

웹 디자인을 실현해 주는 CSS

□ 왜 CSS를 사용해야 할까?

- 웹 문서의 디자인과 내용을 분리합니다.
- 다양한 매체에 적합한 문서를 만들 수 있습니다.

□ 스타일의 형식

```
p { color:blue; }  
  
h2 {  
    font-size:20px;  
    color:purple;  
}
```

□ 스타일 위치

1) HTML 문서의 <head> 태그 안에 선언

```
<!doctype html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <title> 요안도라 </title>  
    <style>  
        body {  
            background-color:#333; /* 배경색은 짙은 회색 */  
            color:#fff; /* 글자색은 흰색 */  
        }  
    </style>  
</head>  
<body>  
    <h2>예약 방법 및 요금</h2>  
    <p>요안도라 예약의 가장 빠른 방법은?</p>  
</body>  
</html>
```

2) 외부 스타일시트 파일을 불러서 사용

```
<!doctype html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <title> 요안도라 </title>  
    <link href="style.css" rel="stylesheet" type="text/css"></link>  
</head>  
<body>  
    <h2>예약 방법 및 요금</h2>  
    <p>요안도라 예약의 가장 빠른 방법은?</p>  
</body>  
</html>
```

□ 스타일의 우선순위

- 1) 스타일은 상속된다. : HTML 문서의 <body>태그에 스타일을 적용하면 <body>와 </body>사이의 모든 태그에 스타일이 적용된다.
- 2) 스타일은 단계적으로 적용된다. : 스타일이 중복되면 맨 마지막 스타일이 적용된다.
단, 스타일 규칙에 !important를 추가하면 해당 스타일이 가장 높은 순위를 갖게 된다.

□ 선택자

1) 전체 선택자 : 모든 요소에 스타일 적용, *(별표)를 선택자로 사용

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

2) 태그 선택자 : 특정 태그에 스타일 적용, 태그 이름을 선택자로 사용

```
p {
  font-size:12px;
  font-family:"돋움";
}
```

3) 클래스 선택자 : 특정 부분에 스타일 적용, .(마침표)를 선택자로 사용

```
.redaccent { color:red; /* 글자색 빨강 */ }
h3.bluetxt { color:blue; }
```

4) id 선택자 : 특정 부분에 스타일 적용, #을 선택자로 사용

```
#title{
  clear:both; /* 이전 요소와 줄바꿔서 표시 */
  float:left; /* 왼쪽으로 플로팅 */
  margin-left:10px; /* 왼쪽 여백 10px */
  width:200px; /* 너비 200px */
  height:110px; /* 높이 110px */
  background-color:orange; /* 배경색 오렌지색 */
  border:1px solid black; /* 테두리 1px짜리 검은색 실선 */
  text-align:center; /* 텍스트 가운데 정렬 */
}
```

5) 하위 선택자 : 모든 하위 요소에 스타일 적용,

```
section p { color:blue; } /* <section> 태그 안에 있는 모든 <p> 태그의 글자색을 파란색으로 지정 */
#container ul { border:1px dotted black; } /* id가 container인 태그의 모든 <ul> 태그에 스타일 지정 */
```

6) 자식 선택자 : 자식 요소에만 스타일 적용

```
section > p { color:blue; } /* <section> 태그 안에 있는 오직 자식 <p> 태그만 스타일 지정(손자 태그는 안 됨) */
#container > ul { border:1px dotted black; } /* id가 container인 태그의 오직 자식 <ul> 태그만 스타일 지정(손자 태그는 안 됨) */
```

7) 인접 형제 선택자 : 첫 번째 동생 요소에 스타일 적용,

```
h1 + p { color:blue; } /* <h1> 태그 다음에 나오는 <p> 태그만 스타일 지정 */
```

8) 형제 선택자 : 형제 요소에 스타일 적용,

```
h1 ~ p { color:blue; } /* <h1> 태그와 형제인 모든 <p> 태그에 스타일 지정 */
```

9) 그룹 선택자 : 스타일 한꺼번에 정의

```
h1, h2, p { color:blue; } /* <h1>, <h2>, <p> 태그 모두에 스타일 지정 */
```

□ 속성 선택자

1) [속성] 형식 : 지정한 속성을 지니고 있는 태그에 스타일 지정

```
a[href] {
    background:yellow; /* <a>태그 중 href 속성이 있는 것에만 스타일 지정 */
}
```

2) [속성 = 값] 형식 : 주어진 속성과 값이 일치하는 태그에 스타일 지정

```
a[target="_blank"] {
    background:yellow; /* <a>태그 중 target 속성이 _blank 인 것에만 스타일 지정 */
}
```

3) [속성 ~= 값] 형식 : 주어진 속성의 값에 주어진 값이 포함(공백으로 구분)되어있는 태그에 스타일 지정

```
h4[title ~= "4인실"] { /* title 속성에 "4인실" 값이 있는 h4 요소 */
    font-style:italic; /* 글자를 이탤릭체로 */
}

<h4 title="4인실">남자 도미토리</h4>
<h4 title="욕실">여자 도미토리</h4>
<h4 title="4인실 욕실">펜션</h4>
```

4) [속성 ^= 값] 형식 : 주어진 속성의 값이 주어진 값으로 시작하는 태그에 스타일 지정

```
h4[title ^= "남자"] { /* title 속성 값이 "남자"로 시작하는 h4 요소 */
    background :url(man.png) no-repeat; /* man.png를 배경 이미지로 */
    padding: 10px 30px; /* h4 요소의 패딩(안여백) 상하 10px 좌우 30px */
}

<h4 title="남자 도미토리">남자 도미토리</h4>
<h4 title="여자 도미토리">여자 도미토리</h4>
<h4 title="펜션">펜션</h4>
```

5) [속성 \$= 값] 형식 : 주어진 속성의 값이 주어진 값으로 끝나는 태그에 스타일 지정

```
a[href $= ".hwp"] { /* 연결한 파일의 확장자가 hwp인 링크 */
    background: url(hwp_icon.gif) center right no-repeat; /* 배경으로 hwp 아이콘 표시 */
    padding-right:25px; /* 아이콘을 표시할 수 있도록 오른쪽에 25px 여백 */
}

<ul>
<li><a href="intro.hwp">hwp 파일</a></li>
<li><a href="intro.xls">엑셀 파일</a></li>
</ul>
```

6) [속성 |= 값] 형식 : 주어진 속성의 값이 주어진 값으로 시작하는 태그에 스타일 지정

7) [속성 *= 값] 형식 : 주어진 속성의 값에 주어진 값이 포함되어 있는 태그에 스타일 지정

```
a[hreflang |= "en"] { /* hreflang 속성이 en일 경우 */
    background:url(en.jpg) center right no-repeat; /* 오른쪽에 en.jpg 표시 */
    padding-right:30px; /* 배경 그림과 내용 사이의 여백*/
}

li[title *= "2011"] { /* title 속성 값에 '2011'이란 문자열이 있을 경우 */
    font-style:italic; /* 글자는 이탤릭체로 */
    font-weight:700; /* 글자 조금 굵게 */
}

<ul>
<li title="last update:2011-11"><a href="#" hreflang="en">사이트1</a></li>
<li title="last update:2010-01"><a href="#" hreflang="en">사이트2</a></li>
<li title="last update:2010-08"><a href="#" hreflang="ko">사이트3</a></li>
<li title="last update:2011-01"><a href="#" hreflang="fr">사이트4</a></li>
</ul>
```

□ 가상 클래스

1) 링크와 관련된 가상 클래스 선택자

```
:link
:visited
:hover
:focus
```

```
.navi a, .navi a:visited {
padding: 10px 5px 5px 35px;
display: block;
color:#fff;           /* 글자색 흰색 */
width: 150px;         /* 너비 150px */
text-decoration: none; /* 밑줄 없음 */
}

.navi a:hover, .navi a:focus {
text-shadow:0px 2px 2px #000; /* 글자에 검은색 그림자 */
color:#FC0; /* 글자가 노랑 계열로 */
}

.navi a:active {
color:red; /* 글자색이 빨강으로 */
}

<nav class="navi">
<ul>
<li><a href="desk.html">이음 안내</a></li>
<li><a href="room.html">객실 소개</a></li>
<li><a href="intro.html">예약 방법 및 요금</a></li>
<li><a href="reservation.html">예약하기</a></li>
</ul>
</nav>
```

2) 위치를 기준으로 하는 가상 클래스 선택자

```
:root
:nth-child(n)      :nth-last-child(n)      : n번째 자식 요소와 끝에서 n번째 자식 요소
:nth-of-type(n)   :nth-last-of-type(n)   : n번째 요소와 끝에서 n번째 요소
:first-child      :last-child            : 첫 번째 자식 요소와 마지막 자식 요소
:first-of-type    :last-of-type          : 첫 번째 요소와 마지막 요소
:only-child       :only-of-type          : 자식이 1개일 때 자식요소, 요소가 1개인 요소
:target
:enabled          :disabled              : 사용할 수 있을 때/없을 때, 체크되었을 때
:checked
```

```
/* 첫번째 항목의 스타일 */
ul.navi li:first-child {
border-top-left-radius: .5em; /* 왼쪽 상단 코너를 부드럽게 */
border-bottom-left-radius: .5em; /* 왼쪽 하단 코너를 부드럽게 */
}

/* 마지막 항목의 스타일 */
ul.navi li:last-child {
border-top-right-radius: .5em; /* 오른쪽 상단 코너를 부드럽게 */
border-bottom-right-radius: .5em; /* 오른쪽 하단 코너를 부드럽게 */
}
```

```
article.intro p:nth-child(2) {
border:1px solid black; /* 1px 검은색 실선 */
margin:10px; /* 마진(바깥여백) 10px */
padding:15px; /* 패딩(안여백) 15px */
background-color:lightgreen; /* 배경색 연두색 */
}
```

```

}
article.intro p:last-of-type {
width:80%; /* 너비는 부모 요소의 80% */
padding:15px;
margin-left:15%;
background:yellow; /* 배경색 노랑 */
color:blue; /* 글자색 파랑 */
}
<article class="intro">
<p>요안도라는 게스트 하우스(Guest House) 형식의 농어촌 민박입니다</p>
<p>성산의 날씨는 다음주 내내 높은 구름에 햇살 가득이라고 합니다. 목요일이면 섭씨 27도까지 오른다고 하지만, 늘 부는 바람이 쾌적한 균형을 잡아 마당에 마당에 나가 앉아 있는 시간이 많아질듯 합니다.</p>
<p>오늘은 사진에 보이는 긴 돌담을 따라 들어오는 요안도레 올레 입구에 특근색의 대문을 달았습니다.</p>
<p>내일은 두달 여동안 밖거리에 만든 데스트 하우스에 연백색의 황토 페인트를 칠할 예정입니다. <br>
그리고 이것저것 사소한 저이를 마치고 나면, 나이 드시고 젊고 한 미지의 새식구들을 설렘으로 만나고 함께 하고, 도시의 바쁜 생활로 소원해진 오래된 친구와의 우정을 이 제주에서 새롭게 열어나가기 위해 요안도라를 세상에 알리려고 합니다.</p>
<p>그럼 이 홈페이지를 공식적으로 체계적으로 갖추면서, 그동안 공사로 어수선했던 올려드리지 못한 집 안팎 사진들을 상세하게 올리고 당신을 만날 준비를 마치겠습니다.</p>
</article>

```

요안도라 소개
요안도라는 게스트 하우스(Guest House) 형식의 농어촌 민박입니다

성산의 날씨는 다음주 내내 높은 구름에 햇살 가득이라고 합니다. 목요일이면 섭씨 27도까지 오른다고 하지만, 늘 부는 바람이 쾌적한 균형을 잡아 마당에 마당에 나가 앉아 있는 시간이 많아질듯 합니다.

오늘은 사진에 보이는 긴 돌담을 따라 들어오는 요안도레 올레 입구에 특근색의 대문을 달았습니다.
내일은 두달 여동안 밖거리에 만든 데스트 하우스에 연백색의 황토 페인트를 칠할 예정입니다.
그리고 이것저것 사소한 저이를 마치고 나면, 나이 드시고 젊고 한 미지의 새식구들을 설렘으로 만나고 함께 하고, 도시의 바쁜 생활로 소원해진 오래된 친구와의 우정을 이 제주에서 새롭게 열어나가기 위해 요안도라를 세상에 알리려고 합니다.

그럼 이 홈페이지를 공식적으로 체계적으로 갖추면서, 그동안 공사로 어수선했던 올려드리지 못한 집 안팎 사진들을 상세하게 올리고 당신을 만날 준비를 마치겠습니다.

3) 그 외의 가상 클래스 선택자

- ::before ::after
- ::selection
- :not

```

input:not([type="text"]) {
width:150px; /* 너비 150px */
background-color:lime; /* 배경색 라임색 */
border:1px solid #000; /* 1px짜리 검은색 실선 */
}
<ul>
<li> <label for="name">예약자</label> <input type="text" id="name"> </li>
<li> <label for="contact">연락처</label> <input type="tel" id="contact"> </li>
<li> <label for="people">예약 인원</label> <input type="number" id="people"> </li>
<li> <label for="bank">입금 은행</label><input type="text" id="bank"> </li>
</ul>

```

예약자

연락처

예약 인원

입금 은행

□ CSS3와 브라우저 접두사(prefix)

- 1) CSS3의 표준 규약이 완성되지 않아 브라우저들이 새로운 CSS3 속성을 부분적으로만 구현할 수 있기 때문에 CSS3의 속성을 사용하려면 속성 이름 앞에 브라우저를 식별할 수 있는 접두사를 붙여야 한다.
- 2) 브라우저 접두사

접두사	설	명
-webkit-	웹킷 방식 브라우저	사파리, 크롬 등

-moz-	게코 방식 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷익스플로러

<pre> .myphoto{ position:absolute; left:50px; top:70px; transform: rotate(15deg); -webkit-transform:rotate(15deg); -moz-transform:rotate(15deg); } </pre>	
---	---

텍스트 관련 스타일

□ 글꼴 관련 스타일

1) font-family : 웹 문서에서 사용할 글꼴 지정

```
body { font-family: "맑은 고딕", "돋움", "굴림"; } /* 맑은 고딕 글꼴이 없으면 돋움, 그것도 없으면 굴림 */
```

2) @font-face : 웹 폰트 사용하기

```
@font-face{
  font-family:'myBand';
  src:url('Band.eot');
  src:url('Band.ttf') format('truetype');
  src:url('Band.woff') format('woff');
}
h1{
  font-family:myBand;
  font-size:3em;
  color:yellow;
}
```

3) font-size : 글자 크기 조절하기(기본 크기 : 12pt, 16px, 1em, 100%)

상대 크기(1.2배씩 커짐) : xx-small < x-small < small < medium(기본값) < large < x-large < xx-large

가) 크기 값

단 위	설 명
em	해당 글꼴의 대문자 M의 너비를 기준
ex	x-height, 해당 글꼴의 소문자 x의 높이를 기준
px	픽셀, 모니터에 따라 상대적인 크기 적용
pt	포인트, 보통 문서에서 많이 사용하는 단위

나) 브라우저 기본 글꼴 크기 비교

포인트(pt)	픽셀(px)	em	퍼센트(%)
10pt	13px	0.8em	80%
10.5pt	14px	0.875em	87.5%
11pt	15px	0.95em	95%
12pt	16px	1em	100%
13pt	17px	1.05em	105%

4) font-style : 이탤릭체 표시

속성 값	설 명
normal	일반적인 형태
italic	이탤릭체 표시
oblique	이탤릭체 표시

5) font-variant : 작은 대문자로 표시

속성 값	설 명
normal	일반적인 형태
small-caps	작은 대문자로 표시

6) font-weight : 글자 굵기 지정

속성 값	설 명
normal	일반적인 형태
bold	진하게 표시
lighter	원래 굵기보다 연하게 표시
bolder	원래 굵기보다 진하게 표시
100~900 사이의 수치	400은 보통, 700은 진하게를 기준으로 숫자 지정

7) font : 글꼴 속성을 한꺼번에 묶어서 표현

```
p { font:12pt/14pt 굴림 } /* 텍스트 크기 12pt/줄간격 14pt 글꼴 굴림 */
p { font:medium/110% "굴림 돋움" bold } /* 텍스트 크기 medium/줄간격 110% 글꼴 굴림 돋움, 진하게 */
```

□ 텍스트 스타일

1) direction : 텍스트의 쓰기 방향

```
h3{ direction:ltr; }
h4{ direction:rtl; }
```

2) text-align : 텍스트 정렬하기

```
#para1{ text-align:left;}
#para2{ text-align:right;}
#para3{ text-align:center;}
#para4{ text-align:justify;}
```

3) text-shadow : 텍스트에 그림자 효과 추가

속성 값	설 명
h-shadow	그림자의 가로 오프셋 거리, 양수는 오른쪽에, 음수는 왼쪽에 그림자 만들 (px 단위)
v-shadow	그림자의 세로 오프셋 거리, 양수는 아래에, 음수는 위쪽에 그림자 만들 (px 단위)
blur	그림자의 번지는 정도 지정, 생략가능 (px 단위)
color	그림자 색상, 16진수, rgb, 색상 이름 사용 가능

```
.shadow1 {
  text-shadow:1px 1px 3px #000; /* h-shadow v-shadow blur color 순 */
}
.shadow2 {
  color:black;
  text-shadow:3px -3px 5px #fff;
}
```

4) text-overflow : 넘치는 텍스트 표기

속성 값	설 명
------	-----

clip	넘치는 텍스트를 잘라버립니다.
ellipsis	텍스트가 잘렸다는 의미로 말줄임표("...")를 표시합니다.

```
.over1{
  text-overflow:clip;
}
.over2{
  text-overflow:ellipsis;
}
```

5) text-decoration : 텍스트에 줄 표시

속성 값	설 명
none	밑줄을 표시하지 않음
underline	밑줄 표시
overline	영역 위로 선을 표시
line-through	영역을 가로지르는 선을 표시

```
.line1{ text-decoration:underline; }
.line2{ text-decoration:overline;}
.line3{ text-decoration:line-through;}
.line4 { text-decoration:overline underline;}
```

6) text-indent : 텍스트 들여쓰기

```
#indnt1{ text-indent:10px;}
#indnt2{ text-indent:20px;}
#indnt3{ text-indent:30px;}
```

7) text-transform : 대,소문자 변환하기

속성 값	설 명
capitalize	시작하는 첫 번째 글자를 대문자로 변환
uppercase	모든 글자를 대문자로 변환
lowercase	모든 글자를 소문자로 변환
none	변환하지 않음

```
.tr1{ text-transform:none;}
.tr2{ text-transform:capitalize;}
.tr3{ text-transform:uppercase;}
.tr4{ text-transform:lowercase;}
```

□ 텍스트 간격을 조절하는 스타일

1) letter-spacing과 word-spacing : 텍스트 간격 조절하기

```
.heading1{
  font-family:myBand;
  font-size:5em;
  letter-spacing:0.1em;
}
.heading2{
  font-family:myBand;
  font-size:5em;
  letter-spacing:0.3em;
}
```

2) line-height : 줄간격 조절하기

```
p { font-size:10pt; line-height:1.2; }
p { font-size:10pt; line-height:120%; }
p { font-size:10pt; line-height:12pt; }

p { font:12pt/14pt 굴림 } /* 텍스트 크기 12pt/줄간격 14pt 글꼴 굴림 */
p { font:mediumt/110% "굴림 돌음" bold } /* 텍스트 크기 medium/줄간격 110% 글꼴 굴림 돌음, 진하게 */
```

3) white-space : 공백 처리

속성 값	설 명
normal	기본값으로 연속하는 공백을 하나로 처리
nowrap	연속하는 공백을 하나로 처리하고 줄바꿈을 하지 않음
pre	연속하는 공백을 원본 그대로 표시하고 줄바꿈도 함, 즉, 원본 그대로 표시
pre-line	연속하는 공백은 하나로 취급하고 줄바꿈은 함
pre-wrap	연속하는 공백을 원본 그대로 표시하고 줄바꿈도 함

□ 목록 스타일

1) list-style-type : 목록의 불릿과 번호 스타일 지정

가) 순서 없는 목록에서 사용하는 불릿 유형

속성 값	설 명
none	불릿 없음
disc (●)	속이 찬 원
circle (○)	속이 빈 원
square (■)	속이 찬 사각형

나) 순서 있는 목록에서 사용하는 불릿 유형

속성 값	설 명
decimal	
decimal-leading-zero	
lower-roman	
upper-roman	
lower-greek	
lower-alpha 또는 lower-latin	
upper-alpha 또는 upper-latin	
hebrew	
armenian	
georgian	
ckj-ideographic	
hiragana	
hiragana-iroha	
katakana	
katakana-iroha	

```
article ul li{
  list-style-type:square;
}
```

2) list-style-position : 목록에 들여쓰기 효과

속성 값	설 명
inside	불릿이나 숫자를 안으로 들여쓰
outside	기본값으로 불릿이나 숫자를 밖으로 내어쓰

```
.out{
  list-style-type:square;
  list-style-position:outside;
}
.in{
  list-style-type:square;
  list-style-position:inside;
}

<ul>
  <li class="out">1인: 40,000원 </li>
  <li class="out">2인: 60,000원</li>
  <li class="out">3인: 80,000원</li>
  <li class="in">4인: 100,000원</li>
  <li class="in">5인: 120,000원</li>
  <li class="in">6인: 140,000원</li>
</ul>
```

- 1인: 40,000원
- 2인: 60,000원
- 3인: 80,000원
- 4인: 100,000원
- 5인: 120,000원
- 6인: 140,000원

3) list-style-image : 불릿 대신 이미지 넣기

```
article ul li {
  font-size:14px;
  line-height:1.5;
  list-style-image:url(bullet.png);
}
```

4) list-style : list-style-type, list-style-position, list-style-image 속성을 한꺼번에 설정

```
article ul li {
  list-style:upper-alpha inside;
}
```

색상과 배경을 위한 스타일

□ 색상과 관련된 스타일

1) 색상 표기법

- 가) 16진수 표기법 : #RRGGBB #rgb
- 나) rgb 표기법 : rgb(r g b) // rgb(0~255)
- 다) hsl 표기법 : hsl(h s l) // h(색상: 0 ~ 355), s(채도: 0%~100%), l(명도: 0%~100%)
- 라) 색상 이름 표기법 : 기본 17색(aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow) + 130색
- 마) rgba 표기법 : rgba(r g b a) // a(0.0~1.0)
- 바) hsla 표기법 : hsla(h s l a) // a(0.0~1.0)

2) color : 글자색 지정

```
h1{ color:rgb(255,255,0); }
h2{ color:#00f; }
.op1{ color:rgb(255,255,255); }
.op2{ color:rgba(255,255,255,0.8); }
.op3{ color:rgba(255,255,255,0.5); }
.op4{ color:rgba(255,255,255,0.2); }
```

□ 배경 관련 스타일

1) background-color : 배경색 지정

```
.box1 { background:rgb(0,0,255);}
.box2 { background:rgba(0,0,255,0.5);}
.box3 { background:hsl(360,100%,50%);}
.box4 { background:hsla(360,100%,50%,0.5);}
```

2) background-image : 배경에 이미지 넣기

```
body{
  background-image:url(bg-img1.png);
}
```

3) background-repeat : 배경 이미지 반복하기

속성 값	설 명
repeat	브라우저 화면에 가득 찰 때까지 배경 이미지를 가로와 세로로 반복
repeat-x	브라우저 창 너비와 같아질 때까지 배경 이미지를 가로로 반복
repeat-y	브라우저 창 높이와 같아질 때까지 배경 이미지를 세로로 반복
no-repeat	배경 이미지를 한 번만 표시하고 반복하지 않음

```
body{
  background-color:#12007a;
  background-image:url(bg-img2.png);
  background-repeat:repeat-x;
}
```

4) background-position : 배경 이미지 위치 조절

- 가) 백분율(%) 표시 : 왼쪽 위(0% 0%), 중앙(50% 50%) 오른쪽 아래(100% 100%)
- 나) 길이(px) 표시 : 왼쪽 위(0px 0px)
- 다) 키워드 표시 : left, center, right / top, center, bottom

```
.bg{
```

```

background-image:url(bg2.png);
background-repeat:no-repeat;
background-position:0% 0%;
}

body {
background-image:url(bg3.png);
background-repeat:no-repeat;
background-attachment:fixed;
background-position: bottom right;
}

```

5) background-attachment : 배경 이미지 고정하기(fixed 또는 scroll:기본값)

```

body {
background-image:url(bg3.png);
background-repeat:no-repeat;
background-attachment:fixed;
background-position: bottom right;
}

```

6) background : background-image, background-repeat, background-position, background-attachment를 묶어서 한 번에 적용

```

header {
width:100%;
height: 280px;
margin-top: 0;
background: url(bg.png) no-repeat left top;
}

```

7) background-size : 배경 이미지 크기 조절, 값이 1개만 주어지면 width값이고, height=auto로 설정됨

속성 값	설	명
auto	원래 이미지 크기	
크기 값	너비와 높이 지정(높이는 생략 가능)	
백분율	배율 지정	
contain	너비나 높이 중 큰 값을 배경 이미지가 삽입되는 요소에 맞춤	
cover	너비나 높이 중 작은 값을 배경 이미지가 삽입되는 요소에 맞춤	

```

body{
background-size : auto auto;
background-size : 300px;
background-size : 200px 150px;
background-size : 45% auto;
background-size : contain;
background-size : cover;
}

```

8) background-clip : 배경 적용 범위 조절

속성 값	설	명
border-box	가장 외곽인 테두리까지 적용	
padding-box	테두리를 뺀 패딩 범위까지 적용	
content-box	내용 부분에만 적용	

```

body{
background-clip:border-box;
}

```

```
background-clip:padding-box;
background-clip:content-box;
}
```

9) background-origin : 배경 이미지 위치 기준 조절

속성 값	설 명
border-box	가장 외곽인 테두리 기준
padding-box	테두리를 뺀 패딩 기준
content-box	내용 영역 기준

```
body {
background-origin:border-box;
background-origin:padding-box;
background-origin:content-box;
}
```

□ 그라데이션 효과로 배경 꾸미기

1) linear-gradient : 선형 그라데이션

```
background:linear-gradient(135deg, blue, white); /*왼쪽 위에서 오른쪽 아래로, 파란색에서 흰색으로*/
background:linear-gradient(to bottom, blue, white); /*위쪽에서 아래쪽으로, 파란색에서 흰색으로*/
background:linear-gradient(to bottom, #06f, white 30%, #06f); /* 시작색 06f, 30%위치 흰색, 끝색 06f */
background:linear-gradient(150deg, blue, white, skyblue); /*파란색->흰색->하늘색 순으로 */
```

속성 값	설 명
to top	아래에서 시작해서 위쪽으로 그라데이션
to left	오른쪽에서 시작해서 왼쪽으로 그라데이션
to right	왼쪽에서 시작해서 오른쪽으로 그라데이션
to bottom	위에서 시작해서 아래쪽으로 그라데이션

2) radial-gradient : 원형 그라데이션

```
background:-prefix-radial-gradient(60% 60%, circle, white, blue);
/*중심(60%,60%)에서 원형으로 흰색에서 파란색으로 퍼지는 그라데이션*/
background:-prefix-radial-gradient(10% 10%, ellipse, white, blue);
/*중심(10%,10%)에서 타원형으로 흰색에서 파란색으로 퍼지는 그라데이션*/
background:-prefix-radial-gradient(30% 40%, circle closest-side, white, green);
/*background box의 경계선(수직 또는 수평선)을 만날 때까지만 그라데이션 적용*/
```

속성 값	설 명
closest-side	원이 가장 가까운 수직 또는 수평 경계선을 만날 때까지만 그라데이션 적용
closest-corner	원이 4개의 꼭지점 중 가장 가까운 점을 만날 때까지만 그라데이션 적용
farthest-side	원이 가장 멀리있는 수직 또는 수평 경계선을 만날 때까지만 그라데이션 적용
farthest-corner	원이 4개의 꼭지점 중 가장 멀리있는 점을 만날 때까지만 그라데이션 적용

레이아웃을 위한 스타일

□ 박스 모델

1) width, height : 박스 모델 크기 설정(크기, 백분유(%), auto)

```

p.size1 {
  width: 400px;
  height: 50px;
}
p.size2{
  width:50%;
  height:50px;
}
    
```

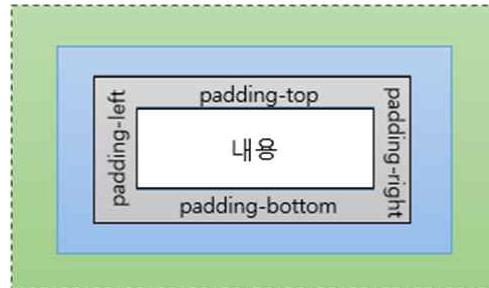
2) margin : 요소 주변 여백 설정(margin-top, margin-right, margin-bottom, margin-left 순으로 지정)

3) padding : 콘텐츠 영역과 테두리 사이 여백 설정

(padding-top, padding-right, padding-bottom, padding-left 순으로 지정)

```

.content1{
  border:1px solid black;
  background:lightyellow;
  padding:10px;          /* 네 방향의 패딩 모두 10px */
}
.content2{
  border:1px solid black;
  background:lightyellow;
  margin-bottom:10px;   /* 아래쪽 마진 10px */
  margin-left:50px;     /* 왼쪽 마진 50px */
  padding:10px 30px;   /* 위아래 패딩 10px, 좌우 패딩 30px */
}
.content3{
  border:1px solid black;
  background:lightyellow;
  margin:20px 10px 0 100px; /* 위쪽 마진 20px, 오른쪽 10px, 아래쪽 없음, 왼쪽 100px */
  padding:10px 30px 10px 50px; /* 위쪽 패딩 10px, 오른쪽 30px, 아래쪽 10px, 왼쪽 50px */
}
    
```



□ 테두리 관련 속성들

1) border-width : 테두리 두께 지정

(border-top-width, border-right-width, border-bottom-width, border-left-width 순으로 지정)

속성 값	설	명
thin		
medium		
thick		
크기		

--

2) border-color : 테두리 색상 지정

(border-top-color, border-right-color, border-bottom-color, border-left-color 순으로 지정)

속성 값	설	명
색상 값	색사 이름, 16진수 값, rgb 값 등 색상	
transparent	투명, 기본 값	

--

3) border-style : 테두리선 스타일 지정

(border-top-style, border-right-style, border-bottom-style, border-left-style 순으로 지정)

속성 값	설	명
none	테두리가 나타나지 않음	
hidden		
dotted		
dashed		
solid		
double		
groove		
ridge		
inset		
outset		

--

4) border : 테두리 스타일 묶어서 지정(두께 값, 색상 값, 스타일 순으로 지정)

(border-top, border-right, border-bottom, border-left를 따로 따로 지정 가능)

```
h3 {
  padding-bottom:5px;
  border-bottom:3px solid blue;
}

p {
  padding:10px;
  border:1px red solid;
}
```

□ 박스 관련 속성들

1) border-radius : 박스 모서리 둥글게 만들기

(top-left, top-right, bottom-right, bottom-left 순으로 지정)

```
.content1 {
  border-radius:10px;
  -moz-border-radius:10px;
}

.content2 {
```



```
border-radius:30px/10px; /* 4꼭지점 모두 가로 30px/세로 10px로 타원형으로 만들 */
-moz-border-radius:30px/10px;
}
```

2) border-*-radius : 박스 모서리 다르게 지정

```
.brd1 {
border-top-left-radius:100px 50px;
-moz-border-radius-topleft:100px 50px;
}
.brd2 {
border-bottom-right-radius:50% 30%;
-moz-border-radius-bottomright:50% 30%;
}
.brd3 {
border-top-right-radius:50px;
-moz-border-radius-topright:50px;
}
```

3) box-shadow : 그림자 효과 지정(수평거리 수직거리 blur 확장거리 색상 inset 순으로 지정)

```
#shd1 {
box-shadow: 10px 10px 5px 5px #666 ;
}
#shd2 {
box-shadow: 10px 10px 50px -1px white inset;
}
```

□ 배치와 관련된 블록 레벨과 인라인 레벨

종류	태그
블록 레벨	<p>, <h1>~<h6>, , , <div>, <blockquote>, <form>, <hr>, <table>, <fieldset>, <address>
인라인 레벨	, <object>, , <sub>, <sup>, , <input>, <textarea>, <label>, <button>

1) display : 요소를 어떻게 보일까 지정

속성 값	설명
block	
inline	
inline-block	
none	
inherit	
table	
inline-table	
table-row	
table-row-group	
table-header-group	
table-footer-group	
table-column	
table-column-group	

table-cell	
table-caption	
list-item	

```

div{
  display:inline; /* 인라인 레벨로 바꿈 */
  float:left; /* 왼쪽부터 차례로 배치 */
  width:200px;
  height:auto;
  margin:10px 5px;
  padding:5px;
  border:1px solid black;
}

img {
  display:block; /* 인라인 레벨인 img를 블록 레벨로 표시 */
  margin:10px 5px;
  border:2px solid black;
  border-radius:10px;
  box-shadow:5px 5px 3px #ccc;
}

```

□ 여러 요소를 원하는 형태로 배치하기

1) position : 원하는 형태로 요소 배치

속성 값	설 명
static	요소를 문서의 흐름에 맞춰 배치(다른 요소와 겹치지 않게 배치)
relative	이전 요소에 자연스럽게 연결하여 배치
absolute	원하는 위치를 지정하여 배치 (요소가 화면 밖으로 나가면 스크롤바 생김)
fixed	지정한 위치에 고정해서 배치 (요소가 화면 밖으로 나가면 스크롤바 없이 잘림)

```

#cont1 {
  position:static; /* 다른 요소와 겹치지 않게 배치 */
  background:lightyellow;
}
#cont2 {
  position:static; /* 다른 요소와 겹치지 않게 배치 */
  background:lightgreen;
}
#cont3 {
  position:static; /* 다른 요소와 겹치지 않게 배치 */
  background:lightblue;
}

```

입실은 오후 3시, 퇴실은 오전 11시입니다.
이곳은 관광지가 아닌, 관광사를 찾는 중간간 마을입니다.

반배류, 야외식사 등 바깥채, 외부 앞물에서의 활동은 밤 11시 이전까지 마쳐 주셔야 합니다. 바깥채 내부의 소등은 자유롭게 하시되, 자정 이후로 음주나 소란스러운 행위를 자제해 주시면 고맙겠습니다.

수도, 성신일출봉, 십지코지, 망갈갈리리, 용늪이 오후 2시부터 관광지와 한라산 동쪽 봉산으로 오가실 경우 요안도라에서 플랫폼을 이용하실 수 있습니다.

```

#cont1 {
  position:relative; /* 다른 요소와 자연스럽게 어울리도록 */
  background:lightyellow;
}
#cont2 {
  position:relative; /* 다른 요소와 자연스럽게 어울리도록 */
  left:20px; /* 원래 위치에서 오른쪽으로 20px 떨어져서 배치 */
  top:-20px; /* 원래 위치에서 위로 20px 떨어져서 배치 */
  background:lightgreen;
}
#cont3 {
  position:relative; /* 다른 요소와 자연스럽게 어울리도록 */
  left:-30px; /* 원래 위치에서 왼쪽으로 30px 떨어져서 배치 */
  top:20px; /* 원래 위치에서 아래로 20px 떨어져서 배치 */
  background:lightblue;
}

```

입실은 오후 3시, 퇴실은 오전 11시입니다.
이곳은 관광지가 아닌, 관광사를 찾는 중간간 마을입니다.

반배류, 야외식사 등 바깥채, 외부 앞물에서의 활동은 밤 11시 이전까지 마쳐 주셔야 합니다. 바깥채 내부의 소등은 자유롭게 하시되, 자정 이후로 음주나 소란스러운 행위를 자제해 주시면 고맙겠습니다.

수도, 성신일출봉, 십지코지, 망갈갈리리, 용늪이 오후 2시부터 관광지와 한라산 동쪽 봉산으로 오가실 경우 요안도라에서 플랫폼을 이용하실 수 있습니다.

```

#cont1 {
  position:absolute;
  left:20px;          /* 부모 요소(body) 왼쪽 위로부터 오른쪽으로 20px 떨어져서 배치 */
  top:50px;          /* 부모 요소(body) 왼쪽 위로부터 아래로 50px 떨어져서 배치 */
  background:lightyellow;
}
#cont2 {
  position:absolute;
  left:270px;        /* 부모 요소(body) 왼쪽 위로부터 오른쪽으로 27px 떨어져서 배치 */
  top:100px;         /* 부모 요소(body) 왼쪽 위로부터 아래로 100px 떨어져서 배치 */
  background:lightgreen;
}
#cont3 {
  position:absolute;
  left:520px;
  top:150px;
  background:lightblue;
}

```

입실은 오후 3시, 퇴실은 오전 11시입니다.
이곳은 관광지가 아닌, 골동품을 파는 중산간 마을입니다.

바베큐, 야외식사 등 바깥채, 외부 앞뜰에서의 활동은 밤 11시 이전까지 마쳐 주셔야 합니다. 바깥채 내부의 소동은 자주 흔하게 하시되, 자정 이후로 음주나 소란스러운 대화를 자제해 주시면 고맙겠습니다.

수도, 성산일출봉, 섬자코지, 일영강 걸리리, 용눈이 오름 등 주변 관광지와의 한라산 등해 일산도로 오가실 경우 유안도라에서 플랫폼을 이용하실 수 있습니다.

2) visibility : 요소를 보이게/안보이게

속성 값	설 명
visible	화면에 표시, 기본 값
hidden	자리는 차지하고 보이지 않음
collapse	표의 행/열 등에 지정하면 서로 겹치도록 조절
inherit	상속

```

.visible{
  visibility:visible;
}
.hidden{
  visibility:hidden;
}




```

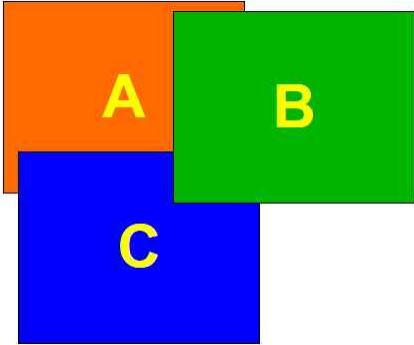


3) z-index : 요소 쌓는 순서 정하기

```

#cont1 {
  position:absolute;
  left:10px;
  top:10px;
  background:#ff6600;
  z-index:1;
}
#cont2 {
  position:absolute;
  left:180px;
  top:20px;
  background:#00b000;
  z-index:3;
}
#cont3 {
  position:absolute;
  left:25px;
  top:160px;
  background:#0000ff;
  z-index:2;
}

```



4) float : 요소를 떠 있게 하기 (position 속성 중 absolute 사용 시에는 동작 안 함)

한 태그에 float 속성이 설정되면 그 다음의 모든 태그에도 자동으로 적용 됨

```
.content {
  float: left; /* left와 right 중 하나 사용 */
  background-color: #FFFFFF;
  padding-top:5px;
  padding-left:10px;
  width: 60%;
}
```

5) clear : 앞 태그에 float가 설정되면 다음 태그에서 설정되는데, 이것을 해지하는 기능

속성 값	설 명
left	왼쪽으로 떠 있지 않도록 함
right	오른쪽으로 떠 있지 않도록 함
both	왼쪽이나 오른쪽 어디에도 떠 있지 않도록 함
none	왼쪽이나 오른쪽 모두에 뜸, 기본 값

```
aside{
  float: right; /* aside는 오른쪽에 뜨게 됨 */
  width: 30%;
  height:auto;
}
footer {
  clear: both; /* clear가 없으면 footer도 오른쪽에 뜨지만, clear:both;로 왼쪽/오른쪽 모두 안 뜸 */
  color:#fff;
  text-align:center;
}
```

□ 다단으로 편집하기

1) column-width : 단의 너비 지정(창의 너비가 넓어지면 단의 개수가 많아짐)(값 or auto)

```
div {
  column-width:80px;
  -webkit-column-width:80px;
  -moz-column-width:80px;
}
```

2) column-count : 단의 개수 지정(창의 너비가 넓어지면 단의 너비가 넓어짐)(개수 or auto)

```
div {
  column-count:3;
  -webkit-column-count:3;
  -moz-column-count:3;
}
```

3) column-gap : 단과 단 사이 여백(값 or normal)

```
div {
  column-gap:3px; /* 또는 auto */
}
```

4) column-rule : 구분선의 두께, 스타일, 색상 순으로 지정

(column-rule-width, column-rule-style, column-rule-color로 각각 나누어 지정해도 됨)

```
.multicol {
  columns: auto 3;
  column-rule:1px dotted black;
  -webkit-columns: auto 3;
  -webkit-column-rule:1px dotted black;
```

```

-moz-columns: auto 3;
-moz-column-rule:1px dotted black;
}

```

5) break-before, break-after, break-inside

```

.multicol article {
  break-after: column; /* .multicol article 태그 아래 모든 것을 다단 설정, avoid-column은 다단 해제*/
  -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;
}

```

6) column-span : 여러 단을 하나로 합치기

```

.subtitle {
  -webkit-column-span:all;
  -moz-column-span:all;
  column-span:all;
  background:skyblue;
  padding:5px;
  border:2px double blue;
}

```

□ 표 관련 스타일

1) table-layout : 콘텐츠에 맞게 셀 너비 지정

속성 값	설	명
fixed	셀 너비를 고정	
auto	셀 안의 내용에 따라 셀의 너비가 달라짐, 기본 값	
inherit	상속	

```

#tb1 {
  width:300px;
  table-layout:fixed;
  word-break:break-all;
  height:auto;
}
th, td {
  border:1px solid black;
  padding:5px;
}

```

2) text-align : 수평 정렬

속성 값	설	명
left		
right		
center		
justify	양쪽 끝에 맞춰 정렬	

```


```

3) vertical-align : 수직 정렬

속성 값	설	명
baseline		
top		
bottom		
middle		
sub, super, text-top, text-bottom, 길이값, 백분율	이 값들은 셀에 적용되지 않음	

4) border-style : 표 테두리 스타일 지정 (앞쪽에 설명 있음)

--

5) border-collapse : 테두리 통합/분리

속성 값	설	명
collapse	테두리가 통합	
separate	테두리가 분리 됨, 기본 값	
inherit	상속	

6) border-spacing : 인접한 셀 테두리 사이 거리 지정(크기 or inherit)

```
table{
  border-collapse:seperate;
  border:outset 10px;
  border-spacing:15px;
  margin-bottom:20px;
}
th, td{
  border:inset 5px;
}
```

7) empty-cells : 빈 셀의 표시 여부 지정

속성 값	설	명
show	빈 셀 주위에 테두리를 그려 빈 셀 표시, 기본 값	
hide	빈 셀 주위에 테두리를 그지 않음	
inherit	상속	

8) caption-side : 캡션 위치

속성 값	설	명
top	위쪽	
bottom	아래쪽	



변형을 위한 스타일

□ transform : 웹 요소 변형

속성 값	설	명
translate(tx, ty)		
translate3d(tx, ty, tz)		
translateX(tx)		
translateY(ty)		
translateZ(tz)		
scale(sx, sy)		
scale3d(sx, sy, sz)		
scaleX(sx)		
scaleY(sy)		
scaleZ(sz)		
rotate(rx, ry, 각도)		
rotate3d(rx, ry, rz, 각도)		
rotateX(rx, 각도)		
rotateY(ry, 각도)		
rotateZ(rz, 각도)		
skew(x각도, y각도)		
skewX(x각도)		
skewY(y각도)		

```

.translateX img{
  transform:translateX(-10px);
  -webkit-transform:translateX(-10px);
  -moz-transform:translateX(-10px);
}
.translateXYZ img{
  transform:translate3d(10px, 20px, 10px);
  -webkit-transform:translate3d(10px, 20px, 10px);
  -moz-transform:translate3d(10px, 20px, 10px);
}
.scaleZ img{
  transform:scaleZ(0.7);
  -webkit-transform:scaleZ(0.7);
  -moz-transform:scaleZ(0.7);
}
.scaleXYZ img{
  transform:scale3d(1.2, 1.5, 0.7);
  -webkit-transform:scale3d(1.2, 1.5, 0.7);
  -moz-transform:scale3d(1.2, 1.5, 0.7);
}
.rotateZ img{
  transform:rotateZ(45deg);
  -webkit-transform:rotateZ(45deg);
  -moz-transform:rotateZ(45deg);
}
.rotateXYZ img{
  transform:rotate3d(2.5, 1.2, -1.5, 45deg);
  -webkit-transform:rotate3d(2.5, 1.2, -1.5, 45deg);
}

```



```

    -moz-transform:rotate3d(2.5, 1.2, -1.5, 45deg);
}

```

1) transform-origin 속성 : 변형 원점 설정하기(회전에서 많이 사용됨)

속성 값	설	명
x축	좌푼값, 백분율, left, center, right	
y축	좌푼값, 백분율, top, center, bottom	
z축	좌푼값만 사용 가능	

```

.ex img{
  transform:rotateZ(10deg);
  -webkit-transform:rotateZ(10deg);
  -moz-transform:rotateZ(10deg);
}
.ltop img{
  transform-origin:left top;
  -webkit-transform-origin:left top;
  -moz-transform-origin:left top;
}
.rtop img{
  transform-origin:right top;
  -webkit-transform-origin:right top;
  -moz-transform-origin:right top;
}
.lbt img{
  transform-origin:left bottom;
  -webkit-transform-origin:left bottom;
  -moz-transform-origin:left bottom;
}
.rbt img{
  transform-origin:right bottom;
  -webkit-transform-origin:right bottom;
  -moz-transform-origin:right bottom;
}

```



2) perspective : 원근감 표현

```

div#pers{
  perspective:200px;
  -webkit-perspective:200px;
}

```

3) perspective-origin : 원근감 조절

속성 값	설	명
x축	좌푼값, 백분율, left, center, right, 기본값은 50%	
y축	좌푼값, 백분율, top, center, bottom, 기본값은 50%	

```

div#pers{
  perspective-origin:right bottom;
  perspective:200px;
  -webkit-perspective:200px;
}

```

4) transform-style : 3D 변형 적용하기

속성 값	설	명
flat	하위 요소를 평면으로 처리	

preserve-3d	하위 요소들에 3D 효과를 적용

5) backface-visibility : 요소의 뒷면 표시여부 (visible 또는 hidden)

```
backface-visibility : hidden;
```

□ 시간에 따른 변화를 만들어주는 트랜지션

1) transition-property : 적용할 트랜지션 지정

2) transition-duration : 트랜지션 진행 시간

속성 값	설	명
none	트랜지션 동안 아무 속성도 바뀌지 않음	
all	트랜지션 동안 아무 모든 속성이 대상, 기본 값	
속성 이름	트랜지션 효과를 지정할 속성, 여럿일 때는 콤마(,)로 구분	

3) transition-timing-function : 트랜지션 형태 선택하기

속성 값	설	명
linear	시작에서 끝까지 같은 속도로 진행	
ease	처음에는 천천히, 점점 빨라지다가, 마지막엔 천천히, 기본값	
ease-in	시작을 느리게	
ease-out	느리게 끝냄	
ease-in-out	느리게 시작하고 느리게 끝남	
cubic-bezier(n,n,n,n)	직접 베지에 함수를 정의해서 사용(0~1 사이값 사용)	

4) transition-delay : 트랜지션 지연 시간 설정

```
#ex div{
  width:100px;
  height:100px;
  background-color:red;
  border-radius:0px;
  transition-property:background-color, border-radius; /* 적용 속성 : all 또는 none */
  -webkit-transition-property:background-color, border-radius;
  -moz-transition-property:background-color, border-radius;
  transition-duration:1s; /* 진행 시간 1초 */
  -webkit-transition-duration:1s;
  -moz-transition-duration:1s;
  transition-delay:2s; /* 지연 시간 2초 */
  -webkit-transition-delay:2s;
  -moz-transition-delay:2s;
}
#ex:hover div{
  background-color:blue;
  border-radius:50px;
}
```

5) transition : 트랜지션 속성 한꺼번에 표기(property duration timing-function delay 순)

```

#ex div{
width:100px;
height:100px;
border-radius:0px;
border:2px solid black;
background:url(f1.png) no-repeat center center padding-box;
transition:all 1s ease-in 0.3s;
-moz-transition:all 1s ease-in 0.3s;
-webkit-transition:all 1s ease-in 0.3s;
}
#ex:hover div{
border-radius:50px;
border:2px solid blue;
background:url(f2.png) no-repeat center center padding-box;
}

```

□ 애니메이션 : @keyframes 속성을 이용해서 애니메이션 정의

- 1) animation-duration : 애니메이션 실행 시간
- 2) animation-name : @keyframes를 이용해 여러 개의 애니메이션을 정의한 후 그 중 사용할 애니메이션 이름
- 3) animation-iteration-count : 반복 횟수 지정(숫자 또는 infinite)
- 4) animation-direction : 애니메이션 방향 지정(normal, alternate)
- 5) animation : 한꺼번에 표기(name duration timing-function delay iteration-count direction 순)

```

#myball {
position:relative;
width:100px;
height:100px;
border-radius:50px;
border:2px solid black;
background:url(f1.png) no-repeat center center padding-box;
animation-duration:3s;
animation-name:myani;
-webkit-animation-duration:3s;
-webkit-animation-name:myani;
-moz-animation-duration:3s;
-moz-animation-name:myani;
}

@keyframes myani {
0%{ left: 10px;}
100% { left:500px; }
}

@-webkit-keyframes myani {
0% {left: 10px;}
100% { left:500px; }
}

@-moz-keyframes myani {
0% {left: 10px;}
100% { left:500px; }
}

@keyframes myani2 {
0%{ left: 10px;}
40% { top:50px; background:url(f2.png) no-repeat center center; }
100% { left:500px; }
}

@-webkit-keyframes myani2 {
0% {left: 10px;}
40% { top:50px; background:url(f2.png) no-repeat center center; }
100% { left:500px; }
}

@-moz-keyframes myani2 {
0% {left: 10px;}

```

```
40% { top:50px; background:url(f2.png) no-repeat center center; }
100% { left:500px; }
}
```

반응형 웹사이트 만들기

□ 반응형 웹디자인 : 화면 크기에 반응하여 화면 요소들을 자동적으로 바꾸어 사이트를 구현하는 것

□ 미디어 쿼리 : CSS3 모듈 중 하나, 사이트에 접속하는 장치에 따라 특정 CSS스타일을 사용하도록 해 주는 기능
: 대소문자 구별 안함

1) 미디어 유형의 종류

속성 값	설 명
all	모든 미디어 유형에서 사용할 CSS 정의
print	인쇄 장치에서 사용할 CSS 정의
screen	컴퓨터 스크린에서 사용할 CSS 정의, 스마트폰의 스크린도 포함
tv	음성과 영상이 동시 출력되는 TV에서 사용할 CSS 정의
aural	음성 합성 장치(주로 화면을 읽어 소리로 출력해 주는 장치)에서 사용할 CSS 정의
braille	점자 표시 장치에서 사용할 CSS 정의
handheld	패드(pad)처럼 손에 들고 다니는 장치를 위한 CSS 정의
projection	프로젝터를 위한 CSS 정의
tty	디스플레이 기능이 제한된 장치에 맞는 CSS 정의(픽셀(px) 단위를 사용할 수 없음)
embossed	점자 프린터를 위한 CSS 정의

2) 미디어 쿼리에서 사용하는 조건

속성 값	설 명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이
device-width, device-height	단말기의 가로 너비, 세로 높이
min-device-width, min-device-height	단말기의 최소 너비, 최소 높이
max-device-width, max-device-height	단말기의 최대 너비, 최대 높이
orientation:portrait	단말기의 세로 방향
orientation:landscape	단말기의 가로 방향
aspect-ratio	화면 비율(width/height)
min-aspect-ratio	최소 화면 비율
max-aspect-ratio	최대 화면 비율
device-aspect-ratio	단말기 화면 비율(단말기 너비/단말기 높이)
min-device-aspect-ratio	단말기 최소 화면 비율
max-device-aspect-ratio	단말기 최대 화면 비율
color	비트 수
min-color	최소 비트수
max-color	최대 비트수

□ 미디어 쿼리 적용하기

1) CSS 파일 연결하기

```
<link href="css/print.css" rel="stylesheet" type="text/css" media="print">
/* 프린트를 사용할 경우 외부 스타일시트 파일 print.css를 불러와서 사용하도록 링크*/
<link href="css/tablet.css" rel="stylesheet" type="text/css" media="screen and (min-width:321px)">
/* 321px 이상의 화면에서는 tablet.css를 링크해서 사용하도록 지정*/
```

2) 웹 문서에서 직접 정의하기

```
body{
    background-color:gray;
}

@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;}
}
```

3) 모바일 해상도

종류	기기	CSS 크기
스마트폰	아이폰 5	320x568
	아이폰 6	375x667
	아이폰 6 Plus	414x736
	LG 옵티머스 G	384x640
	LG G3	360x640
	갤럭시 S3, S4, S5 / 갤럭시 노트2, 노트3	360x640
태블릿	아이패드 mini 2.3 / 아이패드 3, 4, air	768x1024
	갤럭시 탭(8.9인치)/ 갤럭시 탭 2.3(10인치)	800x1280
	MS 서피스 프로	720x1280
	MS 서피스	768x1366
	킨들 파이어 HD7	604x966
	킨들 파이어 HD 8.9	600x960

4) 뷰포트 지정 : 웹킷(webkit) 기반인 모바일 브라우저들은 기본 뷰포트 너비가 980px이다.

웹페이지를 320px로 맞춰 제작하더라도 기본 뷰포트의 너비가 980px이기 때문에 웹 페이지의 너비를 무조건 980px로 표시하려고 하여, 의도한 것과 달리 작은 글씨와 작은 그림으로 표시됨

```
<meta name="viewport" content="width=device-width"> /*뷰포트의 너비를 스마트폰의 화면 너비로 지정*/
```

속성 값	설	명
width	뷰포트의 너비 지정, 기본 값은 device-width	
height	뷰포트의 높이 지정, 기본 값은 device-height	
initial-scale	초기 배율, 기본 값은 1, 1보다 작은 값은 축소, 큰 값은 확대	

user-scalable	사용자가 페이지를 확대/축소할 수 있는지의 여부 지정, 디폴트 yes, 반대 no
minimum-scale	사용자가 축소할 수 있는 최소값, 기본 값은 0.25, 가로 값을 기준으로 함
maximum-scale	사용자가 확대할 수 있는 최대값. 기본 값은 5.0

<pre><meta name="viewport" content="width=device-width, user-scalable=yes, initial-scale=1.0, maximum-scale=3.0></pre>
--

□ 유동형 레이아웃

- 1) 처음부터 유동형 레이아웃으로 만들기보다는
픽셀(px)값을 사용하는 고정형을 사용하여 만든 뒤 백분율(%)로 변경하면 됨
- 2) 전체를 감싸는 div를 추가
가) 웹 문서 내용 전체의 크기나 배경색 등을 한꺼번에 조절할 수 있음
나) 브라우저 화면 크기에 상관없이 웹 문서의 내용을 중앙에 배치할 수 있음
- 3) 요소의 너비를 백분율(%)로 변경
- 4) 패딩과 마진을 백분율(%)로 변경
- 5) 글자 크기를 em 단위로 변환

<pre><!doctype html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width"> <title>너비가 고정된 레이아웃</title> <style> body{ background:#ccc; } #wrapper{ width:96%; margin:0 auto; } header{ width:100%; height:120px; } header h1{ padding-top:3.1%; padding-left:12.5%; } .header1{ font-size:2.5em; } article { width:62.5%; padding:2.1%; } aside { width:31.25%; padding:0.08%; } footer{ background:#677b75;</pre>	<pre>padding:1.0%; } </style> </head> <body> <div id="wrapper"> <header> <h1 class="header1">유동형</h1> </header> <section> <article> <p>실제 내용이 표시됩니다</p> <p>실제 내용이 표시됩니다</p> <p>실제 내용이 표시됩니다</p> <p>실제 내용이 표시됩니다</p> <p>실제 내용이 표시됩니다</p> </article> <aside> <p>사이드바의 내용이 표시됨</p> <p>사이드바의 내용이 표시됨</p> <p>사이드바의 내용이 표시됨</p> <p>사이드바의 내용이 표시됨</p> <p>사이드바의 내용이 표시됨</p> </aside> </section> <footer> <address>제작자 연락처 정보</address> <p> 저작권 정보 및 주소 </p> </footer> </div> </body> </html></pre>
--	--