

9강. CSS3 [6] : 고급 기능

1. 단단 레이아웃

1.1 단단 문서의 개수 및 폭 지정

- column-count 속성은 단단을 구성할 때 단의 개수를 지정하는 속성이다.
- column-width 속성은 단의 폭을 지정하는 속성이다.
- column-width와 column-count 속성은 동시에 auto 속성으로 지정할 수 없다.
- columns 속성은 단의 개수와 폭을 한번에 지정하기 속성이다.
- 단단 관련 속성의 사용은 벤더 프리픽스가 필요하다.

<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-count 속성은 단을 구성할 때 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-count 속성은 단을 구성할 때 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-count 속성은 단을 구성할 때 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-count 속성은 단을 구성할 때 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다. 이 속성은 단의 개수를 지정하는 속성이다.</p>
<code>div { column-count : 1 }</code>	<code>div { column-count : 2 }</code>	<code>div { column-count : 3 }</code>	<code>div { column-count : 5 }</code>

1.2 열의 간격 지정

- column-gap 속성은 단과 단 사이의 간격 지정하는 것으로 단 사이에 구분선이 있다면 구분선도 이 여백에 포함된다.

■ 속성값 → <length> | normal

<p>Cascading Style Sheet Level 3</p> <p>이러한 각종 디바이스들에 대응하여 다양한 크기의 화면에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-gap 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>이러한 각종 디바이스들에 대응하여 다양한 크기의 화면에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-gap 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>이러한 각종 디바이스들에 대응하여 다양한 크기의 화면에서 HTML5로 업그레이드된 CSS3의 새로운 속성 중 하나인 column-gap 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다. 이 속성은 단과 단 사이의 간격을 지정하는 속성이다.</p>
<code>div { column-count: 3; column-gap:normal}</code>	<code>div { column-count: 3; column-gap:100px}</code>	<code>div { column-count: 3; column-gap:50px}</code>

1.3 단 구분선 스타일 지정

- column-rule-color 속성은 단과 단 사이의 색상을 지정하는 속성으로 <color> 속성 값은 color 속성과 동일하다.
- column-rule-style 속성은 단 사이의 구분선의 형태를 지정하는 속성으로 border-style 속성에서 사용할 수 있는 값들과 동일하게 값을 지정할 수 있다.
- column-rule-width 속성은 단 사이의 구분선 굵기를 지정한다. border-style 속성과 동일하게 thin, medium, thick 길이 단위로 속성 값을 지정할 수 있지만 음수 값을 지정할 수 없다.



- column-rule 속성은 단 사이의 구분선을 표현하기 위한 속성으로 column-rule-color, column-rule-style, column-rule-width 속성을 한번에 지정할 수 있다.

<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>
--	--	--

```
div { column-count: 3;
column-rule : 3px dotted blue }
```

```
div { column-count: 3;
column-rule : 5px dashed red }
```

1.4 단의 확장

- column-span 속성은 단안을 구성할 때 해당 요소가 얼마나 많은 단을 차지하는 지를 지정하는 속성이다.
- 속성값 → none | all

<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>
--	--

column-span: none

column-span: all

1.5 단의 확장

- column-fill 속성은 단안 영역의 콘텐츠 흐름의 영향을 받는 방식을 지정하는 속성으로, 단에 있는 콘텐츠의 균형을 조절하지(단 길이의 차이를 최소화하지) 여부를 지정할 수 있다.
- 속성값 → auto | balance (기본)

<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>	<p>Cascading Style Sheet Level 3</p> <p>HTML5의 HTML4.01에서 HTML5로 업그레이드가 되는 동안 CSS2.1과 CSS3에서 CSS3로 업그레이드가 되면서 다양한 표현을 추가하였다. 요즘은 모바일 기기 및 태블릿 PC용의 다양한 디바이스들이 출시되고 있고 관련 산업이 발전하면서 이러한 디바이스들에 대응이 용이하고 있는 상태이다.</p>
--	--

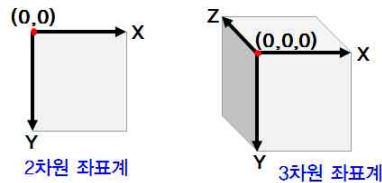
```
#myDIV { -moz-column-fill: auto ; }
```

```
#myDIV { -moz-column-fill : balance ; }
```

2. 변형transform

2.1 변형

- 요소의 위치를 옮기거나 크기를 변경하고 회전 및 왜곡시키는 것이다. 변형 관련 속성들은 2차원뿐만 아니라 3차원 변형도 가능하다.
- 다음 그림은 2차원 좌표계와 3차원 좌표계를 나타낸 것으로 2차원 좌표계는 수평의 X축과 수직의 Y축으로 이루어져 있으며, 3차원 좌표계는 2차원 좌표계에 Z축을 더해서 평면의 깊이나 원근감을 나타내도록 하고 있다.

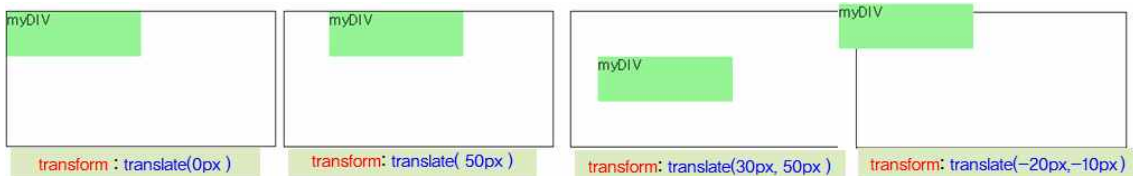


- transform 속성은 요소 박스의 변형을 지정하는 속성으로, 변형하고자 하는 형태별로 함수타입의 속성 값을 이용한다. 이때 함수의 속성 값은 공백으로 구분하여 여러 개의 속성 값을 지정할 수 있다.

■ 속성값 → none | <transform-function>+

2.1.1 translate() 변형 함수

- 요소 박스를 지정한 거리만큼 수평/수직/앞뒤로 이동시키는 함수로써, 음수 값을 지정하면 반대 방향으로 이동된다. 그리고 이동된 요소는 다른 요소의 레이아웃에는 영향을 주지 않는다.



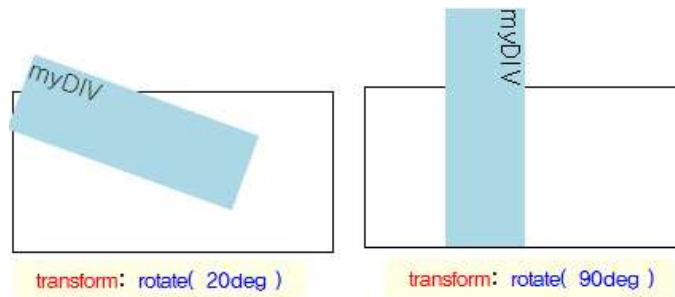
2.1.2 scale() 변형 함수

- 요소 박스를 지정한 크기 만큼 확대 또는 축소시키는 함수로써, 요소 본래의 크기에 지정한 배율만큼의 크기로 적용된다.



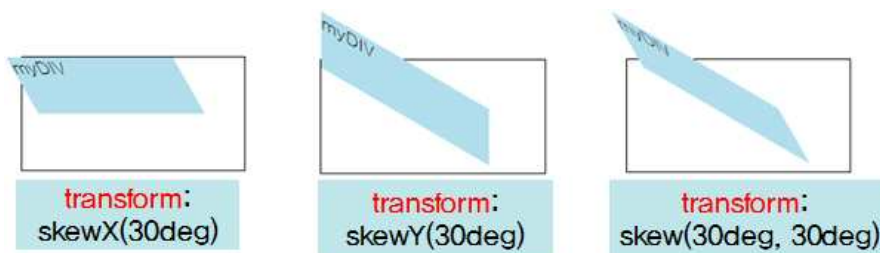
2.1.3 rotate() 변형 함수

- 요소 박스를 지정한 각도만큼 회전시키는 함수로써, 지정한 각도가 양수 값이면 시계 방향으로, 음수 값이면 시계 반대 방향으로 회전한다.



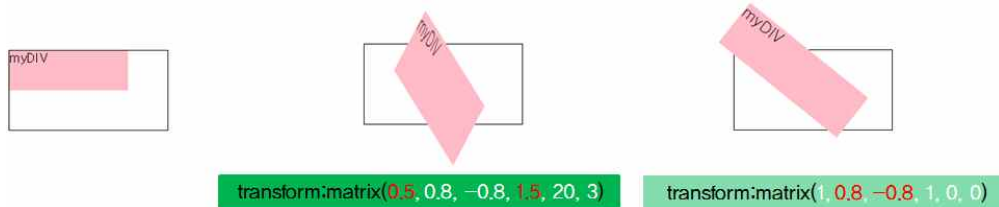
2.1.4 skew() 변환 함수

- 요소 박스를 지정한 각도만큼 기울이는 함수이다.



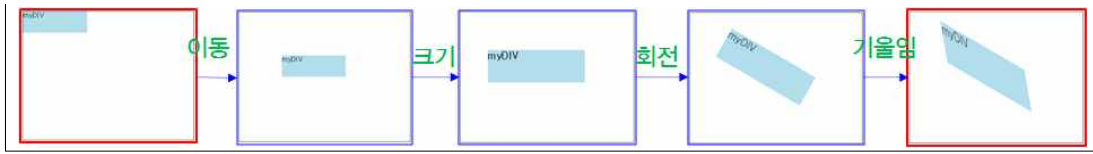
2.1.5 matrix() 변환 함수

- matrix(scaleX, skewX, skewY, scaleY, translateX, translateY)
- 요소 박스의 이동, 크기 변환, 기울임의 2D 변환 등을 일괄 지정하는 함수이다.



```

예제 transform 속성
<!DOCTYPE html> <html> <head>
  <style>
    .block { width: 400px; height: 300px; margin: 70px; border:1px solid black }
    #myDIV { width:150px; height: 50px; background-color: lightblue;
              transform: translate(100px, 100px)
                          scale(1.5, 1.5)
                          rotate(30deg)
                          skew(40deg) ;}
  </style>
</head>
<body> <div class="block">   <div id="myDIV">myDIV</div> </div> </body>
</html>
    
```



2.2 transform-origin 속성

- 요소 박스의 변형 기준점을 지정하는 속성으로 transform 속성과 함께 사용한다. 따라서 변형 함수라도 변형의 기준점이 달라지면 결과가 다르게 표시되기 때문에 변형 관련 여러 속성을 함께 사용하면 원근감을 표현하거나 요소의 뒷면을 표시할 수 있다.

■ 각 축에서 사용할 수 있는 속성값

- X축 → <length> | <%> | left | center (기본) | right
- Y축 → <length> | <%> | top | center (기본) | bottom
- Z축 → <length>

예제

```
<!DOCTYPE html> <html> <head>
  <style>
    .block { width: 200px; height: 100px; margin: 100px; border:1px solid black }
    #myDIV { width:150px; height: 50px; background-color: coral;
             transform: rotate(30deg);
             transform-origin: left bottom }
  </style>
</head>
<body> <div class="block">   <div id="myDIV">myDIV</div> </div> </body>
</html>
```



transform-origin: left top

transform-origin: left bottom

2.3 transform-style 속성

- 3D 공간에서 변형이 적용된 좌표를 하위 요소에게 전달할 것인지를 결정하는 속성이다.
- 속성값 → flat (기본) | preserve-3d

2.4 perspective 속성

- 3차원 공간에서 해당 요소와 관측 점과의 거리를 조절해서(사용자가 있는 방향으로 잡아당기거나 반대 방향으로 밀어내서) 원근감을 지정하는 속성이다.

▪ 속성값 → none | <length>

2.5 perspective-origin 속성

- 3차원 공간에서 원근감의 방향을 지정하는 속성으로 perspective 속성과 함께 사용해야 한다.

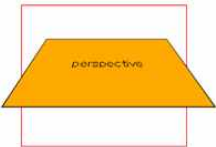
▪ 속성값 → <length> | <%> | left | center | right | top | bottom

예제

```

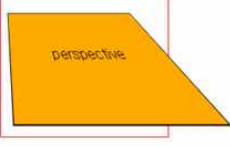
<!DOCTYPE html> <html> <head>
<style>
#div1 {
  position: relative; height: 150px; width: 180px; margin: 50px;
  padding: 10px; border: 1px solid red;
  perspective: 150px;
  perspective-origin: 100% 50%;
}
#div2 {
  padding: 50px; position: absolute; border: 1px solid black;
  background-color: orange; transform: rotateX(45deg);
}
</style>
</head>
<body> <div id="div1"> <div id="div2">perspective</div></div> </body>
</html>

```

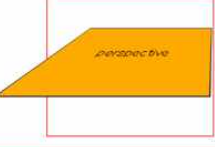


perspective-origin: 50% 50%

 (기본)



perspective-origin: left top



perspective-origin: 100% 50%

2.6 backface-visibility 속성

- 요소의 뒷면(반대쪽 면)을 표시 할 것인지를 결정하는 속성으로 요소 박스가 뒤집혔을 경우에 박스 뒷면의 표시 여부를 지정할 수 있다. 따라서 요소를 회전시키는 애니메이션

선에서 요소의 뒷면을 보이지 않도록 할 때 유용하다.

- 속성값 → visible (기본) | hidden

3. 트랜지션transition

3.1 트랜지션

- 트랜지션(Transition)이란 어떠한 상태가 다른 상태에 영향을 주는 것을 의미하는 것으로, CSS3에서의 트랜지션은 특정 스타일에서 다른 스타일로 바뀌는 것을 말한다. 즉, 요소 박스에 지정된 속성이 점진적으로 바뀌기 때문에 이러한 속성의 변화 과정을 통해서 애니메이션 효과를 줄 수 있으며, 사용자는 동적 콘텐츠의 느낌을 받는다.
- 트랜지션 지정 방법
 1. 어떠한 이벤트에 스타일의 변화를 줄 것인지 지정
 2. 변화를 주고자 하는 속성의 처음 상태와 최종 상태를 지정
 3. transition 속성을 사용하여 움직임 속도나 지연 시간 등을 지정

3.2 transition-property 속성, transition-duration 속성

- transition-property 속성은 요소에 지정된 속성을 변환(transition)하고자 할 때 사용하는 속성이다.
- transition-duration 속성은 트랜지션이 진행되는 시간을 지정하는 속성이다. 트랜지션 대상 속성을 지정했다면, 이 속성을 이용하여 진행 시간을 지정해야 속성이 부드럽게 바뀌는 애니메이션의 효과를 만들 수 있다. 트랜지션 대상이 되는 속성이 여러 개라면 진행 시간도 콤마(,)로 구분하여 여러 개 지정할 수 있다.

3.3 transition-timing-function 속성

- 트랜지션이 진행되는 속도의 형태를 지정하는 속성이다. 즉, 트랜지션이 일어나는 시작과 중간, 끝부분에서의 속도 곡선을 선택해야 한다.

3.4 transition-delay 속성, transition 속성

- transition-delay 속성은 트랜지션이 시작되기 전에 지연되는 시간을 지정하는 속성으로, 지정한 시간만큼 기다렸다가 트랜지션이 시작된다. 지연 시간은 초 단위로 지정하면 된다.
- transition 속성은 지금까지 살펴본 트랜지션 관련 속성들을 한번에 지정할 수 있는 대표 속성이다. 각 속성별로 트랜지션을 지정하려면 공백으로 구분하여 지정해야 한다.

예제 트랜지션

```

<!DOCTYPE html> <html> <head>
  <style>
    div {width: 150px; height: 100px;
      position: relative; left: 10px;
      background: orange;
      transition-property: width, left, color;
      transition-duration: 2s, 2s, 2s;
      transition-timing-function: ease-in, linear, ease-in-out;
      transition-delay: 0s, 1s, 1s}
    div:active { width: 300px; left: 200px; background: red; }
  </style>
</head>
<body>   <div> transition 속성 </div> </body>
</html>
    
```

3.5 사용 가능한 속성

트랜지션을 모든 속성과 속성 값에 적용할 수 있는 것은 아님!

background-color	border-top-width	margin-right	padding-left
background-position	bottom	margin-top	padding-right
border-bottom-color	clip	max-height	padding-top
border-bottom-width	color	max-width	right
border-left-color	font-size	min-height	text-indent
border-left-width	font-weight	min-width	text-shadow
border-right-color	height	opacity	top
border-right-width	left	outline-color	vertical-align
border-spacing	letter-spacing	outline-width	visibility
border-top-color	line-height	padding-bottom	width

4. 마키marquee

4.1 마키

- 시간의 흐름에 따라서 박스의 콘텐츠를 지정한 특성에 맞게 이동시키도록 정의한 것이

다.

- 어떤 이벤트의 도움 없이도 콘텐츠의 움직임을 지정할 수 있다.
- overflow-style 속성은 CSS3에서 마키기능의 스크롤 방법을 지정할 수 있도록 새로운 속성 값이 추가되었다.

▪ overflow-style 속성 값 → auto | marquee-line | marquee-block

4.2 마키 스타일 지정

- marquee-style 속성은 요소의 모든 내용을 임의의 박스로 생성한 후에 해당 요소 박스의 바깥으로부터 스크롤링하여 움직이도록 지정하는 속성이다.
- marquee-play-count 속성은 요소 내용의 움직임 횟수를 지정하는 속성이다.
- marquee-direction 속성은 움직이는 방향을 지정하는 속성이다.
- marquee-speed 속성은 요소 내용의 움직임 속도를 지정할 수 있는 속성이다.



5. 애니메이션animation

5.1 애니메이션

- 애니메이션 관련 속성은 키프레임 규칙을 사용하여 특정 시점에 애니메이션을 정의할 수 있다. 재생 횟수, 진행 방향, 일시 정지 등의 기능을 지원하기 때문에 보다 능동적으로 동적 효과를 구현 할 수 있다.
- 키 프레임(keyframe)은 애니메이션을 구현할 때 지정된 애니메이션의 동작이 시작되는 특정 시점으로, 애니메이션 중간 중간에 요소(객체)의 위치와 크기 등이 자동으로 구현 된다.

```
@keyframes 애니메이션이름 {
  from { 애니메이션의 시작 상태 }
  [ percentage { 해당 시점에서의 상태 } ]
  ...
  to { 애니메이션의 종료 상태 }
}
```

5.2 애니메이션 관련 속성

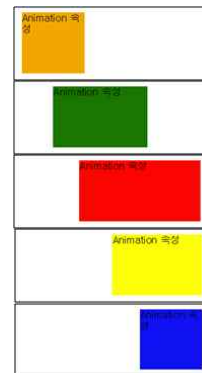
animation-name	키 프레임의 이름을 지정하여 해당 애니메이션을 실행
animation-duration	애니메이션이 진행되는 시간 (초 단위)
animation-timing-function	애니메이션 진행 속도의 변화를 지정 (transition-timing-function 속성 값)
animation-iteration-count	애니메이션의 반복 횟수 (<횟수> infinite)
animation-direction	키 프레임의 진행 방향 지정 normal 순방향 reverse 역방향 alternate 순→역 alternate-reverse 역→순
animation-play-state	애니메이션의 진행 또는 정지 상태를 지정 (running paused)
animation-delay	애니메이션 시작 시의 지연 시간을 지정 (초 단위)
animation-fill-mode	애니메이션의 실행 이전/이후에 애니메이션 효과의 표시 여부 none forwards backwards both
animation	animation-* 속성의 일괄 지정

예제 애니메이션

```

<!DOCTYPE html> <html> <head>
<style>
  @-webkit-keyframes myAnimation {
    0% {width: 100px; left: 0px; background: orange;}
    20% {width: 150px; left: 50px; background: green;}
    40% {width: 200px; left: 100px; background: red;}
    60% {width: 150px; left: 150px; background: yellow;}
    100% {width: 100px; left: 200px; background: blue;}
  }
  div {width:100px; height: 100px;
    position:relative;background: orange;
    -webkit-animation: myAnimation 5s 0.5s ease-in infinite normal; }
</style>
</head>
<body>   <div> Animation 속성 </div> </body>
</html>

```



6. 미디어 쿼리 media query

6.1 미디어 쿼리

- 같은 스타일시트 내에서 서로 다른 미디어에 다른 스타일을 적용하는 경우 사용한다.
- 서로 다른 스타일시트를 사용하여, 텍스트를 화면에 표시하기 위해 sans-serif를 사용하고 프린터에 인쇄할 때는 serif를 사용하는 방법이다.

```
<link rel="stylesheet" media="screen" href="sans-serif.css">
```

```
<link rel="stylesheet" media="print" href="serif.css">
```

- 스타일 시트 내에서 한 섹션이 특정 미디어 유형에 적용 될 수 있도록 선언할 수 있다.

```
@media screen { * { font-family: sans-serif } }
```

6.2 미디어 타입의 종류

all	모든 미디어 타입
aural	음성 합성 장치
braille	점자 표시 장치
handheld	작은 휴대용 장치
print	인쇄 장치
projection	프로젝터 장치
screen	컴퓨터 화면 장치
tty	텔렉스, 터미널 등과 같이 폭이 일정한 문자 간격을 가지는 기기
tv	텔레비전
embossed	종이에 인쇄하는 점자 장치

↳ 대소문자 구별

6.3 사용 방법

- link 요소를 이용하는 방법
 - 스타일시트 파일을 선택적으로 사용할 수 있다.

```
<link rel="stylesheet" media="screen" and (color) href="example.css">
```

```
@import url(example.css) screen and (color);
```

- style 요소의 media 속성을 사용하는 방법
 - 미디어 유형에 따라서 선택적으로 스타일을 적용시킬 수 있다.

```
<style media="screen">
  p { font-family: sans-serif; font-size: 14px }
</style>
```

```
<style media="print">
  p { font-family: serif; font-size: 12px }
</style>
```

- @media 규칙의 사용
 - 특정 미디어 유형에만 선택적으로 스타일을 적용시킬 수 있다.

```

<style>
  @media screen { p { font-family: sans-serif; font-size: 14px } }
  @media print { p { font-family: serif; font-size: 12px } }
  @media screen, print { p { font-weight: bold } }
</style>

```

6.4 문법과 조건

- 미디어 쿼리 문법은 다음의 형태를 가지고 있으며, 콤마(,)로 구분하여 미디어 쿼리 목록에서 결합될 수 있다.
 - ▶ @media [not/only] 미디어타입 [and (미디어기능)]*



```

@media screen and (color), projection and (color) { ... }
@media screen and (min-width: 320px) and (max-width: 800px) { ... }
@media not screen { ... }
@media only screen { ... }

```

6.5 미디어 기능

출력 장치의 대상 표시 영역의 폭	width, min-width, max-width
출력 장치의 대상 표시 영역의 높이	height, min-height, max-height
출력 장치가 제공하는 기본적인 폭	device-width, min-device-width, max-device-width
출력 장치가 제공하는 기본적인 높이	device-height, min-device-height, max-device-height
화면의 가로 및 세로 방향	orientation (landscape portrait)
화면 비율	aspect-ratio, min-aspect-ratio, max-aspect-ratio
출력 장치 디스플레이의 물리적인 화면 비율	device-aspect-ratio, min-device-aspect-ratio, max-device-aspect-ratio
출력 장치의 색 성분에 할당되는 비트 수	color, min-color, max-color
색상 룩업 테이블에 있는 전체 색상 수	color-index, min-color-index, max-color-index
출력 장치의 해상도	resolution, min-resolution, max-resolution
TV 미디어의 화면 주사 방식	scan (progressive interlace)
출력장치의 최소 단위	grid

예제 미디어쿼리

```

<!DOCTYPE html> <html> <head>
<style>
img { width: 100%; height: 100% }
div { width: 100%; height: 150px }
@media screen and (max-width: 300px) { div { background-color: yellow } }
@media screen and (min-width: 300px) and (max-width: 450px) {
    div { background-color: green } }
@media screen and (min-width: 450px) { div { background-color: blue } }
</style>
</head>
<body> <div> </div> </body>
</html>
    
```

