID 선택자의 갯수를 세어서 개당 100 으로 계산 (= a)
- ② 클래스 선택자의 갯수를 세어서 개당 10 으로 계산 (= b)
- ③ 태그 선택자의 갯수를 세어서 개당 1 로 계산 (= c)
- ④ 공용 선택자는 모두 0으로 계산 (= d)
- ⑤ 가상 엘리먼트는 무시

선택자

18. 선택자 우선순위

3) 마지막에 지정된 스타일을 우선 적용

- 마지막에 지정된 스타일이 적용

폰트

1. 폰트의 속성

- 글꼴, 글자 크기, 글자 굵기, 기울임 여부, 줄간격, 대소문자 여부, 글자 색상 등을 표현

속성	풀이
font-family	글꼴을 인용부호로 감싸 지정
font-size	단위를 붙여 표현(px, em, %)
font-weight	두께(bold)나 수치 적용
font-style	Italic, normal
font-variant	Small-caps로 작은 대문자 표현
line-height	줄간격

폰트

2. 폰트의 크기(size)

- 단위로는 기존에 px, em, %가 존재했으며, 반응형이 접목되면서 rem, vw 단위가 추가됨
- em의 경우, 기본값이 1em=16px이며, 상위 요소에서 font-size를 설정했을 경우, 변동되는 단위

```
<head>
  <title> CSS </title>
  <meta charset="utf-8">
  <style>
    p {font-size: 16px; line-height: 2em;}
    div {font-size: 30px; border: 1px solid #000;}
    span {font-size: 0.5em;}
  </style>
</head>
<body>
  <p>요거트로 '비행기 연료' 만든다 (연구)</p>
  <div>요거트를 만들 때 발생하는 찌꺼기 액체를 비행기 연료로 재활용 할 수
  있다는 연구결과가 나왔다. <span>미국 코넬 대학과 독일 튀빙겐 대학 합동 연구진은
  그릭 요거트를 만드는 과정에서 먹을 수 없는 찌꺼기가 발생하는데, 이 찌꺼기와 박테
  리아를 특정 비율로 혼합하면 비행기 연료로 사용할 수 있다고 밝혔다.</span></div>
</body>
```

폰트

3. 폰트의 볼륨(weight)

- 요소마다 다른 font-weight를 갖고 있으나 기본값은 font-weight:normal;
- font-weight:bold; 설정시 글씨 굵게 표현.
- font-weight의 경우 수치상으로 굵기를 변동할 수 있음
- 대략 300~400은 normal, 500~900은 bold 형태를 띠

속성값	풀이
normal	보통 폰트 가중치. 400 과 같음.
bold	굵은 폰트 가중치. 700 과 같음.
lighter	(가능한 폰트 가중치 중) 부모 요소(element) 보다 얇은 폰트 가중치.
bolder	(가능한 폰트 가중치 중) 부모 요소 보다 굵은 폰트 가중치.
100, 200, 300, 400, 500, 600, 700, 800, 900	normal 과 bold 이외를 제공하는 폰트를 위한 숫자형 폰트 가중치.

폰트

3. 폰트의 볼륨(weight)

```
font-weight: normal;  
font-weight: bold;
```

```
font-weight: lighter;  
font-weight: bolder;
```

```
font-weight: 100;  
font-weight: 200;  
font-weight: 300;  
font-weight: 400;  
font-weight: 500;  
font-weight: 600;  
font-weight: 700;  
font-weight: 800;  
font-weight: 900;
```

```
font-weight: inherit;  
font-weight: initial;  
font-weight: unset;
```

폰트

4. 폰트의 타입(font-family)

- 각 브라우저마다 기본적으로 지원하는 폰트가 다름. 레이아웃에 영향을 줌
- 동일한 문자타입이 나오도록 하려면,
- ① 서버로부터 폰트를 다운받는 형식
- ② 외부경로를 통해 import 하는 방식(from google fonts)

```
font-family : '나눔고딕', '맑은고딕', 'Apple Gothic', 'Arial', '굴림', serif;
```

폰트

4. 폰트의 타입(font-family)

1) 서버로부터 폰트를 다운받는 형식

- 폰트 파일을 지정한 폴더 내부에 넣음

(만약, 웹 폰트 파일이 없다면, <https://www.web-font-generator.com/> 에서 폰트 변환)

- css 파일 상단에 @font-face 적용

```
@font-face {  
    font-family: "Samplino"; src:  
    url("fonts/Samplino.eot?") format("eot"),  
    url("fonts/Samplino.woff2") format("woff2"),  
    url("fonts/Samplino.woff") format("woff"),  
    url("fonts/Samplino.ttf") format("truetype"),  
    url("fonts/Samplino.svg#Samplino") format("svg");  
}  
Body{font-family: "Samplino"; }
```

폰트

4. 폰트의 타입(font-family)

2) 외부경로를 통해 import 하는 방식

- google fonts에서 적용할 폰트 선택 후, 적용 형식 copy
- css 파일 상단에 @import 적용

```
@import url('https://fonts.googleapis.com/css?family=Ubuntu');
```

또는

- google fonts에서 적용할 폰트 선택 후, 적용 형식 copy
- <head> 내부에 <link> 태그로 적용

```
<link href="https://fonts.googleapis.com/css?family=Ubuntu"  
rel="stylesheet">
```

폰트

4. 폰트의 타입(font-family)

3) font-family의 적용

- @font-face 또는 @import를 통해서 폰트의 이름을 지정.
- **font-family: '폰트 이름';** 으로 작성

```
body{font-family:'Ubuntu';}
```

- 만약 지원 단말 혹은 OS별로 적용 불가능한 브라우저 지원 폰트일 경우, 순차적으로 폰트가 적용되도록 설정
- 기본 폰트를 제외한 모든 폰트이름은 ' ' (작은따옴표) 내부에 넣고 ,(쉼표)로 구분 (마지막 폰트는 쉼표 넣지 말 것!!!)

```
font-family : 'Nanum Gothic', 'Malgun Gothic', 'Apple Gothic', 'Arial', '굴림', serif;
```

문단(Paragraph)

가로정렬, 세로정렬, 들여쓰기, 대/소문자, 줄치기, 자간조절 등을 표현

1. text-align(가로방향 정렬)

- left(왼쪽 정렬), center(중앙 정렬), right(오른쪽 정렬), justify(양쪽 정렬) 로 정렬 가능
- 블록 요소에만 적용

2. vertical-align(세로방향 정렬)

- top, middle, bottom 로 정렬 가능
- img, input, select, 테이블의 th, td 등에 사용
- 인라인 블록인 요소에서 사용 가능(인라인 블록은 높이가 다를 경우, 하단 기준)

문단(Paragraph)

1. text-align(가로방향 정렬)

- left(왼쪽 정렬), center(중앙 정렬), right(오른쪽 정렬), justify(양쪽 정렬) 로 정렬 가능
- 블록 요소에만 적용

```
<head>
  <style>
    .left-align{text-align : left;}
    .center-align{text-align : center;}
    .right-align{text-align : right;}
  </style>
</head>
<body>
  <p class="left-align">왼쪽 정렬</p>
  <p class="center-align">가운데 정렬</p>
  <p class="right-align">오른쪽 정렬</p>
</body>
```

문단(Paragraph)

2. vertical-align(세로방향 정렬) - 실습

- vertical-align을 이용하여 수직 방향으로 정렬하시오.
- 트위터 아이콘 사이즈는 가로만 72px을 부여. 아이콘 우측 글씨는 18px로 조정.
- 아이콘을 기준으로 수직 방향으로 텍스트가 가운데 올 수 있도록 구성

```
<style>
.icon_menu a{text-decoration:none; color:#777; display:inline-block; background-
color:#ffaaff;}
.icon_menu img{width:72px; display:inline-block; vertical-align:middle;}
.icon_menu span{font-size:18px;}
</style>

<body>
    <div class="icon_menu">
        <a href="#"><span>트위터</span></a>
    </div>
</body>
```

문단(Paragraph)

2. vertical-align(세로방향 정렬) – 실습(직접 실습)

- vertical-align을 이용하여 수직 방향으로 정렬하시오.
- Arrow Icon 사이즈는 가로만 24px을 부여. 아이콘 좌측 글씨는 48px로 조정.
- 텍스트 기준으로 수직 방향으로 아이콘이 가운데 올 수 있도록 구성

```
<body>
  <div class="main_menu">
    <a href="#"><span>판매상품</span></a>
  </div>
</body>
```

문단(Paragraph)

가로정렬, 세로정렬, 들여쓰기, 대/소문자, 줄치기, 자간조절 등을 표현

3. text-indent (문단 들여쓰기)

- 입력한 수치만큼 좌우로 이동
- px, %, em 의 수치로 조정

4. text-transform (대소문자 변경)

- uppercase(대문자), lowercase(소문자), capitalize(첫 글자만 대문자)
- 영문 가능, 한글 적용 불가

문단(Paragraph)

가로정렬, 세로정렬, 들여쓰기, 대/소문자, 줄치기, 자간조절 등을 표현

5. text-decoration (줄을 치는 위치)

- underline(밑줄), overline(윗줄), line-through(가운데 줄), none(밑줄 제거)
- 대표적인 사용 예로 링크 <a>태그에서 밑줄을 제거

6. letter-spacing (자간 조정)

- 공간상 한 줄이 들어가야 할 때 사용
- px, %, em 의 수치로 조정

7. line-height (줄 간격)

- 공간상 한 줄이 들어가야 할 때 사용
- px, %, em 의 수치로 조정

문단(Paragraph)

```
<head>
  <title> CSS3 </title>
  <meta charset="utf-8">

  <style type="text/css">
    body { font : 12px "굴림", Gulim ; }
    h1 { letter-spacing : 5px ; }
    p { width:500px;
        text-align : justify ;
        text-indent : 5px ;
        text-transform : lowercase ;
      }
  </style>
</head>
<body>
  <h1>영상과 인쇄의 공존</h1>
  <p>영상의 발달로 인해 인쇄는 곧 사라질 것처럼 생각했지만 예상을 뒤집고 인쇄물들은 점점 고급화를 향해 달리고 있다. e-book 의 출현과 함께 서점에서 책을 사모으는 취미는 여전하고 FACEBOOK 의 도전에도 불구하고 많은 사람들은 책장을 손으로 넘기고 있다. 좋은 인쇄물을 갖는 것은 큰 즐거움이며 스마트 폰을 두드리는 손가락에 의해 흥미로운 책장도 계속 넘어간다.</p>
</body>
```

배경(background)

배경색, 배경 이미지, 배경의 반복여부, 배경 위치, 배경 고정 여부를 표현

속성	풀이
background-color	배경색을 지정. 생략시 투명색 적용.
background-image	배경으로 이미지를 지정. <code>url('images/bg-001.jpg')</code>
background-repeat	배경 이미지를 어떻게 반복할지 결정. repeat, no-repeat, repeat-x, repeat-y
background-position	배경의 위치를 정함. (가로위치, 세로위치). % 또는 문자적용
background-attachment	배경 이미지를 고정시킬지, 스크롤 시킬지를 결정 scroll, fixed
background-clip	테두리 내의 배경에 대한 확장 속성을 정의 content-box (콘텐츠 영역까지), padding-box (패딩 영역까지), border-box (보더 영역까지)
background-origin	배경 이미지 시작 기준을 설정 content-box (콘텐츠 영역부터), padding-box (패딩 영역부터), border-box (보더 영역부터)
background	상단의 요소를 통합

배경(background)

```
<head>
  <title> CSS3 </title>
  <meta charset="utf-8">

  <style type="text/css">
    .bg-img{
      width:100%;
      height:250px;
      background-image:url('../img/img-01.jpg');
      background-repeat:no-repeat;
      background-position:50% 50%;
      background-size:cover;
      background-color:#edefed;
    }
  </style>
</head>
<body>
  <div class="bg-img"> </div>
</body>
```


상자

1. width

- 가로 사이즈
- 여백을 포함하지 않는 가로 길이
- 반응형 웹에서 디바이스의 기준이 되는 척도(미디어 쿼리 사용시)

2. height

- 세로 사이즈
- 여백을 포함하지 않는 세로 길이

상자

3. padding

- 전체 박스 안쪽의 여백
- 내용(contents)와 테두리(border) 사이의 간격
- 박스 외곽 전체에 동일한 패딩값 부여 가능 (예시) **padding:20px;**
- 위치별 별도의 패딩값 부여 가능 (예시) **padding-left:10px;**
- **padding:10px 5px 15px 20px → padding: top right bottom left(시계방향)**
- **padding 값이 상하 동일, 좌우 동일할 경우**
→ **padding: top(bottom) right(left)**
(예시) **padding:10px 20px**
- **padding 값이 상하 상이, 좌우 동일할 경우**
→ **padding: top right(left) bottom**
(예시) **padding:10px 20px 15px**

상자

4. border

- 박스의 테두리 정의
- 색상과 모양, 굵기를 지정 가능
- **border-width**(테두리의 두께)
- **border-style**(테두리의 스타일-선 형태)
- **border-color**(테두리의 색상)
- 전체 테두리를 통합적으로 적용 가능 (예시) **border:2px solid #ffff00;**
- 위치별 선을 부여가 가능 (예시) **border-top:1px solid #999;**

값	설명
Solid	실선 형태의 선
Dashed	긴 점선 형태의 선
Dotted	점선 형태의 선
Double	이중선
None	없음
Groove, ridge, inset, outset	액자형태의 선 혹은 테두리

상자

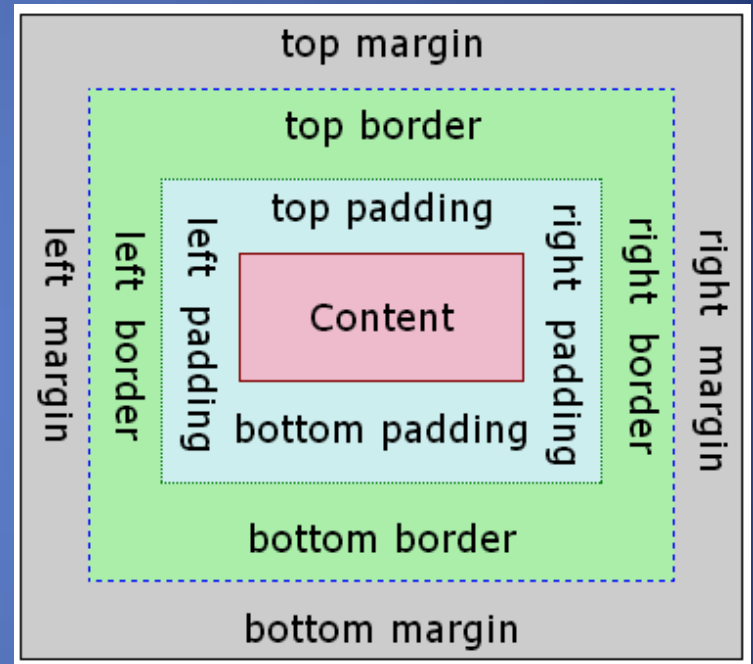
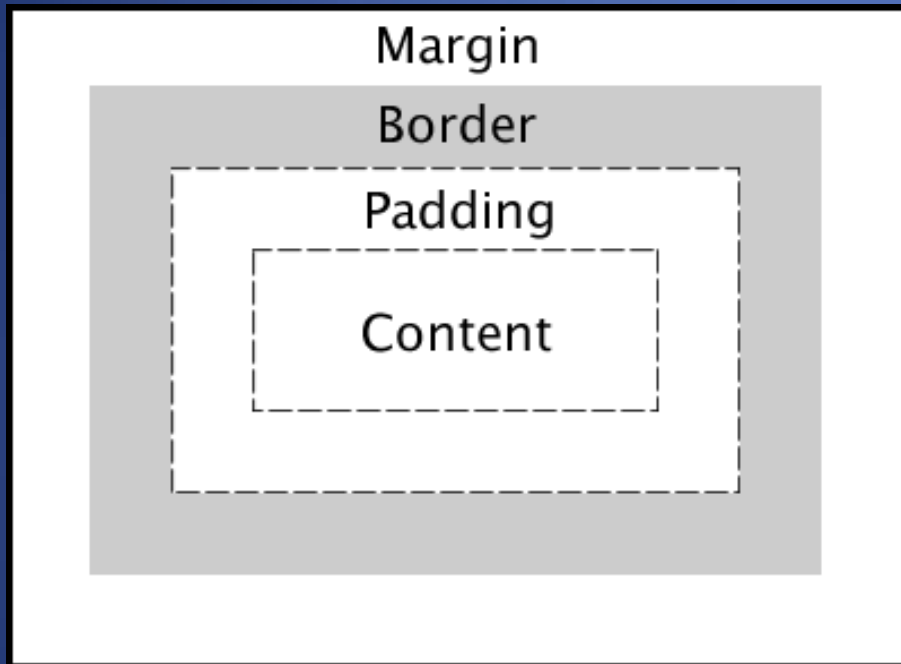
5. margin

- 박스의 바깥쪽 여백
- 테두리와 다음 박스(또는 테두리)와의 사이 간격
- 박스 외곽 전체에 동일한 마진값 부여 가능 (예시) **margin:20px;**
- 위치별 마진값 부여 가능 (예시) **margin-left:10px;**
- **margin:10px 5px 15px 20px → margin: top right bottom left(시계방향)**

- **margin 값이 상하 동일, 좌우 동일할 경우**
→ **margin: top(bottom) right(left)**
(예시) **margin:10px 20px**

- **margin 값이 상하 상이, 좌우 동일할 경우**
→ **margin: top right(left) bottom**
(예시) **margin:10px 20px 15px**

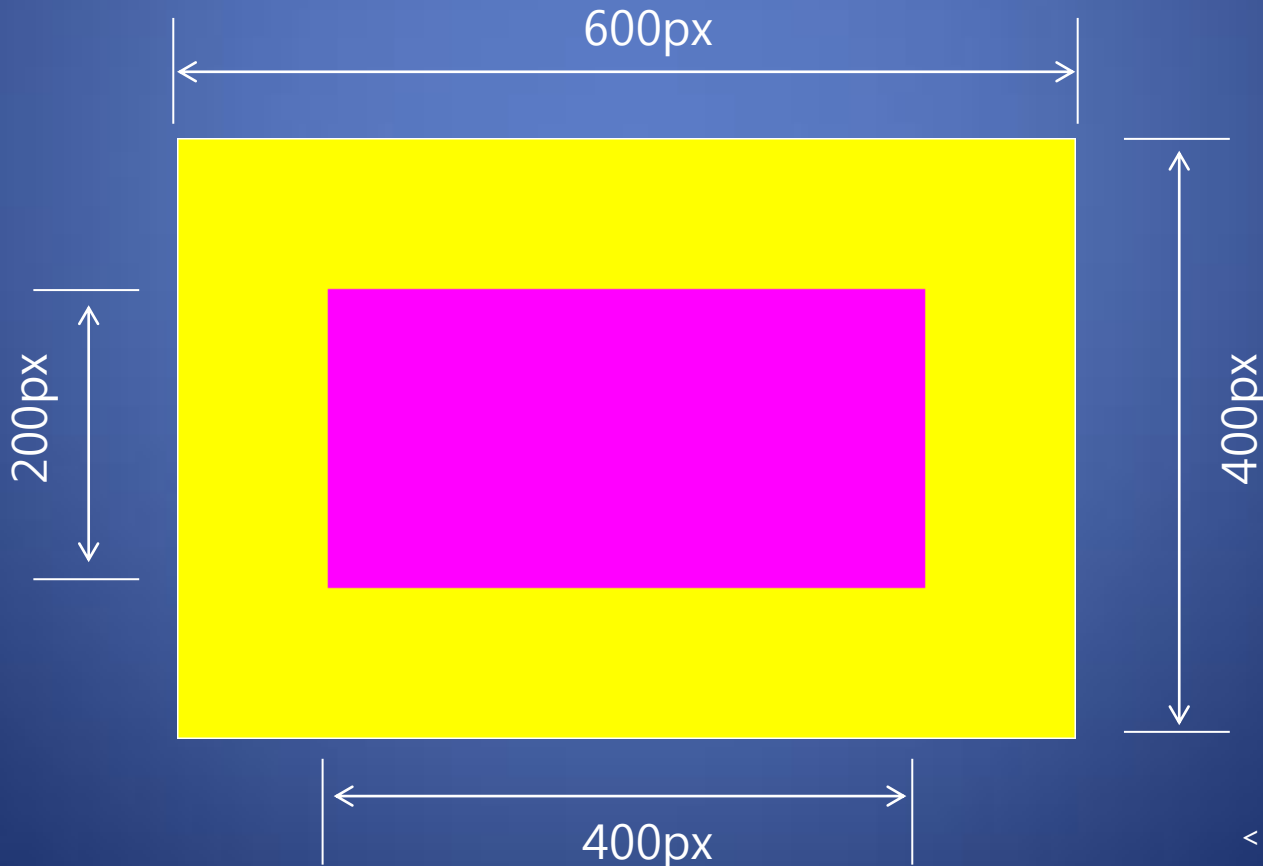
상자



상자

<<실습과제-01>>

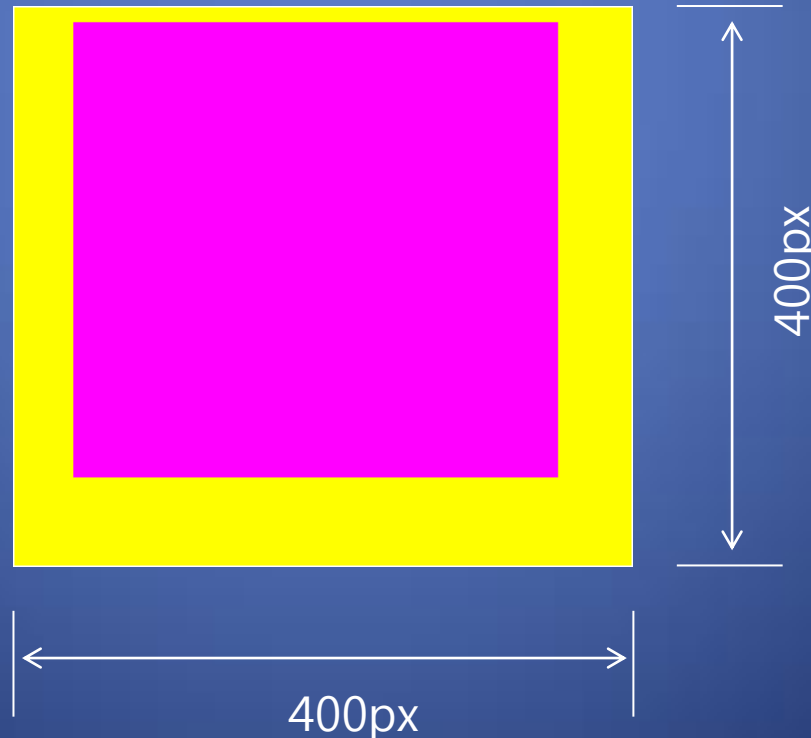
- 가로 600px, 세로 400px 크기의 박스 내부에 가로 400px, 세로 200px 크기의 박스를 정가운데 위치시키세요. (div 태그 이용)
- 조건-01 : padding을 이용
- 조건-02 : 색상은 자유(단, 두박스의 색상이 달라야 함)(색상 예시: 외곽 박스-#ffff00; 내부 박스-#ff00ff;)



상자

<<실습과제-02>>

- 가로 400px, 세로 400px 크기의 박스 내부에 박스를 넣으세요. (div 태그 이용)
- 조건-01 : padding을 이용
- 조건-02 : 외부 박스와 내부 박스의 간격 - 상단에서 10px, 오른쪽에서 50px, 왼쪽에서 40px, 하단에서 60px padding을 이용
- 조건-03 : 색상은 자유(단, 두 박스의 색상이 달라야 함)(색상 예시: 외곽 박스-#ffff00; 내부 박스-#ff00ff;)



레이아웃(Layout)

1. float

- 박스를 좌측이나 우측에 붙이고 그 아래의 내용들이 그 주변으로 흐르게 하는 역할

속성값	설명
left	박스를 왼쪽에 배치, 나머지 콘텐츠를 주변에 흐르도록 지정
right	박스를 오른쪽에 배치, 나머지 콘텐츠를 주변에 흐르도록 지정
unset	float 을 적용하지 않음(IE에서는 적용되지 않기 때문에 none으로 변경)

2. clear

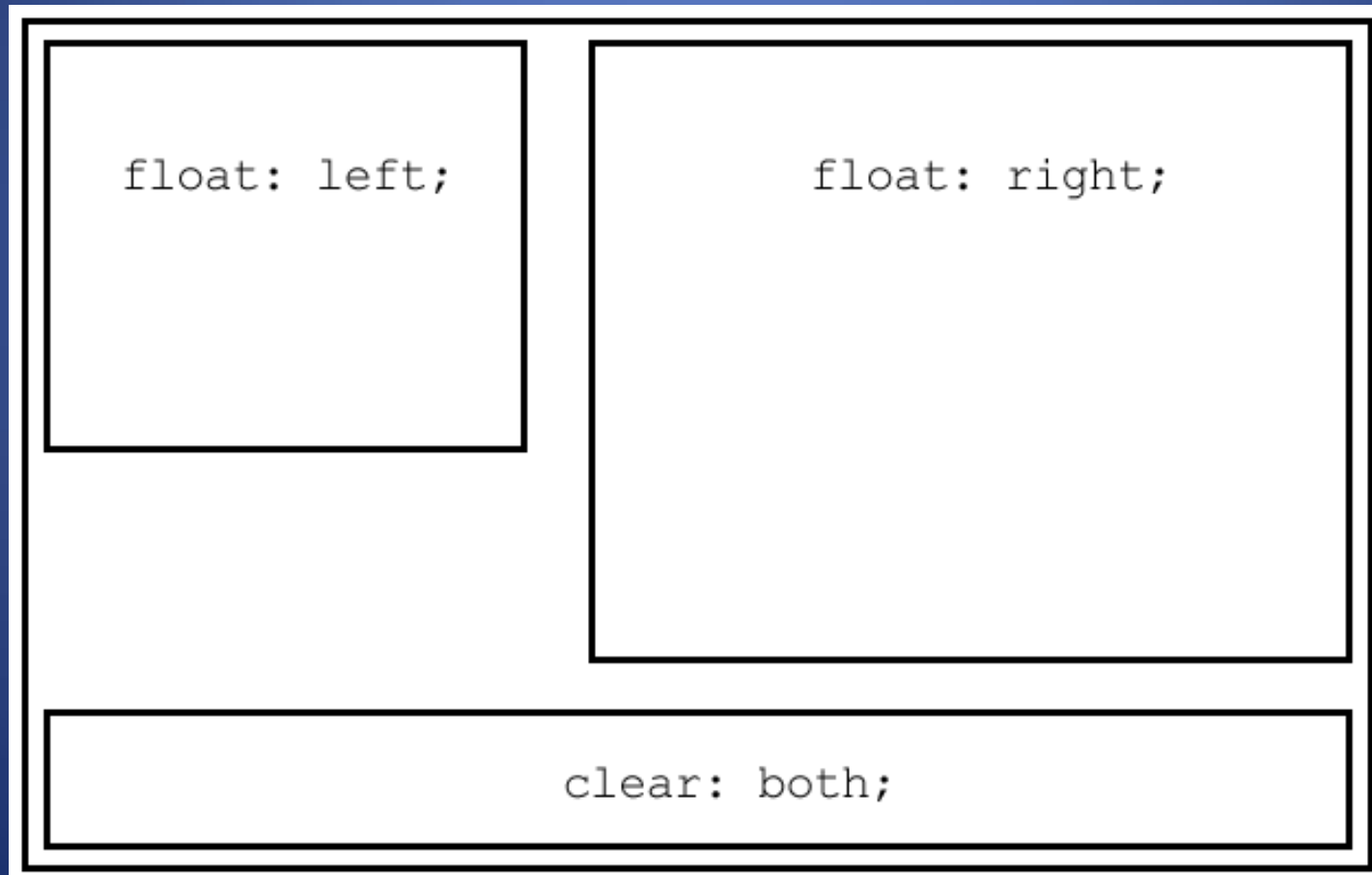
- float으로 강제 설정된 값을 해제하는 역할

속성값	설명
left	float 된 박스 중 좌측의 빈 공간부터 채워 내려옴
right	float 된 박스 중 우측의 빈 공간부터 채워 내려옴
both	float 된 박스 중 어느 쪽도 채우지 않고 내려옴

레이아웃(Layout)

2. clear

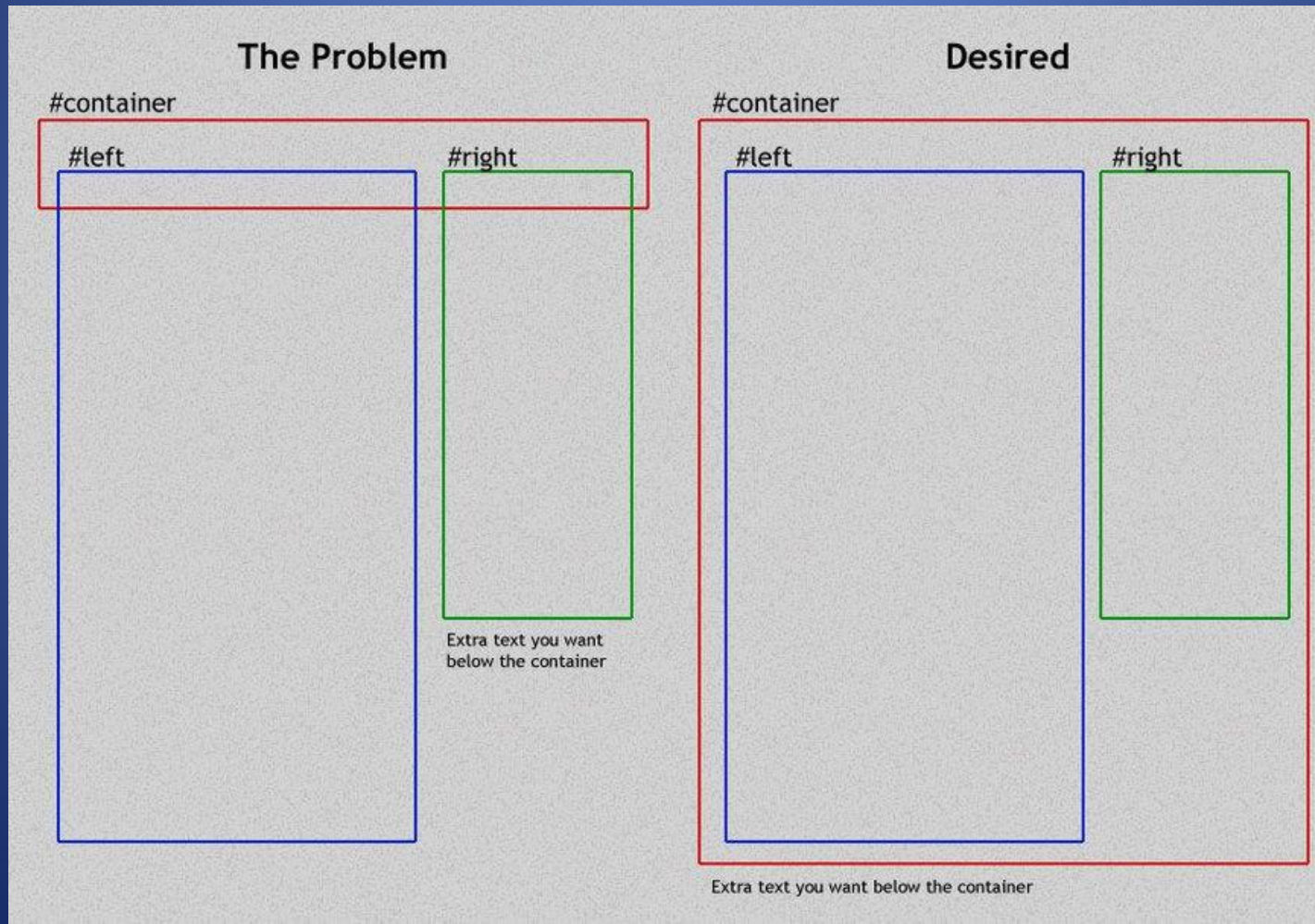
- 일반적인 경우 `<div style="clear:both;"></div>`가 블록 요소 사이에 들어오면 반응형으로 축소 시에도 박스 형태가 재조합 되어 깨지는 현상이 나타나지 않음.



레이아웃(Layout)

2. clear

- 일반적으로 float을 사용하는 경우, #container 공간에 대한 높이 값이 명시하지 않아서 하단의 콘텐츠(텍스트 또는 이미지)가 상단으로 침범하는 사례가 많음.



레이아웃(Layout)

2. clear

- 이런 경우, 반응형에서 의도하지 않은 공간(높이값)을 부여하거나 억지로 맞추려고 무리한 css를 적용시키기 됨.
- `#container:after`라는 가상의 선택자를 설정하여 `{clear:both}`를 적용하면 해결

```
<head>
  <title> CSS3 </title>
  <meta charset="utf-8">
  <style type="text/css">
    #container{width:100%; background:#ffaaff;}
    #container:after{content: ""; visibility: hidden; display: block; height: 0; clear: both;}
    #container:after{content: ""; display: table; clear: both;}/*ie8이하*/
    div#left {float:left; width:30%; height:200px; background:#ffffaa;}
    div#right {float:right; width:30%; height:150px; background:#aaa;}
  </style>
</head>
<body>
  <div id="container">
    <div id="left"> </div>
    <div id="right"> </div>
  </div>
  <div id="left"> </div>
  <div id="right"> </div>
</body>
```

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 속성

값	설명
block	박스를 블록 요소로 만듦 (줄 바꿈 현상 발생)
inline	박스를 인라인 요소로 만듦 (줄바꿈 현상 발생 없음)
inline-block	박스에 인라인 요소를 갖게 하여 줄 바꿈은 일어나지 않게 하면서 블록에 쓸 수 있는 속성으로 사용
None	박스를 보이지 않게 함

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 속성

```
.div-1, .div-2, .div-3{display:block;
width:100px; height:100px;}
.div-1{background:#ffaa00;}
.div-2{background:#00aaff;}
.div-3{background:#aaff00;}
```

```
.div-4, .div-5, .div-6{display:inline;
width:100px; height:100px;}
.div-4{background:#ffaa00;}
.div-5{background:#00aaff;}
.div-6{background:#aaff00;}
```

```
.div-7, .div-8, .div-9{display:inline-block;
width:100px; height:100px;}
.div-7{background:#ffaa00;}
.div-8{background:#00aaff;}
.div-9{background:#aaff00;}
```

```
<body>
<p> - display:block -</p>
    <div class="div-1">일러스트</div>
    <div class="div-2">포토샵</div>
    <div class="div-3">드림위버</div>
<br/>
<p> - display:inline -</p>
    <div class="div-4">Illustrator</div>
    <div class="div-5">Photoshop</div>
    <div class="div-6">Dreaweaver</div>
<br/>
<p> - display:inline-block -</p>
    <div class="div-7">flash</div>
    <div class="div-8">max</div>
    <div class="div-9">alias</div>
</body>
```

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 추가 속성들(table 편)
- <table>이라는 태그의 성격을 반영

속성값	설명
table	테이블 영역을 지정
table-caption	테이블 캡션으로 표시
table-cell	테이블의 셀 영역을 표시
table-column	테이블의 열 영역을 표시
table-column-group	테이블의 열 그룹 영역 표시
table-row	테이블의 행 영역을 표시
table-row-group	테이블의 행 그룹 영역 표시
table-header-group	테이블의 머리글 행 그룹 표시
table-footer-group	테이블의 바닥글 행 그룹 표시

레이아웃(Layout)

2. display

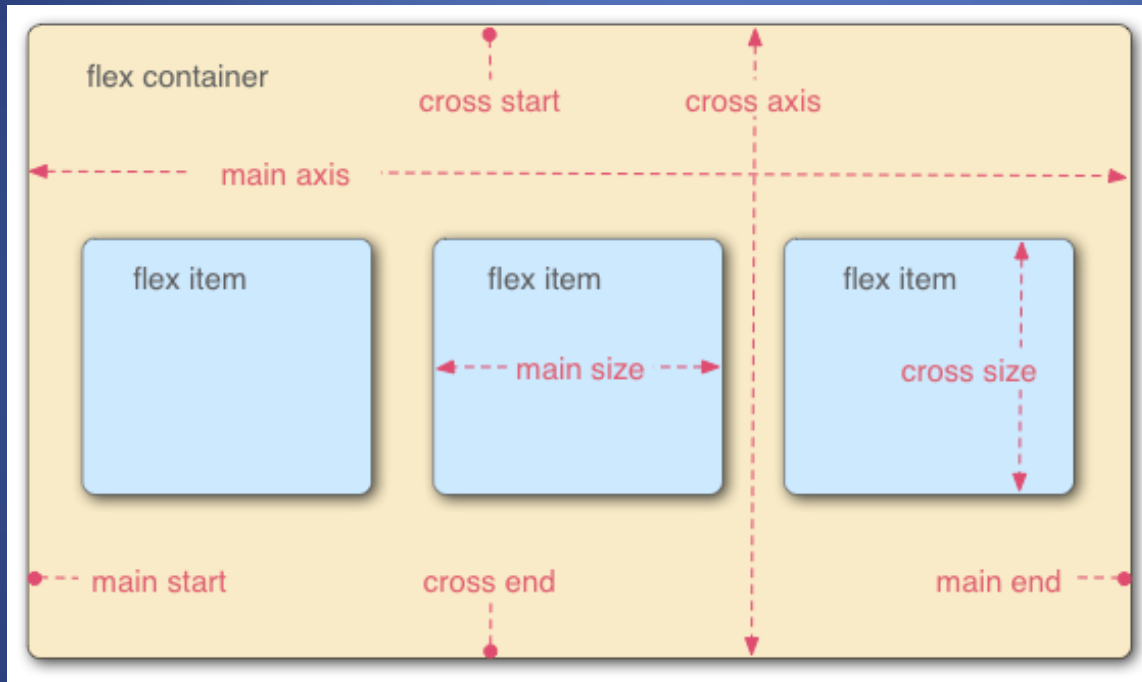
- 요소의 성격을 변경하는 추가 속성들(table 편)

```
<div style="display:table">
  <div style="display:table-row">
    <div style="display:table-cell">1-1</div>
    <div style="display:table-cell">2-1</div>
    <div style="display:table-cell">3-1</div>
  </div>
  <div style="display:table-row">
    <div style="display:table-cell">1-2</div>
    <div style="display:table-cell">2-2</div>
    <div style="display:table-cell">3-2</div>
  </div>
</div>
```

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 추가 속성들(flex 편) – IE9 이하 버전에서 미지원
- 유연성을 뜻하는 레이아웃 형태



- [과거 문법] `display: box(old)`와 같이 사용
- 비공식적인 문법으로 `display: flexbox(hybrid)`와 같이 사용
- 최근 `display: flex(modern)`로 변경
- IE는 `display: flexbox` 로 사용 (IE10 에서는 `-ms-display: flexbox`와 같이 사용)

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 추가 속성들(flex 편)

```
.page-wrap {  
    display: -webkit-box; /* OLD - iOS 6-, Safari 3.1-6 */  
    display: -moz-box; /* OLD - Firefox 19- (buggy but mostly works) */  
    display: -ms-flexbox; /* TWEENER - IE 10 */  
    display: -webkit-flex; /* NEW - Chrome */  
    display: flex; /* NEW, Spec - Opera 12.1, Firefox 20+ */  
}
```

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 추가 속성들(flex 편) – IE9 이하 버전에서 미지원
- 유연성을 뜻하는 레이아웃 형태

속성(값)	설명
flex	내부 요소와 유연성 있게 레이아웃 구성 가능
flex-direction: column;	수평 축을 수직 축으로 전환 column-reverse일 경우 역순으로 표현
flex-direction: row;	수직 축을 수평 축으로 전환 row-reverse일 경우 역순으로 표현
justify-content: center;	가로방향으로 중심축을 중앙정렬 flex-start(default), flex-end, center, space-between, space-around
align-items: center;	세로방향으로 중심축을 중앙정렬 flex-start(default), flex-end, center, baseline, Stretch

레이아웃(Layout)

2. display

- 요소의 성격을 변경하는 추가 속성들(flex 편)

[css]

```
.contain{display:flex; min-height:50px; background:#0000ff; justify-content:center; align-items: center; flex-direction: row;}  
.s-box{width:50px; height:50px; background:#ffaa00; text-align:center; line-height:50px;}
```

```
<div class="contain">  
  <div class="s-box"> 일 </div>  
  <div class="s-box"> 월 </div>  
  <div class="s-box"> 화 </div>  
  <div class="s-box"> 수 </div>  
  <div class="s-box"> 목 </div>  
  <div class="s-box"> 금 </div>  
  <div class="s-box"> 토 </div>  
</div>
```

레이아웃(Layout)

3. position

- 블록 박스의 위치 기준으로 배치하는 기술

값	설명
statics	요소를 이동하거나 겹칠 수 없는 상태. 기본값
relative	left와 top으로 이동할 수 있으며, absolute로 지정된 요소의 부모 역할도 가능
absolute	다른 요소와 겹칠 수 있으며, left와 top을 지정해서 위치를 잡을 수 있음. (예시, 팝업창 위치 조정)
fixed	화면을 기준으로 위치 지정

레이아웃(Layout)

3. position

```
<style>
*{ }
body{ margin:0px; }
#box { position:relative; width:550px; }
p { width:130px; height:80px; padding:10px; color:#fff; font-weight:bold; }
.myred { background:red; }
.myblue { background:blue; position:absolute; left:50px; top:30px; z-index:2; }
.mygreen { background:green; position:relative; left:30px; top:20px; }
</style>
<body>
<div id="box">
    <p class="myred"> </p>
    <p class="myblue"> </p>
    <p class="mygreen"> </p>
</div>
</body>
```

레이아웃(Layout)

4. 2단 레이아웃을 잡는 방법

Header

Container

Footer

레이아웃(Layout)

4. 레이아웃을 잡는 방법

- float 방식
- 고정형일 경우, side-menu{float:left;}, contents{float:right;}
- 반응형일 경우, 모두 {float:left;}

Header

Container

Side-menu

Contents

Footer

레이아웃(Layout)

4. 2단 레이아웃을 잡는 방법

- position 방식
- container{position:relative;}, side-menu{position:absolute; left:0;}, contents{position:absolute; right:0;}

Header

Container

Side-menu

Contents

Footer

레이아웃(Layout)

4. 3단 레이아웃을 잡는 방법

- float 방식
- 고정형일 경우, `side-menu{float:left;}`, `contents{float:left;}`, `banner{float:right;}`
- 반응형일 경우, 모두 `{float:left;}`

Header

Container

Side-menu

Contents

Banner

Footer

레이아웃(Layout)

4. 3단 레이아웃을 잡는 방법

- position 방식
- container{position:relative;}, side-menu{position:absolute; left:0;}, contents{position:absolute; left:side-menu의 width + margin-left 값;}, banner{position:absolute; right:0;}

Header

Container

Side-menu

Contents

Banner

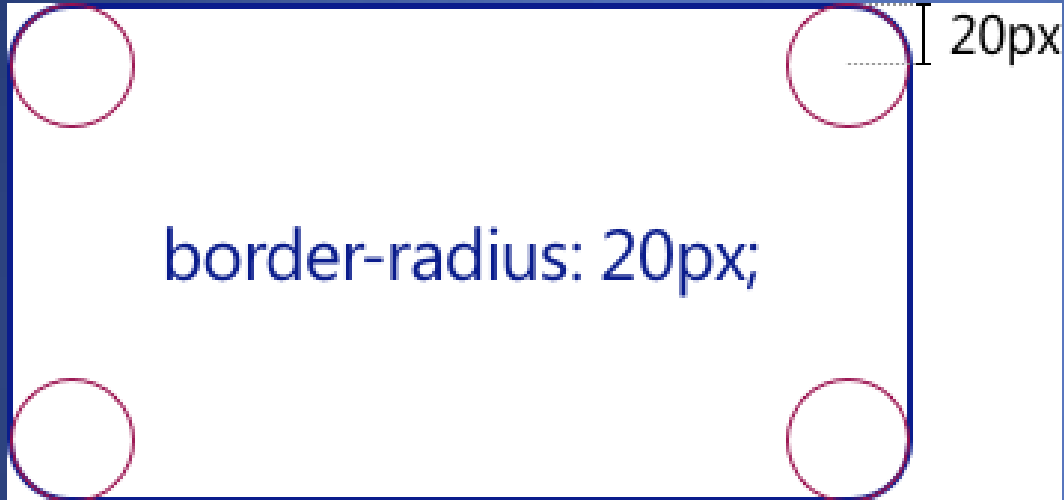
Footer

CSS3 문법

경계선(Border)

1. border-radius:20px

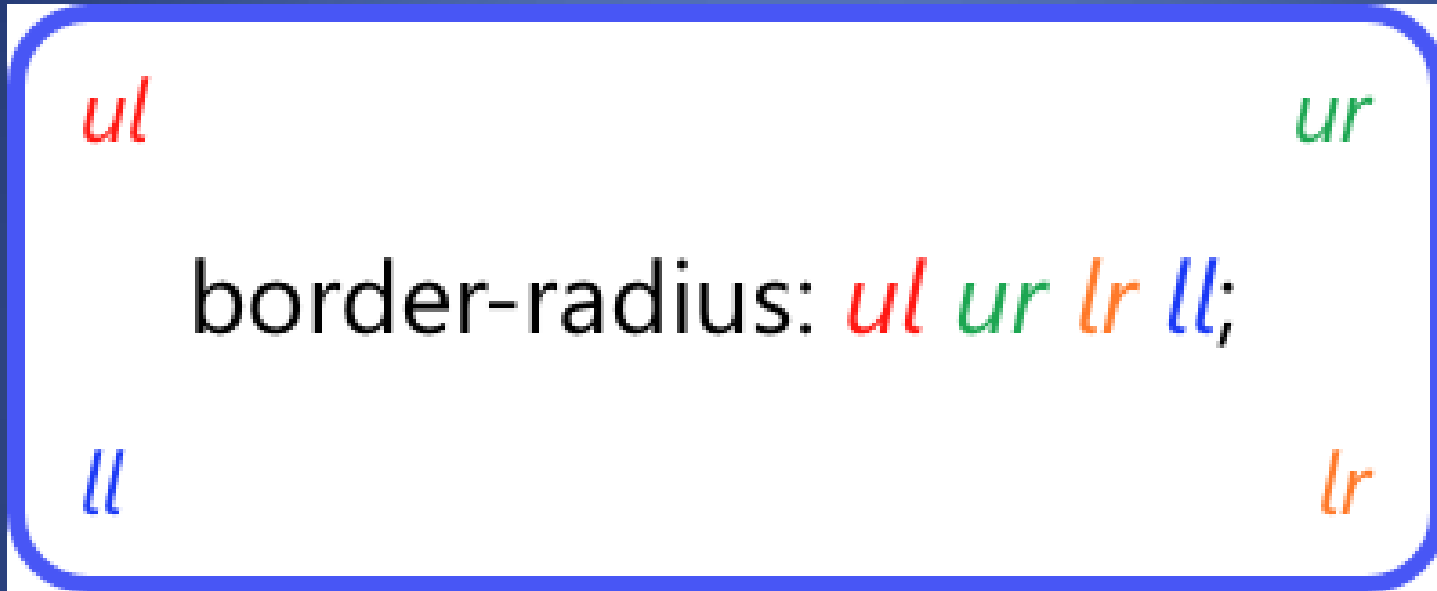
- 4군데(좌상단, 우상단, 우하단, 좌하단)의 모서리가 **동일한** 20px의 반지름을 가진 둥근 박스 표현



경계선(Border)

2. border-radius:10px 20px 30px 40px

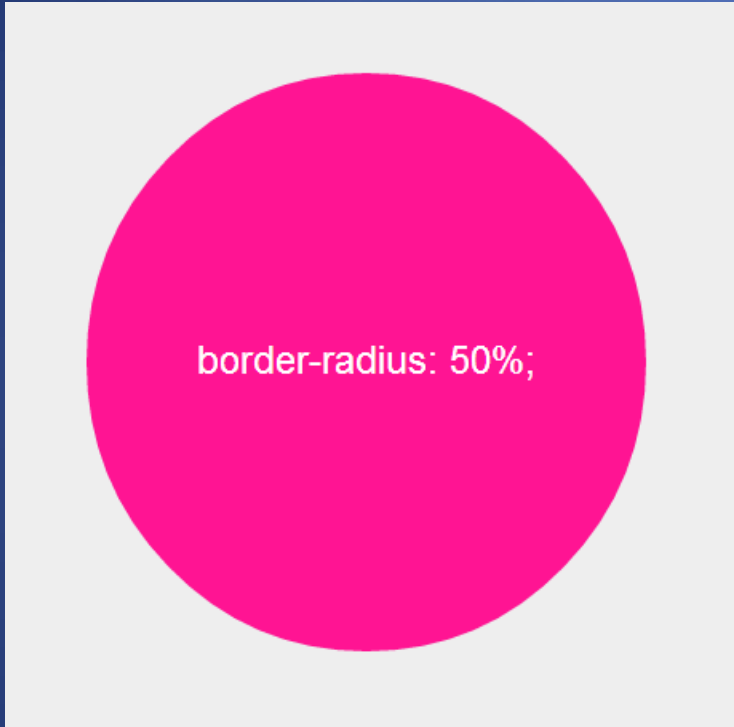
- 4군데(좌상단, 우상단, 우하단, 좌하단 순서)의 모서리가 서로 다른 반지름을 가진 둥근 박스 표현



경계선(Border)

3. **width:100px; height:100px; border-radius:50px;**

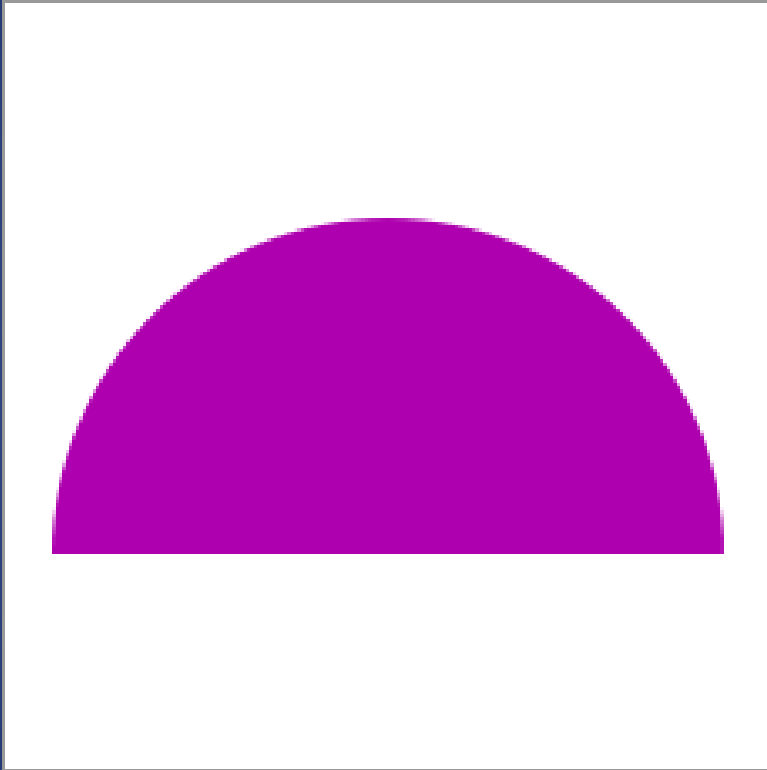
- border-radius값을 width와 height의 1/2로 지정하면 둥근 원형으로 표현 가능



경계선(Border)

4. **width:200px; height:100px; border-radius:100px 100px 0 0;**

- 가로가 긴 박스의 형태에서 border-radius값을 두 군데만 1/2로 지정하면 반원 표현 가능



경계선(Border)

5. `width:100px; height:100px; border-radius:200px 0 0 0;`

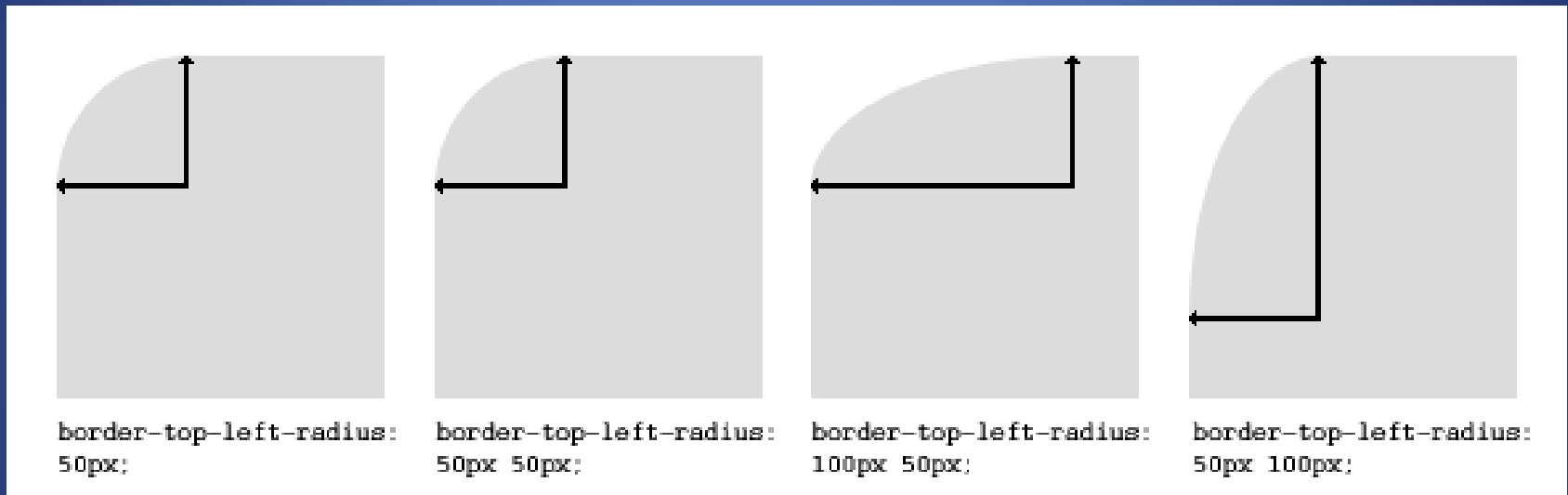
- 네 모서리 중 한쪽만 반지름을 width와 height 값의 2배로 지정하면 ¼ 형태의 원모양 표현 가능



경계선(Border)

6. `border-top-left-radius:50px;`와 `border-top-left-radius:100px 50px;`

- 네 모서리 중 한 쪽만 둥글게 처리가 가능한 CSS3의 개별속성
- `border-top-left-radius:100px 50px;`의 경우, 앞 속성값(100px)은 x축 방향의 radius(반지름) / 뒤 속성값(50px)은 y축 방향의 radius(반지름)를 지칭

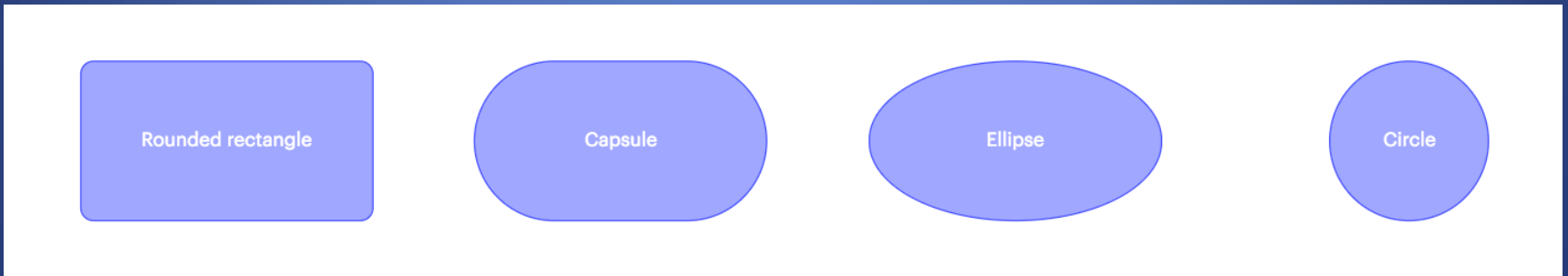


속성	설명
Border-top-left-radius	박스의 좌-상단 모서리 둥글게 지정
Border-top-right-radius	박스의 우-상단 모서리 둥글게 지정
Border-bottom-left-radius	박스의 좌-하단 모서리 둥글게 지정
Border-bottom-right-radius	박스의 우-하단 모서리 둥글게 지정

경계선(Border)

<<실습과제-#01>>

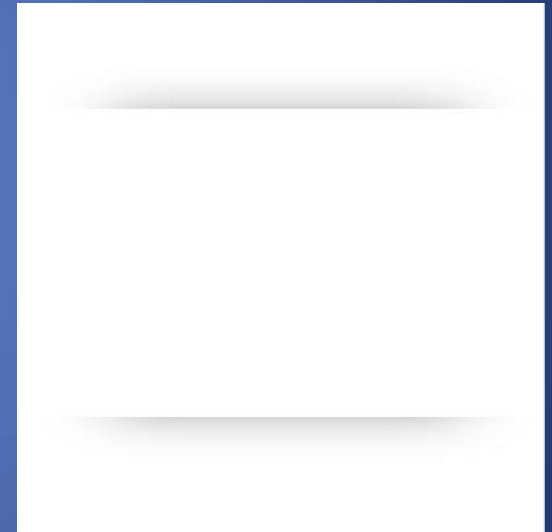
- 다음과 같은 모형을 단계별로 만드세요.(좌측부터... 내부 색상은 자유)
- 조건 01 : 1번 형태는 width:300px; height:100px; border-radius:10px;
- 조건 02 : 2번 형태는 width:300px; height:100px; 및 양 측면에 직선 구간 제거
- 조건 03 : 3번 형태는 width:300px; height:100px; 4 방위 직선구간 없는 타원 형태
- 조건 04 : 4번 형태는 width:100px; height:100px; 정 원 형태



경계선(Border)

7. box-shadow: ①가로방향 ②세로방향 ③스프레드 ④거리 ⑤색상

- 일반적으로 box-shadow의 형식은
- **box-shadow: ①가로방향 ②세로방향 ③스프레드 ④색상 (주로 사용하는 방식)**
- 박스 외곽으로 부여하는 그림자 효과 – box-shadow: 1px 1px 1px rgba(0,0,0,0.75);
- 박스 내부에 부여하는 그림자 효과 – box-shadow: inset 1px 1px 1px 1px rgba(0,0,0,0.75);



- 좌측과 같은 형태의 그림자 부여 가능
- width:400px; height:400px; box-shadow : 0px 0px 10px -5px rgba(10,10,10,0.75);
- box-shadow : **10px 0px 0px 0px rgba(255,255,255,1), -10px 0px 0px 0px**
rgba(255,255,255,1), 0px 0px 10px 0px rgba(0,0,0,1); /*오렌지색 속성값 추가 및 변경*/

배경(Background)

1. Gradient

① 선형 그라디언트

```
background: linear-gradient(left, #1e5799 0%,#7db9e8 100%);
```

- linear-gradient : 선형 방향으로 그라디언트를 속성값으로 설정
- left : 시작 위치를 지정
- 색상 코드와 위치 : 시작 위치를 기준으로 색상 부여

```
background: #1e5799; /* Old browsers */
```

```
background: -moz-linear-gradient(left, #1e5799 0%, #7db9e8 100%); /* FF3.6-15 */
```

```
background: -webkit-linear-gradient(left, #1e5799 0%,#7db9e8 100%); /* Chrome10-25,Safari5.1-6 */
```

```
background: linear-gradient(to right, #1e5799 0%,#7db9e8 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
```

```
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#1e5799', endColorstr='#7db9e8',GradientType=1 ); /* IE6-9 */
```

배경(Background)

1. Gradient

② 원형 그라디언트

```
background: radial-gradient(#1e5799 0%, #7db9e8 100%);
```

- radial-gradient : 원형 방향으로 그라디언트를 속성값으로 설정
- 색상 코드와 위치 : 가운데를 기준으로 기준으로 색상 부여
- 가로와 높이가 다를 경우, 타원의 형태로 표현
- 정원 형태로 맞출 경우 → radial-gradient(circle, #1e5799 0%, #7db9e8 100%);

```
background: #1e5799; /* Old browsers */
background: -moz-radial-gradient(center, ellipse cover, #1e5799 0%, #7db9e8 100%); /*
FF3.6-15 */
background: -webkit-radial-gradient(center, ellipse cover, #1e5799 0%,#7db9e8 100%); /*
Chrome10-25,Safari5.1-6 */
background: radial-gradient(ellipse at center, #1e5799 0%,#7db9e8 100%); /* W3C, IE10+,
FF16+, Chrome26+, Opera12+, Safari7+ */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#1e5799',
endColorstr='#7db9e8',GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
```

<Product by 안재욱>

배경(Background)

1. Gradient

③ 외부 그라디언트 툴 소스 가져오기

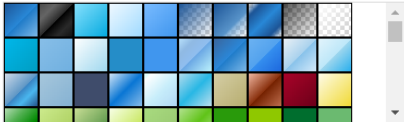
- <http://www.colorzilla.com/gradient-editor/> 접속하여 색상 및 방향 설정한 후, 우측 css 복사 및 지정 된 곳에 붙여넣기!!!

Ultimate CSS Gradient Generator


A powerful Photoshop-like CSS gradient editor from [ColorZilla](#).

For Firefox For Chrome Gradient Generator

Presets



Name:



hue/saturation reverse import css import image

Import from an image-based gradient
Analyzes the image and converts found gradient to CSS

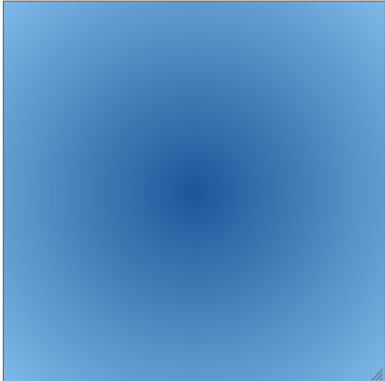
upload a gradient image:
 선택된 파일 없음

or specify gradient image URL:

Stops

Opacity: <input type="text" value="100"/>	Location: <input type="text" value="0"/>	<input type="button" value="delete"/>
Color: <input type="text"/>	Location: <input type="text" value="0"/>	<input type="button" value="delete"/>

Preview



Orientation: Size: x ☐ IE

CSS [switch to scss](#)

```
/* Permalink - use to edit and share this gradient :  
http://colorzilla.com/gradient-editor/#1e5799+0,7db9e8+100 */  
background: #1e5799; /* Old browsers */  
background: -moz-radial-gradient(center, ellipse cover, #1e5799 0%, #7db9e8 100%); /* FF3.6-15 */  
background: -webkit-radial-gradient(center, ellipse cover, #1e5799 0%, #7db9e8 100%); /* Chrome10-25,Safari5,1-6 */  
background: radial-gradient(ellipse at center, #1e5799 0%, #7db9e8 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */  
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#1e5799', endColorstr='#7db9e8', GradientType=1 ); /* IE6-9 fallback on horizontal gradient */
```

Color format: ☒ Comments

[Link to, save or share the current gradient using its permalink.](#)

배경(Background)

2. background-size

- ① background-size : 200px 100px;
 - 배경 이미지 가로 200px, 세로 100px
- ② background-size : 150px;
 - 배경 이미지 가로, 세로 150px
- ③ background-size : contain;
 - 배경 이미지를 박스의 세로 크기에 맞춤
- ④ background-size : cover;
 - 배경 이미지를 박스의 가로 크기에 맞춤
 - 반응형웹 제작시 메인 이미지 파트에서 주로 사용

배경(Background)

3. background-origin

배경 이미지의 영역을 설정

- ① background-origin : border-box;
 - 박스 테두리 경계 부분을 포함한 경계선까지
- ② background-origin : padding-box;
 - 박스 테두리 경계를 제외한 패딩 부분까지
- ③ background-origin : contents-box;
 - 패딩 부분을 제외한 콘텐츠 본인의 영역까지

배경(Background)

4. Multi-background

배경에 여러 장의 이미지를 한꺼번에 넣을 수 있는 기능

`background-image : url('파일명.파일 형식'), url('파일명.파일 형식'), url('파일명.파일 형식');`

- 세 개의 이미지를 지정

`background-repeat : no-repeat, repeat-x, no-repeat;`

- 세 개의 이미지 중 두 번째 이미지만 x축으로 반복 설정

`background-position : top, bottom, center;`

- 세 개의 이미지 위치를 지정

`background-attachment : fixed, fixed, fixed;`

- 세 개의 이미지를 모두 화면에 고정

배경(Background)

4. Multi-background

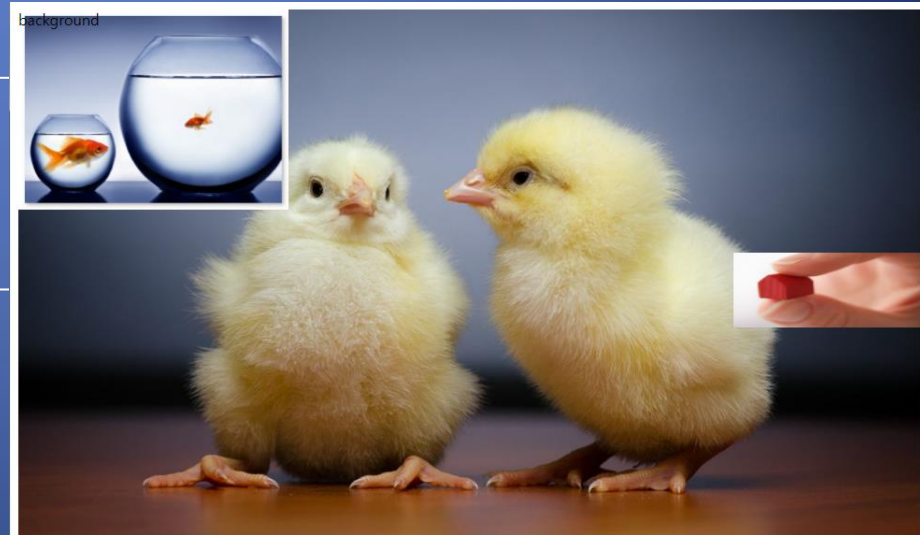
<<CSS>>

```
div{  
  width:100%; height:600px;  
  background-image:url('img/img-01.jpg'), url('img/img-02.jpg'), url('img/img-03.jpg');  
  background-repeat:no-repeat, no-repeat, no-repeat;  
  background-position: left top, right 50%, 50% 100%;  
  background-size: 30%, 200px 80px, cover;  
}
```

<<html>>

```
<body>  
  <div>multi-background</div>  
</body>
```

<<출력 결과>>



배경(Background)

5. background-clip

배경 이미지의 영역을 설정

① background-clip : border-box;

- 배경색이 테두리 영역까지 색상(또는 이미지) 적용

② background-clip : padding-box;

- 배경색이 패딩 영역까지 색상(또는 이미지) 적용

③ background-clip : contents-box;

- 배경색이 콘텐츠 영역까지만 색상(또는 이미지) 적용

텍스트 효과(Text Effect)

1. text-shadow : ①가로방향 ②세로방향 ③스프레드 ④색상

이미지 편집 툴 사용하지 않고 문자에 그림자 효과를 부여할 수 있음

<<CSS>>

```
.main-title{  
  font-size : 32px;  
  text-shadow : 2px 2px 4px rgba(0,0,0,0.85);  
  color:#777;  
}
```

<<html>>

```
<body>  
  <h2 class="main-title">Experience Invest</h2>  
</body>
```

텍스트 효과(Text Effect)

2. text-gradient

이미지 편집 툴 사용하지 않고 텍스트에 그라데이션 효과를 부여할 수 있음.
(단, IE에서는 적용 불가)

<<CSS>>

```
.main-title{  
    background:-webkit-linear-gradient(#eee, #333);  
    -webkit-background-clip: text;  
    -webkit-text-fill-color: transparent;  
}
```

<<html>>

```
<body>  
    <h2 class="main-title"> Experience Invest</h2>  
</body>
```

텍스트 효과(Text Effect)

3. word-wrap

문단에서 줄바꿈의 기준이 글자 또는 단어 시점에서 적용하는가에 대한 컨트롤

<<CSS>>

```
.text-box{width:120px; height:400px;}
.main-content-text{
  word-wrap: break-word;
  font-size:18px;
  font-weight:bold;
}
.sub-content-text{
  word-wrap: keep-all;
  font-size:16px;
  font-weight:normal;
}
```

<<html>>

```
<body>
  <div class="text-box">
    <p class="main-content-text">Japan's Hayabusa 2 will follow in the footsteps
of its pioneering predecessor</p>
    <p class="sub-content-text">일본의 하야부사2는 선구자의 발자취를 따라 간다.</p>
  </div>
</body>
```

텍스트 효과(Text Effect)

4. word-break

단어의 중단점 기준이 글자 또는 단어 시점에서 적용하는가에 대한 컨트롤

<<CSS>>

```
.text-box{width:120px; height:400px;}
.main-content-text{
    word-break: break-word;
    font-size:18px;
    font-weight:bold;
}
.sub-content-text{
    word-break: keep-all;
    font-size:16px;
    font-weight:normal;
}
```

<<html>>

```
<body>
  <div class="text-box">
    <p class="main-content-text">Japan's Hayabusa 2 will follow in the footsteps
    of its pioneering predecessor</p>
    <p class="sub-content-text">일본의 하야부사2는 선구자의 발자취를 따라 간다.</p>
  </div>
</body>
```

<Product by 안재욱>

텍스트 효과(Text Effect)

5. word-break와 word-wrap의 조합

단어 마다 중단점을 부여하면서, 줄 바꿈이 일어나도록 하기 위함

< <CSS> >

```
.text-box{width:120px; height:400px;}
.main-content-text{
    word-break: keep-all;
    word-wrap: break-word;
    font-size:18px;
    font-weight:bold;
}
```

< <html> >

```
<body>
  <div class="text-box">
    <p class="main-content-text"> Meet me on the bridge: Discovering the truth about
    my parents after 20 years
    ##### abcdefghijklmnopqrstuvwxyz!!! 나
    라말씀이 중국과 달라.....</p>
  </div>
</body>
```

<Product by 안재욱>

텍스트 효과(Text Effect)

6. word-spacing

단어 사이의 간격을 조정. 단위는 px.

word-spacing:10px;

7. letter-spacing

글자 사이의 간격(자간)을 조정. 단위는 px.

letter-spacing:-1px;

8. white-space

줄바꿈 여부를 결정.

- normal(defalut) : 글자 줄이 자동변경. 좁은 공간에서 적용시 높이가 변경될 수 있음.
- nowrap : 줄 바꿈이 실행되지 않음.
- pre : 줄 바꿈과 기타 공백이 유지 !DOCTYPE 선언에서 표준 준수 모드를 지정할 경우에 지원. !DOCTYPE 선언에서 표준 준수 모드를 지정하지 않으면 이 값을 검색할 수는 있지만 렌더링에 영향을 주지 않으며 normal 값처럼 동작.
- pre-line : 줄 바꿈 시퀀스가 유지
- pre-wrap : 줄 바꿈 시퀀스가 축소
- inherit : 부모 요소로부터 값을 상속 받는다.

변형(transform)

1. transform

박스를 회전(rotate), 확대 및 축소(scale), 이동(translate), 왜곡(skew)를 수행

- ① transform : rotate(45deg);
 - 박스를 시계방향으로 45도 회전
- ② transform : scale(1.1);
 - 박스의 크기를 X축과 Y축 동일하게 110%로 변형
 - 박스의 크기를 각각 다르게 변형할 때, transform : scale(1.1, 1.2);
- ③ transform : translate(10px, -20px);
 - 박스의 위치를 x축으로 10px, y축으로 -20px 이동
- ④ transform : skew(30deg, 45deg);
 - 박스를 가로 30도, 세로 45도로 왜곡

크로스 브라우징을 위한 각 브라우저별 반드시 5가지 css 모두 적용할 것~!

(transform, -webkit-transform, -moz-transform,
-ms-transform, o-transform)

변형(transform)

1. transform

박스를 회전(rotate), 확대 및 축소(scale), 이동(translate), 왜곡(skew)를 수행

- ⑤ transform : rotateX(180deg);
 - 박스를 X축 방향으로 180도 회전
- ⑥ transform : rotateY(180deg);
 - 박스를 Y축 방향으로 180도 회전
- ⑦ transform : translateX(20px);
 - 박스를 X축 방향으로 20px 이동
- ⑧ transform : translateY(20px);
 - 박스를 Y축 방향으로 20px 이동

크로스 브라우징을 위한 각 브라우저별 반드시 5가지 css 모두 적용해야 함!

**(transform, -webkit-transform, -moz-transform,
-ms-transform, o-transform)** <Product by 안재욱>

변형(transform)

2. filter

요소가 표시되기 전에 요소 렌더링시 흐림 또는 색상 이동과 같은 효과를 제공

filter:필터속성(속성:값,속성:값,속성:값); 또는 **filter:필터속성();**

① 필터속성

- blur(px) : 흐려지게 하는 효과
 - 사용예시 - **filter : blur(5px);**
 - 괄호 () 안에 기입을 하지 않을 경우, 0px로 사용됨
- brightness(%) : 밝기를 조정하는 효과
 - 사용예시 - **filter : brightness(100%);**
 - 0%일 경우, 이미지를 검정으로 만듦
 - 100%일 경우, 원본 이미지 밝기 유지
 - 100% 이상일 경우, 더 밝은 상태 만듦
- contrast(%) : 대비 효과
 - 사용예시 - **filter : contrast(200%);**
 - 0%일 경우, 이미지를 검정으로 만듦
 - 100%일 경우, 원본 이미지 대비 유지
 - 100% 이상일 경우, 더 많은 대비를 만듦

변형(transform)

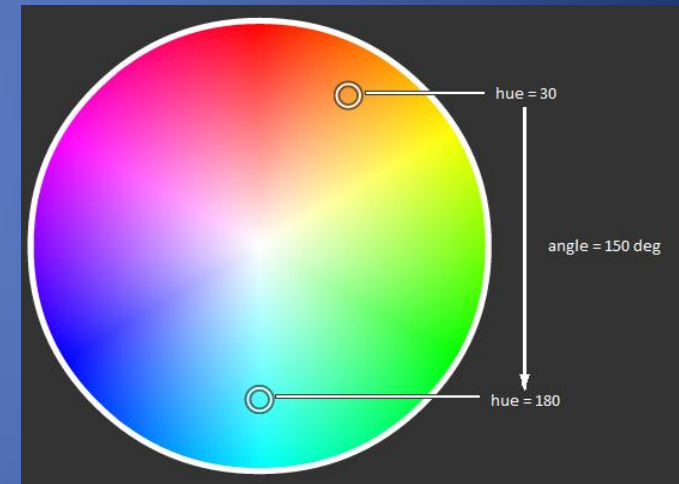
2. filter

요소가 표시되기 전에 요소 렌더링시 흐림 또는 색상 이동과 같은 효과를 제공

filter:필터속성(속성:값,속성:값,속성:값); 또는 **filter:필터속성();**

① 필터속성

- drop-shadow(*x축-px y축-px 스프레드-px 컬러*) : drop shadow 효과 부여
 - 사용예시 - **filter : drop-shadow(8px 8px 10px red);**
- grayscale(%) : 이미지를 흑/백 톤으로 변경
 - 사용예시 - **filter : grayscale(100%);**
 - 0%일 경우, 원본 색상 유지
 - 100%일 경우, 완전한 그레이로 전환
- hue-rotate(deg) : HSL에서 이미지 색상을 변화
 - 사용예시 - **filter : hue-rotate(150deg);**
 - 0deg, 360deg일 경우, 원본색상
 - 다른 값의 경우, 우측 색상환을 기준, 색상변경



변형(transform)

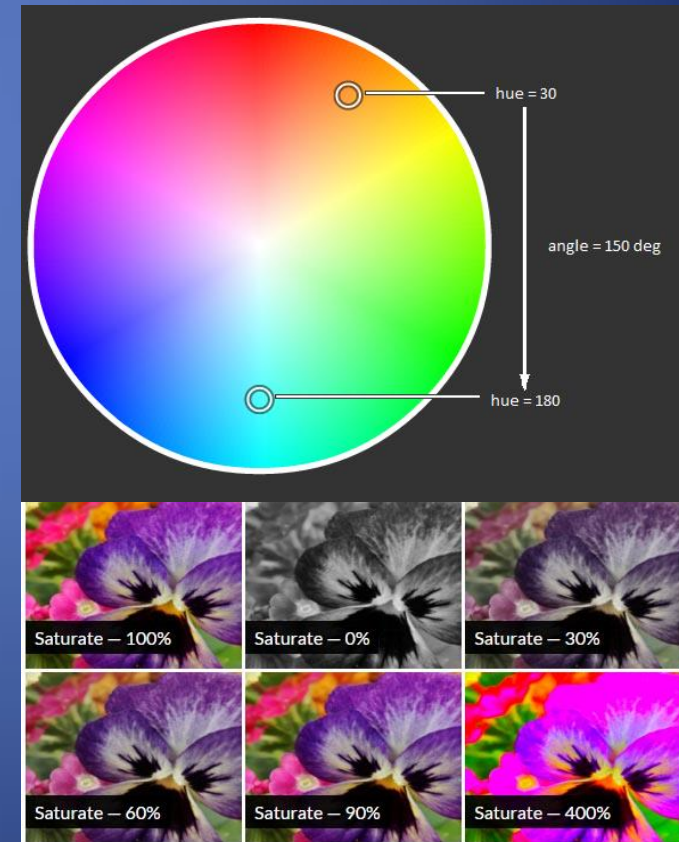
2. filter

요소가 표시되기 전에 요소 렌더링시 흐림 또는 색상 이동과 같은 효과를 제공

filter:필터속성(속성:값,속성:값,속성:값); 또는 **filter:필터속성();**

① 필터속성

- invert(%) : 색상을 반전시키는 스타일 속성
 - 사용예시 - **filter : invert(100%);**
- opacity(%) : 이미지에 투명도를 부여
 - 사용예시 - **filter : opacity(50%);**
 - 0%일 경우, 이미지 노출 없음. 단 위치는 유지
 - 100%일 경우, 원본 이미지
- saturate(%) : HSL에서 이미지 채도를 변화
 - 사용예시 - **filter : saturate(40%);**
 - 0%일 경우, 채도 없음
 - 100% 일 경우 원본 채도



변형(transform)

2. filter

요소가 표시되기 전에 요소 렌더링시 흐림 또는 색상 이동과 같은 효과를 제공

filter:필터속성(속성:값,속성:값,속성:값); 또는 filter:필터속성();

① 필터속성

- sepia(%) : 세피아로 전환
 - 사용예시 - **filter : sepia(80%);**
 - 0%일 경우, 원본 이미지
 - 100%일 경우, 세피아 이미지로 전환
- none : 필터 적용 없음
 - 사용예시 - **filter : none;**

전환(transition)

1. transition : ①전환대상 ②적용시간 ③속도 변화

각 속성(또는 전체)에 애니메이션을 적용하여 모션을 느낄 수 있는 CSS 속성

① 전환대상

- background, width, height, color, opacity 등의 구성 대상을 적용 가능 (전체 일 경우 all)

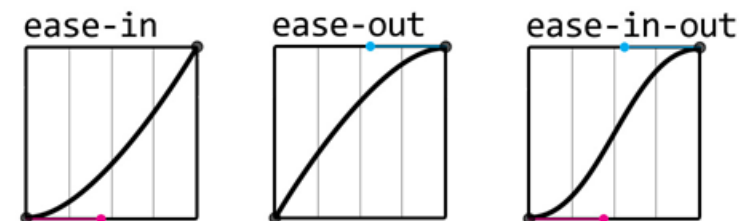
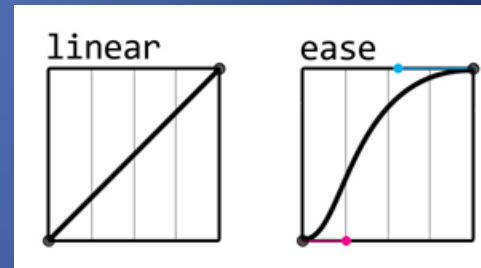
※ 단, display의 경우, 적용이 되지 않음

② 적용시간

- 시작부터 끝까지 적용되는 시간을 작성

③ 속도변화

- linear : 적용되는 속도가 동일하게 함
- ease : 느리게 시작했다가 도달 시점에서 다시 빨라짐
- ease-in : 점점 빨라짐
- ease-out : 점점 느려짐
- ease-in-out : 처음과 끝이 느림. 중간 시간에서는 대체로 동일하게 진행됨
- cubic-bezier(n,n,n,n) : 입력된 값으로 속도를 조작



전환(transition)

2. transition-delay : 0.8s

- 0.8초 후 작동 하도록 설정

3. transition : background 0.5s ease-in, height 0.4s linear;

- 각 속성마다 다양한 전환을 적용할 수 있음.

보여주기 / 감추기(show/hide)

1. opacity

- 투명도 부여
- 속성값은 0~1까지. 소수점 부여 가능
- 하위 자식 요소에게 모두 상속됨. 이를 피하기 위해서는 rgba(0,0,0,0)으로 작성해야 함.

```
<style>
  .main-bg{width:800px; height:600px; background:url('img/img-002.jpg') no-repeat
  50% 50%;}
  .sub-bg-01{width:300px; height:400px; float:left; background:#999; opacity:0.5;}
  .sub-bg-02{width:300px; height:400px; float:left; background:rgba(175, 175, 175, 0.5);}
</style>

<body>
  <div class="main-bg">
    <div class="sub-bg-01"><p>제가 강조하고 싶은 것은 미소짓는 것이 어려울 때
    일수록 서로에게 미소로 대해야 한다는 것입니다.</p></div>
    <div class="sub-bg-02"><p>제가 강조하고 싶은 것은 미소짓는 것이 어려울 때
    일수록 서로에게 미소로 대해야 한다는 것입니다.</p></div>
  </div>
</body>
```

보여주기 / 감추기(show/hide)

2. visibility

- visible: 보임
- hidden: 숨김 (자신의 영역은 계속 차지)
- collapse: 겹치도록 지정(테이블의 행과 열 요소만 지정할 수 있으며, 그 외 요소의 지정하면 hidden으로 해석)
- inherit: 부모 요소의 값을 상속

3. display

- none: 보이지 않음
- none을 제외한 속성값: 보임

※ visibility와 display의 차이점

- v: 공간이 유지됨
- d: 공간 자체가 사라짐



다단 편집(multi-column)

1. **column-count : 4;**

- 박스의 내용을 4단으로 구성

2. **column-gap : 18px;**

- 각 단과 단 사이의 간격을 18px로 띄움

3. **column-rule : 2px dashed #999;**

- 각 단과 단 사이의 구분선을 구성

다단 편집(multi-column)

```
<style>
```

```
.bx-group{width:720px; min-  
height:200px; column-count:3;}  
.bx{width:200px; margin:10px 20px;  
float:left; /*display: inline-block;*/}  
.bx-01{height:100px;  
background:#ff3300;}  
.bx-02{height:160px; background:#ffff00;}  
.bx-03{height:180px;  
background:#ff3300;}  
.bx-04{height:200px; background:#ffff00;}  
.bx-05{height:120px;  
background:#ff3300;}  
.bx-06{height:160px; background:#ffff00;}  
.bx-07{height:180px;  
background:#ff3300;}  
.bx-08{height:200px; background:#ffff00;}  
.bx-09{height:120px;  
background:#ff3300;}
```

```
</style>
```

```
<body>
```

```
<div class="bx-group">  
  <div class="bx bx-01">1</div>  
  <div class="bx bx-02">2</div>  
  <div class="bx bx-03">3</div>  
  <div class="bx bx-04">4</div>  
  <div class="bx bx-05">5</div>  
  <div class="bx bx-06">6</div>  
  <div class="bx bx-07">7</div>  
  <div class="bx bx-08">8</div>  
  <div class="bx bx-09">9</div>
```

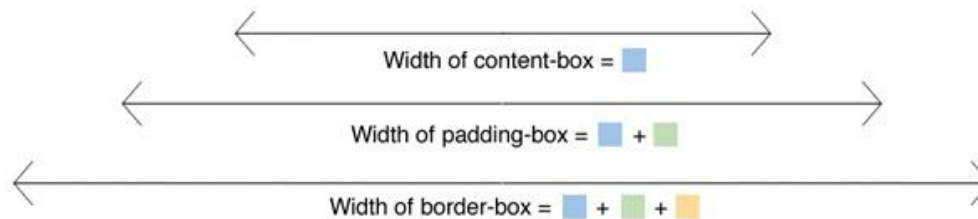
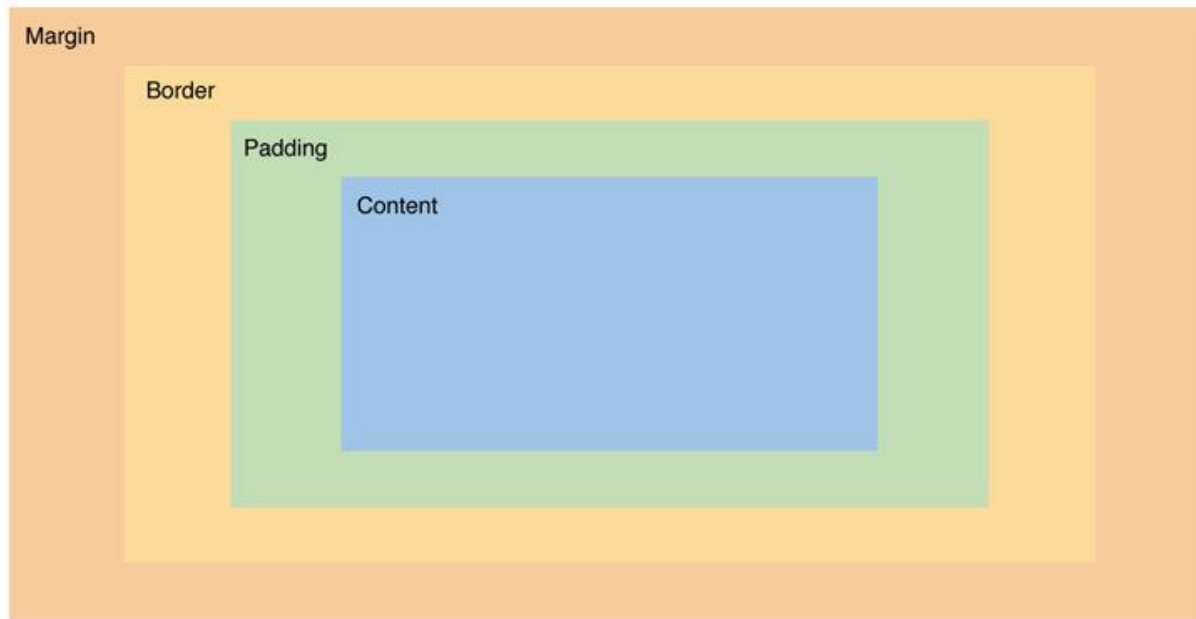
```
</div>
```

```
</body>
```

사용자 환경(user interface)

1. box-sizing

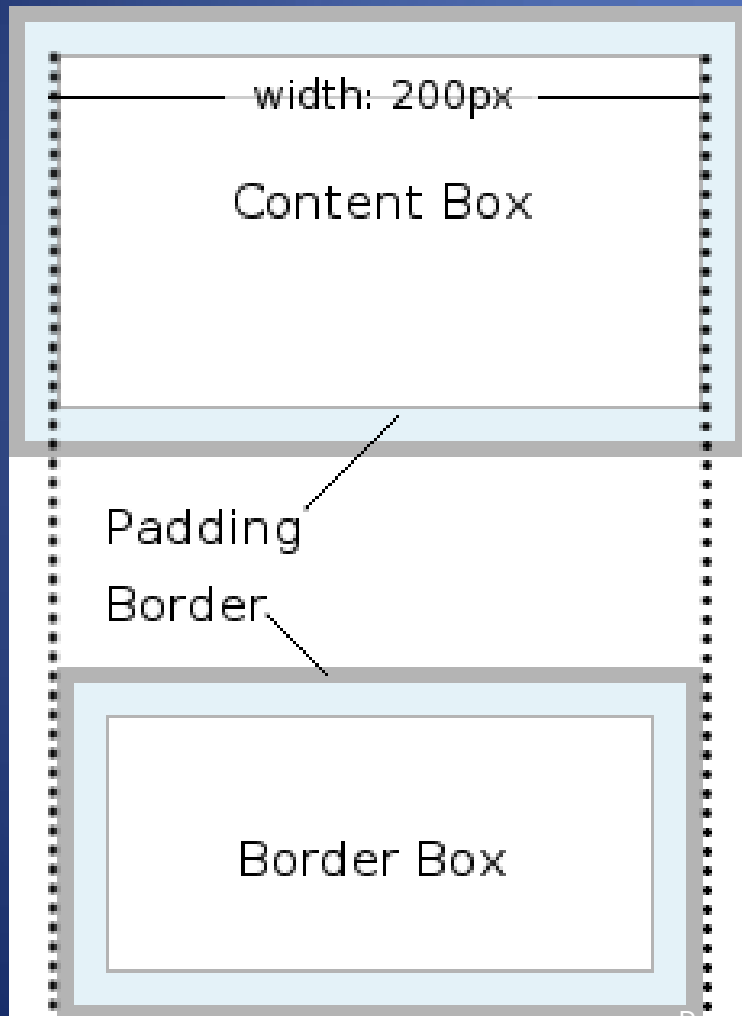
- box-sizing : content-box; border가 박스 크기에 포함되지 않음
- box-sizing : border-box; border가 박스 크기에 포함됨



사용자 환경(user interface)

1. box-sizing

- box-sizing : content-box; border가 박스 크기에 포함되지 않음
- box-sizing : border-box; border가 박스 크기에 포함됨



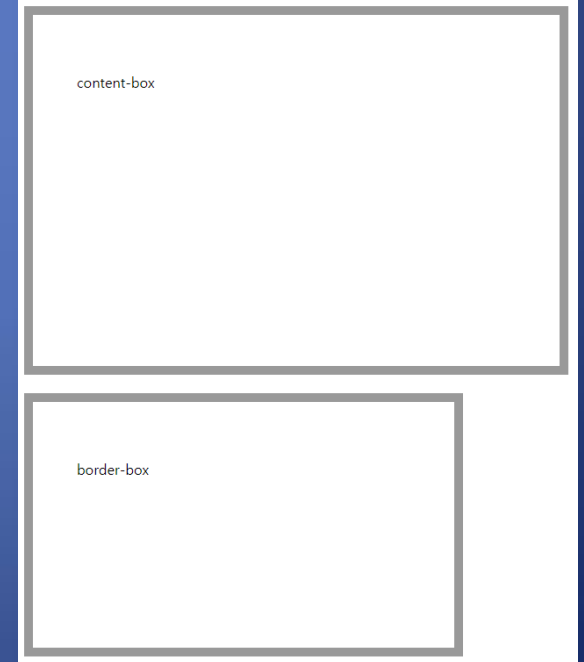
사용자 환경(user interface)

1. box-sizing

- box-sizing : content-box; border가 박스 크기에 포함되지 않음
- box-sizing : border-box; border가 박스 크기에 포함됨

```
<style>
  .content-bx{width:500px; height:300px; padding:50px; border:10px solid #999;
  box-sizing:content-box;}
  .border-bx{width:500px; height:300px; padding:50px; border:10px solid #999;
  box-sizing:border-box;}
</style>

<body>
  <div class="content-bx">
    <p>content-box</p>
  </div>
  <br/>
  <div class="border-bx">
    <p>border-box</p>
  </div>
</body>
```



사용자 환경(user interface)

2. resize

- 박스 형태의 모서리를 조정 가능(<textarea> 태그의 속성 중 resize와 유사한 형태)
- 가로 : horizontal, 세로 : vertical, 양쪽 : both

3. outline

- border를 감싸는 테두리
- outline-offset : border와 outline의 사이 간격
- outline-width : outline의 두께
- outline-style : outline의 형태(solid, dashed, dotted)
- outline-color : outline의 색상
- **outline: *outline-width outline-style outline-color***(또는 initial, inherit);

4. user-select

- 사용자가 선택 유무를 조정함 (텍스트 또는 이미지)
- user-select : none; 일 경우, 사용자가 텍스트 또는 이미지를 Mouse Drag 불가능
- 아래와 같이 함께 사용

```
-webkit-user-select: none; /* Safari 3.1+ */  
-moz-user-select: none; /* Firefox 2+ */  
-ms-user-select: none; /* IE 10+ */  
user-select: none; /* Standard syntax */
```