CSS 기초

〈 css 역사 〉

- ✓ **History**

  : 1997-2001, HTML 4.01, CSS1

  2001-2006, XHTML 1, CSS2

  2007-현재, HTML5, CSS3

〈 vendor prefix 〉

- ✓ **Some CSS rules won't work without the vendor prefix.**

  Mozilla Browsers (Firefox)

  : -moz

- ✓ **Webkit Browsers (Safari, Chrome)**

  : -webkit

- ✓ **Opera**

  : -o

- ✓ **Internet Explorer**

  : -ms

〈 Css Syntax 〉

✓ **Selector 속성 : 값 속성 : 값**

h1 {color: blue; font-size: 12px;}

```
<style>
p {
    color: red;
    text-align: center;
}
</style>
```

```
<body>
    <p>CSS Syntax</p>
    <p>CSS 스타일 적용된 P 태그</p>
</body>
```

〈 css 주석문 〉

/* CSS 주석문.. */

```
<style>
p {
    color: red;
    /* CSS 주석문.. */
    text-align: center;
}
</style>
```

Selector

〈 css 적용 형태 〉

✓ **Inline style**

- 태그에서 스타일 속성을 직접 사용
- 스타일을 바꿀 부분이 많지 않은 경우 사용
- 형식 : **<태그명 style="속성명: 값; 속성명: 값....."></**태그명**>**

```
<h2> 인라인 스타일 </h2>
<p> p 태그 스타일 적용하지 않음 </p>
<p style="color:red; font-size:30px;"> p 태그 스타일 적용</p>
```

〈 css 적용 형태 〉

✓ **Internal style sheet**

- 파일 내에 스타일을 적용하는 방식
- **<style> ~~ </style>** 태그 사이에 스타일을 명시
- **style** 태그는 **<head>~</head>** 태그 사이에 명시

```
<style type="text/css">
  h3 { color:blue; }
  p  { color : green; font-size:12px; margin-left:20px}
</style>
```

```
<H3> JAVA </H3>
<P> Java Programming </P>
<P> JDBC </P>
```

〈 css 적용 형태 〉

✓ **External style sheet**

- <link rel=stylesheet type="text/css" href="css 파일명" />

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

external.css

```
P { font-size:12px; color:blue }
.gBold { color: green; font-weight: bold }
.oBold { color: orange; font-weight: bold }
```

```
<p>나의 하루를 가만히 닫아주는 너</p>
<p class="gBold">은은한 달빛 따라 너의 모습 사라지고</p>
<p class="oBold">홀로 남은 골목길엔 수줍은 내 마음만</p>
```

전체 Selector

- 모든 요소에 스타일 적용하기

```
<STYLE TYPE="text/css">
  * {
      margin: 0;
      padding: 0;
  }
</STYLE>
```

```
<H3> WEB Programming </H3>
<P> Servlet&JSP </P>
<P> JavaScript </P>
```

태그 Selector

- 태그명 { 속성명: 값; 속성: 값... }
- <태그명>...</태그명>

```
<STYLE TYPE="text/css">
  H3 { color:blue; }
  P  { color : green; font-size:12px; margin-left:20px }
</STYLE>
```

```
<H3> WEB Programming </H3>
<P> Servlet&JSP </P>
<P> JavaScript </P>
```

클래스 selector

- . 클래스명 { 속성명: 값; 속성: 값... }
- <태그명 class="클래스명">...</태그명>

```
<STYLE type="text/css">
  P {font-size:12px; color:blue}
  .gBold {color:green; font-weight:bold}
</STYLE>
```

```
<H3> JAVA </H3>
<P> Java Programming </P>
<P class="gBold"> JDBC </P>
```

아이디 selector

- # 아이디 { 속성명: 값; 속성: 값... }
- <태그명 id="아이디"> ... </태그명>

```
P {font-size: 12px; color: blue}

#gBold {color: green; font-weight: bold}
```

```
<H3> JAVA </H3>

<P> Java Programming </P>

<P id="gBold"> JDBC </P>
```

〈 Css 선택 범주 스타일 〉

✓ 형식 : 태그 태그

- DIV UL { 속성명: 값; 속성: 값... }

  DIV 하위의 모든 UL 태그 에 적용

```
div ul {
    border: 2px dotted orange;
}
```

```
<div>
    <ul>
        <li>Java Programming
            <ul>
                <li>제어문 : 조건, 반복, 이동</li>
            </ul>
        </li>
    </ul>
</div>
```

❮ 자식 선택자 ❯

✓ 형식 : 태그 > 태그 (자식 태그만 선택)

- **DIV > UL { 속성명: 값; 속성: 값... }**

  DIV 하위 태그 자식 **UL** 태그에 적용

```
div > ul {
    border: 2px dotted orange;
}
```

```
<div>
    <ul>
        <li>Java Programming
            <ul>
                <li>제어문 : 조건, 반복, 이동</li>
            </ul>
        </li>
    </ul>
</div>
```

<인접 형제 선택자>

✓ 형식 : 태그 + 태그

- **h1 + ul { 속성명: 값; 속성: 값... }**

  h1 태그 바로 다음에 나오는 첫번재 **ul** 태그 선택

```
 h1 + ul {
    border: 2px dotted orange;
 }
```

```
<h1>목차</h1>
<ul>
   <li>자바의 특징 및 역사</li>
   <li>연산자</li>
   <li>제어문 : 조건, 반복, 이동</li>
</ul>
```

< 인접 형제 선택자 >

✓ 형식 : 태그 ~ 태그

- **h1 ~ ul { 속성명: 값; 속성: 값... }**

  h1 태그 다음에 나오는 모든 **ul** 태그 선택

```
h1 ~ ul {
    border: 2px dotted orange;
}
```

```
<h1>목차</h1>
<ul><li>자바의 특징 및 역사</li></ul>
<ul><li>자바의 특징 및 역사</li></ul>
```

<Css 그룹 선택자>

✓ 형식 : 태그, 태그, ..., 태그

h1, h2 { 속성명: 값; 속성: 값... }

```
h1, h2 {
    color: white;
    background-color: black;
}
```

```
<h1>목차</h1>
<h2>java</h2>
```

속성 selector

✓ 형식 : [속성]

[href] { 속성명: 값; 속성: 값... }

```
[href] {
    background-color: orange;
}
```

```
<a href="www.naver.com">네이버</a>
<a href="www.daum.com">다음</a>
```

속성 selector

✓ 형식 : [속성 = 값]

[type="text"] { 속성명: 값; 속성: 값... }

```
input[type="text"] {
    width: 200px;
    background-color: black;
}
```

```
<input type="text" name="id" />
<input type="text" name="name" />
```

속성 selector

✓ 형식 : [속성 ^= 값]

[href ^= “http://”] { 속성명: 값; 속성: 값... }

```
a[href ^= “http://”] {
    background-color: black;
    border: 3px solid #ccc;
}
```

```
<a href=“www.naver.com”>네이버</a>
<a href=“www.daum.com”>다음</a>
```

속성 selector

✓ 형식 : [속성 $= 값]

[href $= “.jpg”] { 속성명: 값; 속성: 값... }

```
a[href $= “.jpg”] {
    background-color: black;
    color: white;
}
```

```
<a href=“flower.jpg”>꽃</a>
<a href=“www.daum.com”>다음</a>
```

속성 selector

✓ 형식 : [속성 *= 값]

[id *= “blog”] { 속성명: 값; 속성: 값... }

```
input[id *= “blog”] {
    background-color: black;
    color: white;
}
```

```
<input type=“text” id=“my-blog” />
<input type=“text” id=“your-blog” />
```

가상 클래스

: active

: hover

: focus

```
-------------------------------------------------
a:active  { background-color: black; color: white; }
a:hover   { background-color: black; color: white; }
a:focus   { background-color: black; color: white; }
-------------------------------------------------
<a href="http://www.naver.com">네이버</a>
<a href="http://www.daum.net">다음</a>
-------------------------------------------------
```

상태 선택자

: checked

: focus

: disabled

```
-------------------------------------------------
input:focus    {background-color: black; color: white;}
input:disabled {background-color: black; color: white;}
-------------------------------------------------
<input type="text" />
<input type="text" disabled="disabled" />
-------------------------------------------------
```

위치 선택자

```
:first-child    : 부모의 첫번째 자식 선택
:last-child     : 부모의 마지막 자식 선택
:nth-child(n)   : 부모의 n 번째 자식을 선택
```

```
h1:nth-child(2n + 0) {background: #ff0000;}
h1:nth-child(2n + 1) {background: #00ff00;}
h1:first-child {background: orange; color: white;}
h1:last-child {background: black; color: white;}
```

```
<h1>1</h1>
<h1>2</h1>
<h1>3</h1>
```

Background

❮ 배경색 설정❯

- ✓ **background-color**
  - **HEX value - #ff0000**
  - **RGB value - rgb(255,0,0)**
  - **RGBA value – rgba(255, 0, 0, 0.2)**
  - **color name – red**
  - **HSL value - hsl(120, 100%, 50%) :**

    (색상 : 0(red) 120(green) 240(blue), 채도, 밝기)

```
h1  { background-color: orange; }
p   { background-color: rgb(255,0,0); }
div { background-color: #b0c4de; }
```

< 배경 이미지 >

- ✓ **background-image**
  - **background-image: url("파일명(경로포함)");**

```
body {
    background-image: url("back01.jpg");
}
```

‹ 배경 이미지 반복 ›

✓ **background-repeat**

- **repeat(기본 설정 속성), repeat-x, repeat-y, no-repeat**

```
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
}
```

❮ 배경 이미지 위치 조정 ❯

✓ **background-position**

- 수평 : left, center, right

- 수직 : top, center, bottom

```
body {
    background-image: url("pic1.jpg");
    background-repeat: no-repeat;
    background-position: right top;
}
```

❮ 배경이미지 고정 ❯

✓ **background-attachment**

- **fixed, scroll**

```
body {
    background-image: url("pci1.jpg");
    background-repeat: no-repeat;
    background-position: right top;
    background-attachment: scroll;
}
```

❮ 배경과 연관된 전체 속성 명시 ❯

✓ **background**

- 모든 속성을 명시

```
background: #ffffff
            url("/css/images/pic1.jpg")
            no-repeat
            fixed
            right top ;
```

〈 background 〉 엘리먼트의 배경에 대한 효과 정의

✓ **background-size :** 배경 이미지 크기 조절

- auto

- 크기값 **:** 너비값과 높이값을 지정, 너비값만 지정할 경우 원래 배경 이미지 크기를 기준으로 축소/확대 배율을 계산해 높이 값을 자동 계산

- 백분율 **:** 원래 배경 이미지 크기를 기준으로 확대하거나 축소

- cover : 배경이미지를 엘리먼트의 큰쪽에 맞추어 비율을 유지한 채 확대 또는 축소

- contain : 배경이미지를 엘리먼트의 작은쪽에 맞추어 비율을 유지한 채 확대 또는 축소

〈 background 〉 엘리먼트의 배경에 대한 효과 정의

```
-------------------------------------------------
background-size: auto;
background-size: 300px;
background-size: 200px 150px;
background-size: 100%;
background-size: cover;
background-size: contain;
-------------------------------------------------
```

〈 background 〉 엘리먼트의 배경에 대한 효과 정의

✓ **background-clip :** 배경 적용 범위 조절

- border-box : 박스 모델의 가장 외곽인 테두리**(border)** 까지 적용

- padding-box : 박스 모델에서 테두리를 뺀 패딩까지 적용

- content-box : 박스 모델에서 내용 부분에만 적용

```
-----------------------------------------------
#clip1 { background-clip: border-box; }
#clip2 { background-clip: padding-box; }
#clip3 { background-clip: content-box; }
-----------------------------------------------
```

Font

〈 font 〉

✓ 글꼴의 종류 지정

- **font-family : “Times New Roman” , 돋움, 굴림**

✓ 글꼴의 크기 지정

- **font-size ： px, em, %(부모요소 기준)**

✓ 글꼴의 굵기 지정

- **font-weight**

: normal(보통 : 400), bold(700), lighter(한단계 낮게),

bolder(한단계 굵게), 100 - 900 (백단위)

〈 font 〉

✓ 보통 글자인지 기울어진 글자인지 지정

- font-style

: normal | italic | oblique

✓ 작은 대문자를 사용할 지 지정

- font-variant

: normal | small-caps

✓ 글꼴 전체 속성 지정

- font

Text

〈 text 〉

✓ 텍스트 색상

- **color : #ff0000**
- **HEX value : #ff0000**
- **RGB value : rgb(255,0,0)**
- **RGBA value : rgba(255,0,0, 0.5)**
- **color name : red**

```
H1 { color:aqua; }
H2 { color:blue; }
H3 { color:gray; }
H4 { color:lime; }
```

〈 text 〉

✓ 텍스트 정렬

- text-align : left(기본), center, right,

justify (왼쪽과 오른쪽에 알맞게 정렬);

✓ 텍스트 장식

- text-decoration: none | underline | overline | line-through

✓ 영문자의 대, 소문자 및 단어의 앞자리만 대문자로 변경할 지 결정

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

✓ 단락의 첫번째 줄을 얼마나 띄울지 결정

- text-indent: 크기 지정

〈 text 〉

✓ 기타

- line-height : 줄 사이 간격
- letter-spacing : 텍스트 글자 사이의 간격
- word-spacing : 단어 사이의 간격

✓ 넘치는 텍스트 표기하기

- text-overflow : 속성

: clip(넘치는 텍스트를 잘라냄)

: ellipsis(말줄임표 ... 표시)

〈 text 〉

✓ 그림자 효과 내기

- text-shadow : h-shadow | v-shadow | blur | 색상

: 수평거리(필수)

- 그림자가 수평으로 얼마나 떨어져 있는지 지정, 양수값은 요소의 오른쪽에 음수값은 요소의 왼쪽에 그림자 표시

: 수직거리(필수)

- 그림자가 수직으로 얼마나 떨어져 있는지 지정, 양수값은 요소의 위쪽에 음수값은 요소의 아래쪽에 그림자 표시

: blur

- 그림자의 흐림 정도, 생략시 0을 기본값으로 진한 그림자, 음수 불가능, 값이 커지면 흐려짐

: 색상

- 그림자 색상

Box

〈 padding 〉

엘리먼트와 컨텐츠 사이의 간격

- ✓ **padding-top: 25px;**
- ✓ **padding-right: 50px;**
- ✓ **padding-bottom: 25px;**
- ✓ **padding-left: 50px;**

- ✓ **padding:** 위쪽 오른쪽 아래쪽 왼쪽

〈 margin 〉

엘리먼트와 엘리먼트 사이의 간격

- ✓ **margin-top: 25px;**
- ✓ **margin-right: 50px;**
- ✓ **margin-bottom: 25px;**
- ✓ **margin-left: 50px;**

- ✓ **margin:** 위쪽 오른쪽 아래쪽 왼쪽

〈 Box 〉 테두리선 속성

✓ 스타일

none, dotted

dashed(짧은 선으로 표시, 직선으로 된 점선**)**

solid, double(테두리를 이중선(겹선)으로 표시**)**

groove(테두리를 창에 조각된 것처럼 표시, 홈이 파인 듯 입체적으로 보임**)**

ridge(테두리를 창에서 튀어나온 듯이**), inset, outset**

✓ **border-bottom-style, border-left-style, border-right-style, border-top-style, border-style**

〈 Box 〉 테두리선 속성

✓ 굵기

- thin, medium, thick, 길이값

- border-bottom-width

- border-left-width

- border-right-width,

- border-top-width

- border-width

〈 Box 〉 테두리선 속성

✓ 색상

- border-bottom-color

- border-left-color

- border-right-color,

- border-top-color

- border-color

〈 Box 〉 테두리선 속성

✓ 종합

- border-bottom

- border-left

- border-right

- border-top

- border

〈 Box 〉 테두리선 속성

✓ 박스 모서리 둥글게 만들기

- border -radius

: 크기, 백분율

: 지정하는 반지름 값에 따라 모서리의 완만함이 조절 가능

- border-*-radius

: top-left, top-right, bottom-left, bottom-right

〈 Box 〉 테두리선 속성

✓ 그림자 효과 내기

➢ **box-shadow :**

▪ 수평거리 **|** 수직거리 **| blur |** 확장거리 **|** 색상 **| inset**

수평거리**(**필수**) :** 그림자가 수평으로 얼마나 떨어져 있는지 지정,
양수값은 요소의 오른쪽에 음수값은 요소의
왼쪽에 그림자 표시

수직거리**(**필수**) :** 그림자가 수직으로 얼마나 떨어져 있는지 지정,
양수값은 요소의 위쪽에 음수값은 요소의
아래쪽에 그림자 표시

〈 Box 〉 테두리선 속성

blur : 그림자의 흐림정도, 생략시 0을 기본값으로
진한 그림자, 음수 불가능, 값이 커지면 흐려짐

확장거리 : 그림자의 번짐 정도, 양수값을 사용하면
모든 방향으로 그림자가 퍼짐으로 박스보다
그림자가 크게 보임, 음수 값은 모든 방향으로
그림자 축소

색상 : 그림자 색상

inset : 키워드 사용시 박스 안쪽에 그림자 표시, 사용
하지 않으면 박스 바깥쪽에 그림자 표시

Layer

<layer> 개요

- ✓ 단순히 영역만을 만들어 주는 개념
- ✓ 레이어는 겹쳐 쌓을 수 있음**(z-index)**
- ✓ 레이어는 위치 조정이 가능**(left, top)**
- ✓ 레이어는 가시성 조절이 가능**(visibility)**

〈 layer 〉

- ✓ **position**
  - 위치를 계산
  - **static(default), fixed, absolute, relative**
  - **static : left, top** 좌표를 설정하더라도 값이 반영되지 않음
  - **absolute :** 절대좌표로 지정**(top, right, bottom, left)**
  - **relative :** 정상적인 흐름에 따라서 배치되면서 상대적인 거리 **(top, right, bottom, left)**를 계산

〈 layer 〉

- ✓ **z-index**
  - 박스가 겹쳐있는 경우 보여지는 순서결정
  - **auto,** 숫자
- ✓ **visibility, display** 속성을 이용하여 레이어의 화면 보임과 보이지 않음 처리
  - **visibility : hidden, visible**
  - **display : none, block**

〈 layer 〉

- ✓ **float**
  - 특정 요소를 떠 있게 하는 것
  - **float** 속성을 이용하여 웹페이지의 요소를 오른쪽이나 왼쪽에 배치하면 그 다음 요소에도 똑같은 속성이 전달
  - **float** 속성을 사용할 때 요소의 위치가 고정되면 안 되기 때문에 **position** 속성의 **absolute**를 사용하면 안됨
  - 태그들의 수평 정렬 또는 화면 레이아웃 구성에 주로 사용
  - **none, left, right**

〈 layer 〉

- ✓ **clear**
  - 지정한 값의 방향에 떠다니는 박스를 허용하지 않음
  - **none, left, right, both(right, left** 인지 모를 때 **/** 가장 많이 사용**)**

- ✓ **overflow**
  - 컨텐츠가 박스의 범위를 넘어가는 경우에 처리
  - **visible, hidden, scroll, auto**

Transition

〈transition〉 개요

- ✓ 시간이 흐르면서 설정된 스타일을 변경하는 것

- ✓ 자바스크립트를 이용하지 않고도 애니메이션 효과를 낼 수 있음

〈 transition〉

✓ **transition-property : 속성 값**

1. none : 아무 속성도 바뀌지 않음

2. all : 요소의 모든 속성이 트랜지션 대상이 됨

3. 특정 속성명 : 트랜지션을 적용할 속성을 지정, 여러 개일 경우 쉼표 구분

```
transition-property: none;
transition-property: all;
transition-property: background-color;
```

〈transition〉

✓ **transition-duration : 시간 지정(초단위)**

- 트랜지션이 적용될 시간을 설정

```
transition-duration: 1s;
```

✓ **transition-delay : 시간**

: 트랜지션 시작 시간의 지연시간을 설정

: 사용할 수 있는 지연시간은 초나 밀리초

: 기본값은 0s 임

```
transition-delay:2s;
```

〈transition〉

✓ **transition-timing-function**

: 트랜지션 효과의 속도를 지정할 수 있음

: 설정 함수명

1. linear : 시작에서 끝까지 똑같은 속도로 진행

2. ease : 처음에는 천천히 시작하고 점점 빨라지다가 마지막에는 천천히

3. ease-in : 트랜지션 시작을 느리게

4. ease-out 트랜지션을 느리게 끝냄

5. ease-in-out : 느리게 시작하고 느리게 끝냄

〈 transition〉

✓ **transition : 위의 4가지를 동시 사용, 필수(duration)**

```
- transition: 2s;
- transition: all 2s;
- transition: all 2s ease-in;
- transition: all 2s ease-in 0.5s;
```

Animation

〈 Animation 〉 @keyframes

애니메이션이 진행되는 상태에 따른 변경사항을 적용

- **from**
- **To**

```
@keyframes 이름 {
        from {background-color: red;}
        to {background-color: yellow;}
    }
```

- %

〈 Animation 〉 @keyframes

```
@keyframes 이름 {
        0%   {background-color: red;}
        25%  {background-color: yellow;}
        50%  {background-color: blue;}
        100% {background-color: green;}
    }
```

〈 Animation 〉 animation

생성된 @keyframes을 적용

- **animation-name: keyframes** 이름;
- **animation-duration :** 시간**(초);**
- **animation-delay:** 시간**(초);**
- **animation-iteration-count:** 반복횟수 **| infinite(**무한**);**
- **animation-direction: normal(**정상**) | reverse(**반대**)**

  | alternate(정상 **<->** 반대**);**
- **animation-timing-function: linear;**

〈 Animation 〉 animation

- ✓ **linear :** 시작에서 끝까지 똑같은 속도로 진행
- ✓ **ease :** 처음에는 천천히 시작하고 점점 빨라지다가 마지막엔 천천히
- ✓ **ease-in :** 트랜지션 시작을 느리게
- ✓ **ease-out :** 트랜션을 느리게 끝냄
- ✓ **ease-in-out :** 느리게 시작하고 느리게 끝냄

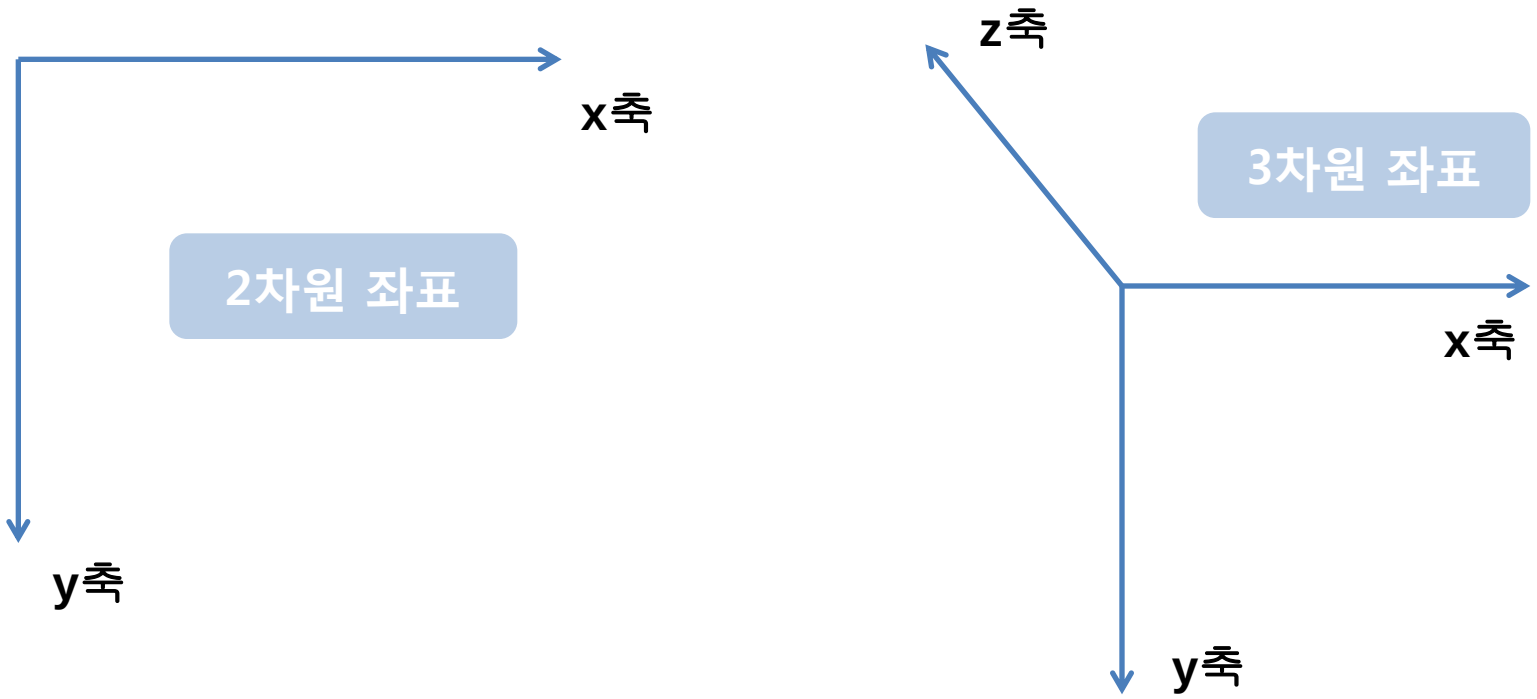
〈 Animation 〉 animation

```
----------------------------------------------------
div {
        width: 100px; height: 100px;
        background-color: red;
        animation-name: example;
        animation-duration: 5s;
        animation-timing-function: linear;
        animation-delay: 2s;
        animation-iteration-count: infinite;
        animation-direction: alternate;
        /* 단축 */
        animation: example 5s linear 2s infinite alternate;
}
----------------------------------------------------
```

Transform

〈transform〉 개요

✓ 웹 요소의 위치를 옮기거나 크기를 조절하고 회전, 왜곡시키는 것



✓ x, y 축 값이 커질수록 오른쪽, 아래쪽으로 내려감

✓ z 축은 값이 커질수록 앞으로 작을 수록 뒤로

〈transform〉 변형함수

- ✓ **translate(이동)**

  1. translate(tx, ty)

  2. translateX(tx)

  3. translateY(ty)

  4. translateZ(tz)

〈transform〉 변형함수

- ✓ **scale(확대 / 축소)**

  1. **scale(tx, ty)**
  2. **scaleX(tx)**
  3. **scaleY(ty)**
  4. **scaleZ(tz)**

〈transform〉 변형함수

- ✓ **rotate(회전)**

  1. rotate(각도)

  2. rotateX(각도)

  3. rotateY(각도)

  4. rotateZ(각도)

  - 각도가 양수일 경우 시계방향 회전, 음수는 반시계 방향

〈transform〉 변형함수

✓ **4. skew(비틀기)**

1. skew(x각도, y각도)

2. skewX(x각도)

3. skewY(y각도)

〈transform-origin〉 x축, y축 값

- ✓ **x축**
  - 원점에서의 좌표나 길이 값이나 백분율
  - **left, center, right**
- ✓ **y축**
  - 원점에서의 좌표나 길이 값이나 백분율
  - **top, center, bottom**