L5-CSS

예제 실습 및 보고서 작성



학 과 : 컴퓨터공학전공

과 목 : 웹프로그래밍

교수님 : 한영선 교수님

학 번 : 202030494

이 름 : 윤원재

제출일 : 2023년 04월 10일

1. **예제 5-1 display 속성으로 박스 유형 설정**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**display 속성**</title>**

**<style>**

div {

border : 2px solid yellowgreen;

color: blue;

background : aliceblue;

}

span {

border: 3px dotted red;

background: yellow;

}

**</style>**

**</head>**

**<body>**

**<h3>**인라인, 인라인 블록, 블록**</h3>**

나는 **<div style="display:none">**

div(none)**</div>**입니다.**<br><br>**

나는 **<div style="display:inline">**

div(inline)**</div>**입니다.**<br><br>**

나는 **<div style="display:inline-block; height: 50px;">**

div(inline-block)**</div>**입니다.**<br><br>**

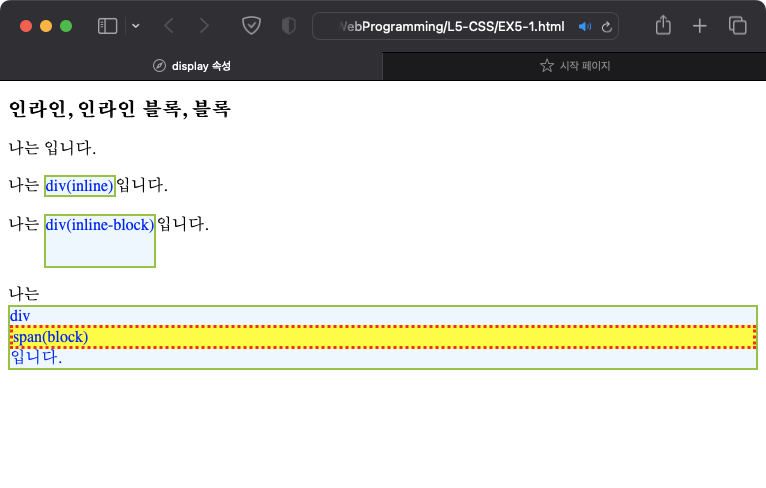
나는 **<div>**div**<span style="display:block">**

span(block)**</span>**입니다.

**</div>**

**</body>**

**</html>**

****

<그림 1-1> 예제 5-1 결과

HTML 태그는 인라인 태그와 블록 태그로 나뉘고 각각 인라인 박스와 블록 박스로 출력된다. 하지만 ‘display’ 프로퍼티를 사용하여 CSS 스타일을 적용시키면 각 태그의 출력되는 박스를 바꿔서 지정할 수 있다. <그림 1-1>에서 첫번째 줄은 ‘none’으로 적용되어 태그가 화면에 출력되지 않는다. 두번째 줄은 ‘inline’으로 인라인 박스가 적용되었다. 인라인 박스는 블록 박스 내에 라인 안에 있으며 옆에 다른 요소를 배치할 수 있다. 다만 크기 조절은 불가능하다. 세번째 줄은 ‘inline-block’으로 인라인 블록 박스가 적용되었다. ‘inline’ 속성과 ‘block’ 속성이 합쳐진 것으로 크기 및 테두리 조절이 가능한 인라인 박스이다. 마지막 줄은 ‘block’으로 블록 박스가 적용되었다. 항상 새 라인에서 시작하며 옆에 다른 요소를 배치할 수 없다. 크기 및 테두리 조절이 가능하다.

1. **예제 5-2 position : relative 상대 배치**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**relative 배치**</title>**

**<style>**

div {

display : inline-block;

height : 50px;

width : 50px;

border: 1px solid lightgray;

text-align: center;

background: red;

}

#down:hover {

position: relative;

left: 20px; top: 20px;

background : green;

}

#up:hover {

position: relative;

right : 20px; bottom : 20px;

background : green;

}

**</style>**

**</head>**

**<body>**

**<h3>**상대 배치, relative**</h3>**

**<div>**T**</div>**

**<div id="down">**h**</div>**

**<div>**a**</div>**

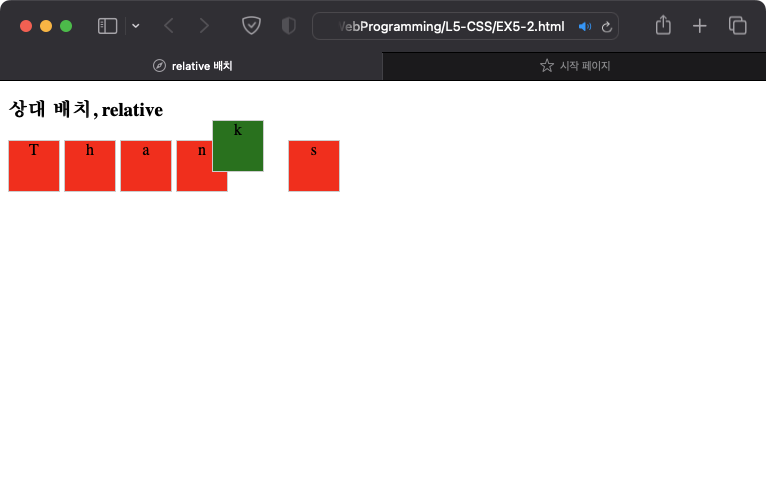
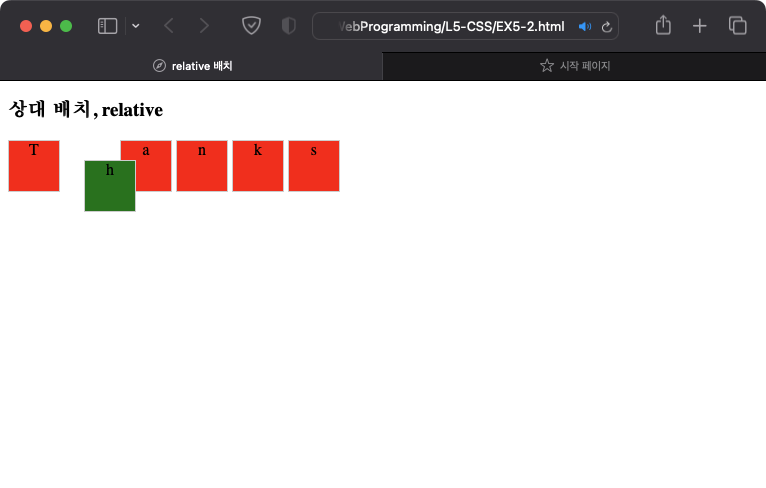
**<div>**n**</div>**

**<div id="up">**k**</div>**

**<div>**s**</div>**

**</body>**

**</html>**

****

<그림 2-1> 예제 5-2 결과

CSS의 ‘position’속성을 사용하여 박스를 원하는 위치에 배치할 수 있다. 또한 ‘top’, ‘bottom’, ‘left’, ‘right’, ‘width’, ‘height’ 속성으로 태그의 위치와 크기를 지정할 수 있다. <그림 2-1>은 ‘position’속성이 상대 배치를 설정하는 ‘relative’로 되어있다. 상대 배치를 하면 기본 위치에서 ‘top’, ‘bottom’, ‘left’, ‘right’ 속성의 값 만큼 이동하게 된다. <그림 2-1>에서 ‘down’ id 셀렉터가 적용된 ‘h’와 ‘up’ id 셀렉터가 적용된 ‘k’를 되었다. 각 박스에 마우스를 올리면 기본 위치에서 ‘h’는 아래, 오른쪽으로 20px만큼, ‘k’는 위, 왼쪽으로 20px만큼 이동한다.

1. **예제 5-3 position : absolute 절대 배치**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**절대 배치**</title>**

**<style>**

div {

display: inline-block;

position: absolute;

border: 1px solid lightgray;

}

div > p {

display : inline-block;

position : absolute;

height: 20px;

width: 15px;

background: lightgray;

}

**</style>**

**</head>**

**<body>**

**<h3>**Merry Christmas!**</h3><hr>**

**<div>**

**<img src="media/christmastree.png" width="200" height="200" alt="크리스마스 트리">**

**<p style="left:50px; top:30px">**M**</p>**

**<p style="left:100px; top:0px">**E**</p>**

**<p style="left:100px; top:80px">**R**</p>**

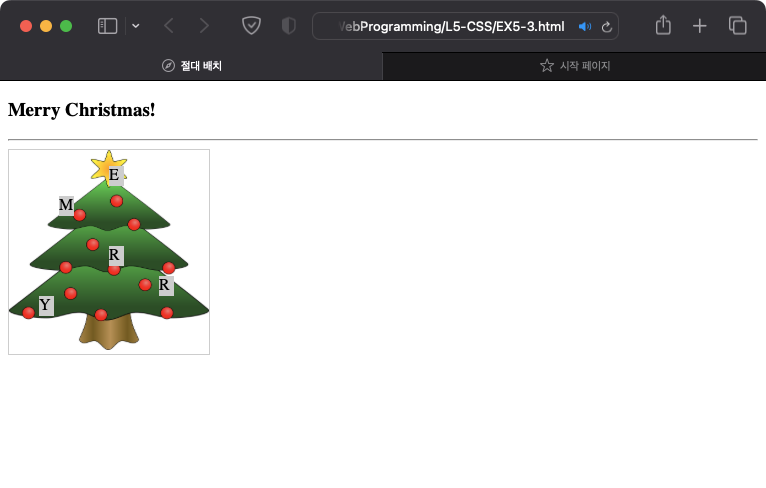
**<p style="left:150px; top:110px">**R**</p>**

**<p style="left:30px; top:130px">**Y**</p>**

**</div>**

**</body>**

**</html>**

****

<그림 3-1> 예제 5-3 결과

‘position’ 속성을 ‘absolute’로 하여 절대 배치를 설정할 수 있다 <그림 3-1>에서 각 <p> 태그들은 ‘style’ 속성으로 지정된 위치로 고정되어 화면에 출력된다. 예를 들어 ‘M’의 경우 <div> 태그 블록 박스 내의 공간에서 절대적 위치인 (50px, 30px)로 위치가 지정되어 출력된다.

1. **예제 5-4 position : fixed로 브라우저 하단 오른쪽에 고정 배치**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**고정 배치**</title>**

**<style>**

#fixed {

position: fixed;

bottom : 10px;

right : 10px;

width: 100px;

padding : 5px;

background: red;

color: white;

}

**</style>**

**</head>**

**<body>**

**<h3>**Merry Christmass!**</h3>**

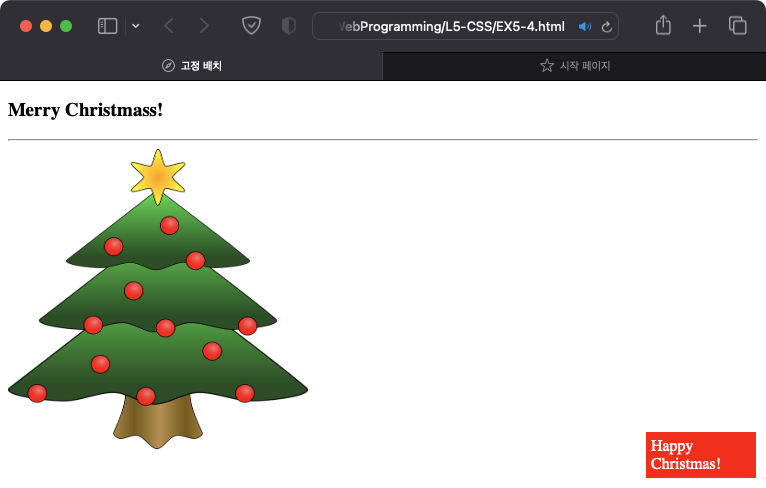
**<hr>**

**<img src="media/christmastree.png" width="300" height="300" alt="크리스마스 트리">**

**<div id="fixed">**Happy Christmas!**</div>**

**</body>**

**</html>**

****

<그림 4-1> 예제 5-4 결과

‘position’ 속성을 ‘fixed’로 하여 고정 배치를 설정할 수 있다. 또한 ‘top’, ‘bottom’, ‘left’, ‘right’ 속성으로 태그의 위치를 지정할 수 있다. <그림 4-1>에서 ‘Happy Christmas!’가 적힌 <div>태그 블록 박스가 아래에서 10px만큼, 오른쪽에서 10px만큼의 거리에 고정된 것을 확인할 수 있다. 만약 브라우저 창의 크기를 조절하더라도 지정된 거리만큼의 거리에 고정되어 있게 된다.

1. **예제 5-5 float : right로 브라우저의 오른편에 항상 배치**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**float 배치**</title>**

**<style>**

#float {

float : right;

border : 1px dotted black;

width : 7em;

padding : 0.25em;

margin : 1em;

}

**</style>**

**</head>**

**<body>**

**<h3>**Merry Christmas!**</h3>**

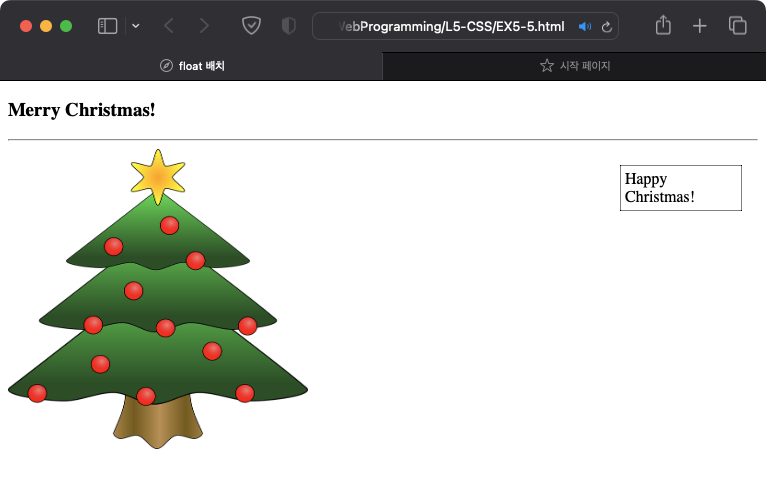
**<hr>**

**<img src="media/christmastree.png" width="300" height="300" alt="크리스마스 트리">**

**<div id="float">**Happy Christmas!**</div>**

**</body>**

**</html>**

****

<그림 5-1> 예제 5-5 결과

‘float’ 속성을 사용하여 레이아웃을 지정할 수 있다. <그림 5-1>에서 ‘float’ 속성을 ‘right’로 설정하여 <div>태그의 내용이 항상 브라우저의 오른편에 배치되도록 하였다.

1. **예제 5-6 z-index로 카드 쌓기**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**z-index 프로퍼티**</title>**

**<style>**

div { position: absolute;}

img { position: absolute;}

#spadeA { z-index: 0; left: 10px; top: 20px;}

#spade2 { z-index: 1; left: 40px; top: 30px;}

#spade3 { z-index: 2; left: 80px; top: 40px;}

#spade7 { z-index: 3; left: 120px; top: 50px;}

**</style>**

**</head>**

**<body>**

**<h3>**z-index 프로퍼티**</h3><hr>**

**<div>**

**<img id="spadeA" src="media/spade-A.png" width="100" height="140" alt="스페이드A">**

**<img id="spade2" src="media/spade-2.png" width="100" height="140" alt="스페이드2">**

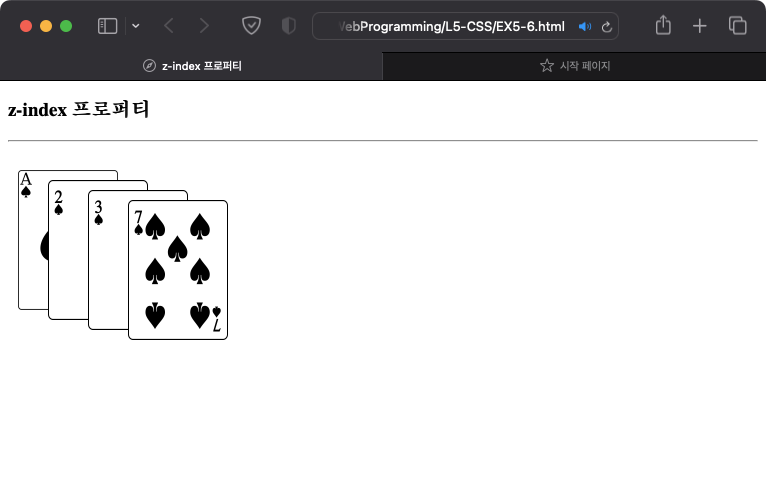
**<img id="spade3" src="media/spade-3.png" width="100" height="140" alt="스페이드3">**

**<img id="spade7" src="media/spade-7.png" width="100" height="140" alt="스페이드7">**

**</div>**

**</body>**

**</html>**

****

<그림 6-1> 예제 5-6 결과

HTML에서 각 블록들의 위치를 설정하다 보면 겹치는 일이 발생한다. 이때 ‘z-index’ 속성을 사용하면 겹치는 요소들을 쌓이는 순서를 지정하여 출력할 수 있다. ‘z-index : (순서 값)’의 형식으로 사용하며 양수나 음수 모두 순서 값으로 설정할 수 있다. 값이 작을수록 뒤로 배치되고 클수록 앞으로 배치된다. <그림 6-1>에서 ‘z-index’ 속성을 사용하여 카드 그림들이 겹쳐진 모양으로 출력된 것을 확인할 수 있다.

1. **예제 5-7 visibility로 텍스트 숨기기**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**visibility 프로퍼티**</title>**

**<style>**

span {

visibility: hidden;

/\* visibility: visible; \*/

}

**</style>**

**</head>**

**<body>**

**<h3>**다음 빈 곳에 숨은 단어는?**</h3>**

**<hr>**

**<ul>**

**<li>** I (**<span>**love**</span>**) you.

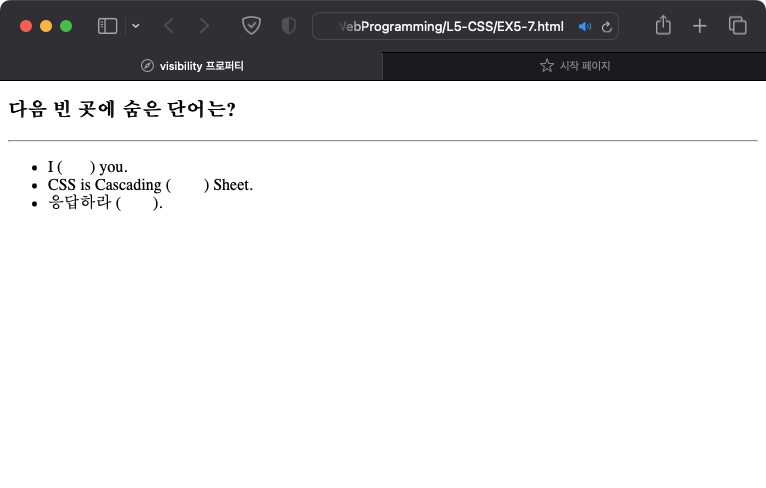
**<li>** CSS is Cascading (**<span>**Sytle**</span>**) Sheet.

**<li>** 응답하라 (**<span>**1988**</span>**).

**</ul>**

**</body>**

**</html>**

****

<그림 7-1> 예제 5-7 결과

‘예제 5-1’에서 ‘display’ 속성을 사용하여 블록을 어떤 형식으로 출력할지에 대한 내용을 다루었다. 하지만 단순히 표현될지 아닐지를 결정할 때에는 ‘visibility’ 속성을 사용하면 된다. ‘visibility’ 속성값은 ‘visible’, ‘hidden’, ‘collapse’가 있다. ‘visible’은 웹 페이지에 해당 요소를 출력할 때 사용한다. ‘hidden’은 웹 페이지에서 나타내지 않을 때 사용한다. 단, 레이아웃은 존재하기 때문에 요소의 공간은 그대로 유지된다. 끝으로 ‘collapse’는 테이블에서 사용하는 ‘hidden’이라 이해하면 쉽다. <그림 7-1>에서 <span>태그의 요소들을 ‘visibility: hidden’으로 설정하여 괄호 안이 빈 공간으로 출력된 것을 확인할 수 있다.

1. **예제 5-8 overflow 프로퍼티 활용**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**overflow 프로퍼티**</title>**

**<style>**

p {

width: 15em;

height: 3em;

border : 1px solid lightgray;

}

.hidden { overflow: hidden; }

.visible { overflow: visible; }

.scroll { overflow: scroll; }

**</style>**

**</head>**

**<body>**

**<h3>**overflow 프로퍼티**</h3>**

**<hr>**

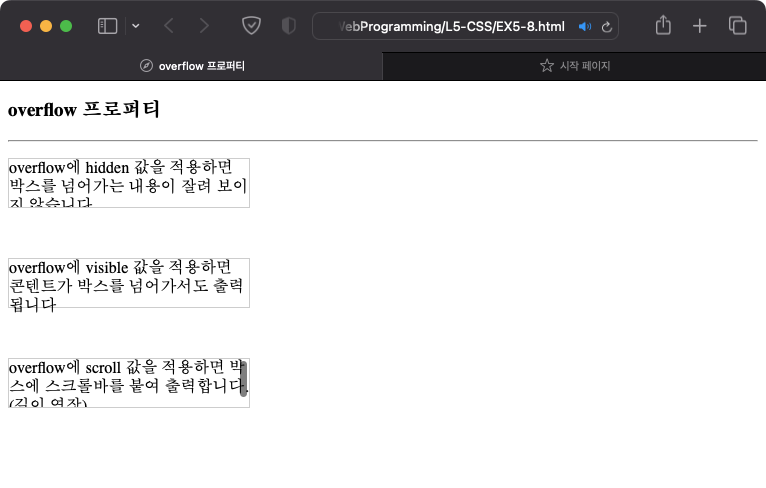
**<p class="hidden">**overflow에 hidden 값을 적용하면 박스를 넘어가는 내용이 잘려 보이지 않습니다.**</p><br>**

**<p class="visible">**overflow에 visible 값을 적용하면 콘텐트가 박스를 넘어가서도 출력됩니다**.</p><br>**

**<p class="scroll">**overflow에 scroll 값을 적용하면 박스에 스크롤바를 붙여 출력합니다. (길이 연장)**</p><br>**

**</body>**

**</html>**

****

<그림 8-1> 예제 5-8 결과

블록의 크기를 설정하고 해당 블록 내에 내용을 작성하다 보면 블록의 크기를 벗어나는 경우가 발생한다. 이때 ‘overflow’속성을 사용하여 어떻게 출력할 것인가를 설정할 수 있다. <그림 8-1>에서 첫번째 박스는 ‘hidden’으로 설정되어 내용이 박스를 넘어갈 경우 잘려서 출력되지 않았다. 두번째 박스는 ‘visible’로 설정되어 내용이 박스를 넘어가서도 잘리지 않고 출력되었다. 세번째 박스는 ‘scroll’로 설정되어 내용이 박스를 넘어간다면 스크롤바가 생겨 스크롤을 통해 내용을 볼 수 있게 된다.

1. **예제 5-9 CSS3 스타일을 응용하여 리스트로 메뉴 만들기**

**<body>**

**<nav id="menubar">**

**<ul>**

**<li><a href="#">**Home**</a></li>**

**<li><a href="#">**Espresso**</a></li>**

**<li><a href="#">**Cappuccino**</a></li>**

**<li><a href="#">**Cafe Latte**</a></li>**

**<li><a href="#">**F.A.Q**</a></li>**

**</ul>**

**</nav>**

**</body>**

**</html>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**리스트로 메뉴 만들기**</title>**

**<style>**

#menubar {

background: olive;

}

#menubar ul {

margin : 0;

padding : 0;

width : 567px;

}

#menubar ul li {

display : inline-block;

list-style-type: none;

padding : 0px 15px;

}

#menubar ul li a {

color: white;

text-decoration : none;

}

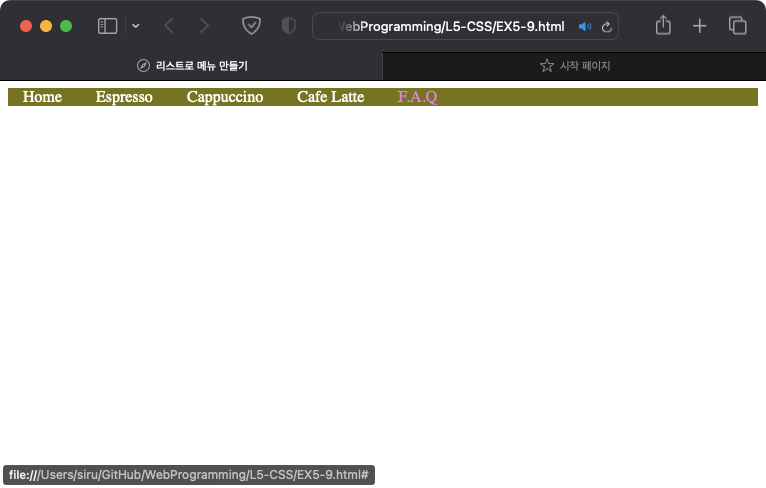
#menubar ul li a:hover {

color: violet;

}

**</style>**

**</head>**

****

<그림 9-1> 예제 5-9 결과

CSS 스타일을 활용하여 HTML의 리스트를 꾸밀 수 있다. ‘list-style-type’ 속성, ‘list-style-image’ 속성, ‘list-style-position’ 속성, ‘list-style’ 속성이 리스트를 꾸미는 데에 사용된다. 우선 ‘list-style-type’ 속성은 마커 종류를 변경할 때 사용한다. ‘list-style-image’ 속성은 마커를 이미지로 지정할 때 사용한다. ‘list-style-image : url(“이미지 주소”)’ 형식으로 사용하며 이미지 주소에 사용하고자 하는 이미지의 위치를 지정한다. ‘list-style-position’ 속성은 마커의 위치를 지정할 때 사용한다. <그림 9-1>에서 <ul>태그, <li>태그, <a>태그로 하이퍼링크 리스트를 만든 다음 ‘display : inline-block;’를 사용하여 인라인 블록 박스로 설정하였다. ‘list-style-type: none;’를 사용하여 리스트 마커를 지웠다. 텍스트 꾸미기 효과를 사용하여 밑줄을 없애고 마우스를 올릴 경우 ‘violet’색으로 바뀌도록 설정하였다.

1. **예제 5-10 마우스가 올라오면 행의 배경색이 변하는 표 만들기**

**<body>**

**<h3>**1학기 성적**</h3><hr>**

**<table>**

**<thead>**

**<tr><th>**이름**</th><th>**HTML**</th><th>**CSS**</th></tr>**

**</thead>**

**<tfoot>**

**<tr><th>**합**</th><th>**310**</th><th>**249**</th></tr>**

**</tfoot>**

**<tbody>**

**<tr><th>**황기태**</th><th>**80**</th><th>**70**</th></tr>**

**<tr><th>**이재문**</th><th>**95**</th><th>**99**</th></tr>**

**<tr><th>**이병은**</th><th>**85**</th><th>**90**</th></tr>**

**<tr><th>**김남윤**</th><th>**50**</th><th>**40**</th></tr>**

**</tbody>**

**</table>**

**</body>**

**</html>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**표 응용**</title>**

**<style>**

table {

border-collapse: collapse;

}

td, th {

text-align: left;

padding : 5px;

height: 15px;

width: 100px;

}

thead, tfoot {

background: darkgray;

color: yellow;

}

tbody tr:nth-child(even){

background: aliceblue;

}

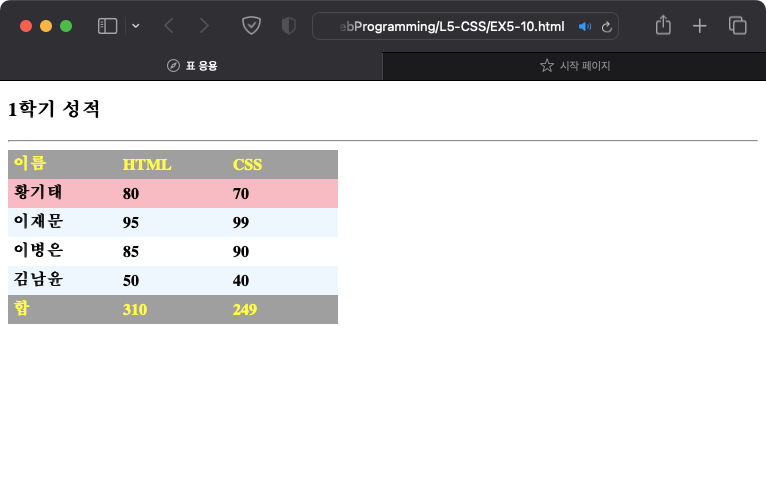
tbody tr:hover {

background: pink;

}

**</style>**

**</head>**

****

<그림 10-1> 예제 5-10 결과

CSS 스타일을 활용하여 표 테이블을 꾸밀 수 있다. 표 테두리를 설정할 때 테이블의 테두리와 <td>태그, <th>태그의 테두리도 설정할 수 있는데 모두 다 테두리를 설정하면 중복되어 겹치게 된다. 이럴 경우 ‘border-collapse: collapse;’를 사용하여 겹치는 테두리를 합칠 수 있다. ‘height’ 속성과 ‘width’ 속성을 사용하여 셀의 크기를 설정할 수 있으며 ‘padding’ 속성을 사용하여 여백을, ‘text-align’ 속성을 사용하여 정렬을 설정할 수 있다.

<그림 10-1>에서 ‘tbody tr:nth-child(even){…}’에 따라 짝수 행에 ‘aliceblue’ 색이 설정되었다. 만일 홀수 행에 지정하고 싶다면 ‘even’을 ‘odd’로 설정하면 된다. 끝으로 마우스를 올리면 해당 행이 분홍색으로 바뀌게 설정되었다.

1. **예제 5-11 스타일로 폼 꾸미기**

**<body>**

**<h3>**CONTACT US**</h3>**

**<hr>**

**<form>**

**<label>**

**<span>**Name**</span>**

**<input type="text" placeholder="Elvis">**

**</label>**

**<label>**

**<span>**Email**</span>**

**<input type="email"**

**placeholder="elvis@graceland.com">**

**</label>**

**<label>**

**<span>**Comment**</span>**

**<textarea placeholder="메시지를 남겨주세요">**

**</textarea>**

**</label>**

**<label>**

**<span></span>**

**<input type="submit" value="submit">**

**</label>**

**</form>**

**</body>**

**</html>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**폼 스타일 주기**</title>**

**<style>**

input[type=text] {

color: red;

}

input:hover, textarea:hover {

background: aliceblue;

}

input[type=text]:focus, input[type=email]:focus {

font-size: 120%;

}

label {

display: block;

padding: 5px;

}

label span {

display: inline-block;

width: 90px;

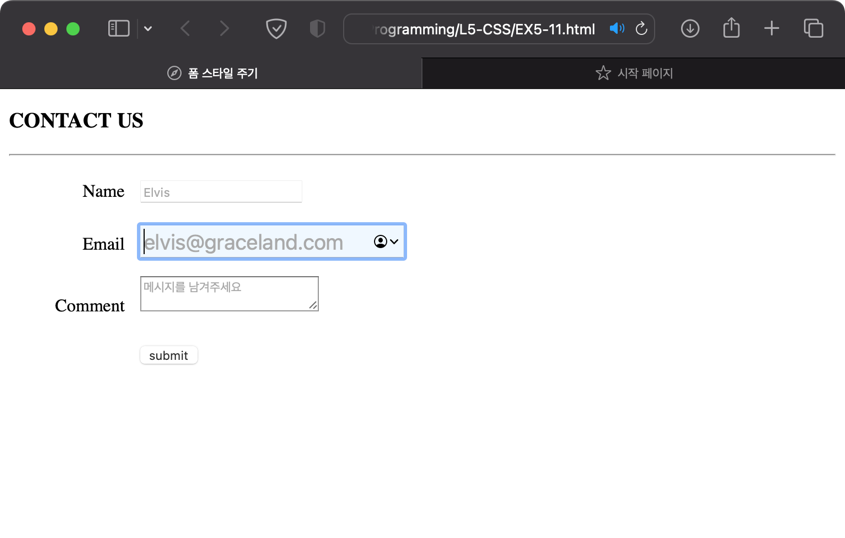
text-align: right;

padding : 10px

}

**</style>**

**</head>**

****

<그림 11-1> 예제 5-11 결과

CSS 스타일에서 ‘input[type=…]’을 사용하여 입력 창을 꾸밀 수 있다. ‘color’ 속성으로 입력되는 글자의 색을 지정할 수 있고 ‘border’ 속성 및 ‘border-radius’ 속성으로 테두리를 꾸밀 수 있다. 뿐만 아니라 가상 클래스 셀렉터를 사용하여 입력 창을 꾸밀 수 있다. <그림 11-1>에서 <input>태그의 ‘type’ 속성이 text로 되어있는 입력 창은 빨간색 글자로 입력된다. email로 되어있는 입력 창과 함께 포커스를 받을 때 글자 크기가 120%로 확대된다. 또한 모든 <input>태그와 <textarea>태그에 마우스를 올리면 입력 창이 ‘aliceblue’색으로 변경된다.

1. **예제 5-12 애니메이션 만들기 연습**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**애니메이션**</title>**

**<style>**

@keyframes bomb {

from { font-size: 500%; }

to { font-size: 100%; }

}

h3 {

animation-name: bomb;

animation-duration: 3s;

animation-iteration-count: infinite;

}

**</style>**

**</head>**

**<body>**

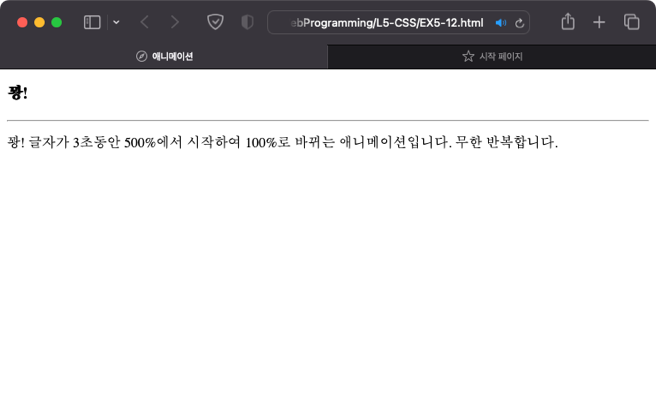
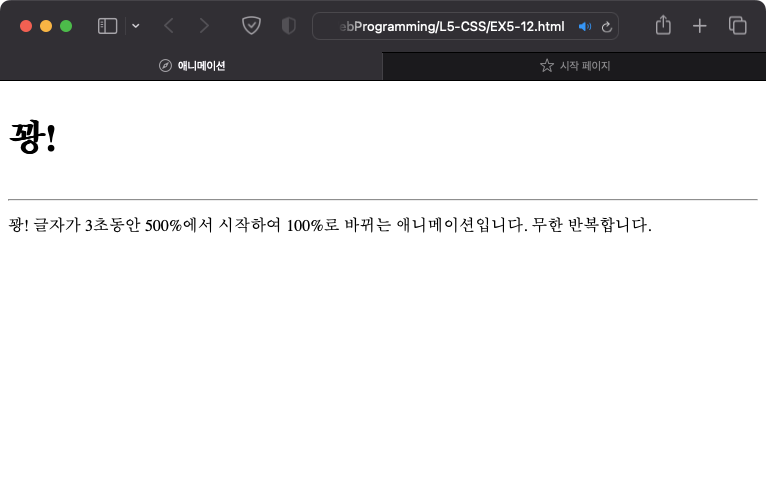
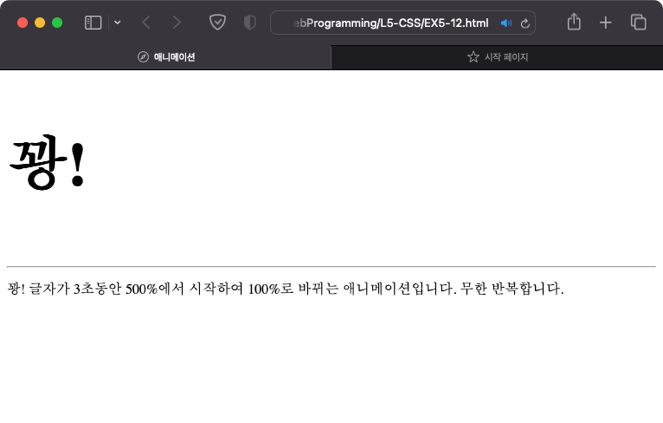
**<h3>**꽝!**</h3>**

**<hr>**

**<p>**꽝! 글자가 3초동안 500%에서 시작하여 100%로 바뀌는 애니메이션입니다. 무한 반복합니다.**</p>**

**</body>**

**</html>**

****

<그림 12-1> 예제 5-12 결과

CSS에서 애니메이션(animation)을 사용하여 텍스트를 꾸밀 수 있다. @keyframes를 사용하여 시간별 모양의 변화를 지정할 수 있다. 애니메이션 관련 속성은 ‘animation-name’, ‘animation-durations’, ‘animation-iteration-count’가 있다. 우선 ‘animation-name’ 속성은 애니메이션의 이름을 지정할 때 사용한다. ‘animation-durations’ 속성은 애니메이션의 1회 시간을 지정한다. ‘animation-iteration-count’ 속성은 반복 횟수를 지정한다. infinite로 설정하면 무한 반복을 한다. <그림 12-1>에서 ‘꽝!’이라는 글자는 ‘bomb’이라는 애니메이션의 이름으로 3초 동안 500%에서 100%로 크기가 작아지며 무한 반복한다.

1. **예제 5-13 font-size에 대한 전환 효과 만들기**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**전환**</title>**

**<style>**

span {

transition: font-size 5s;

}

span:hover {

font-size: 500%;

}

**</style>**

**</head>**

**<body>**

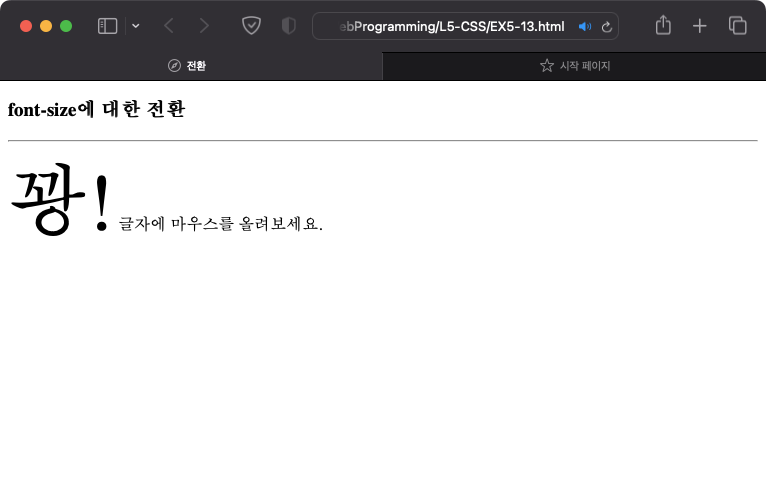
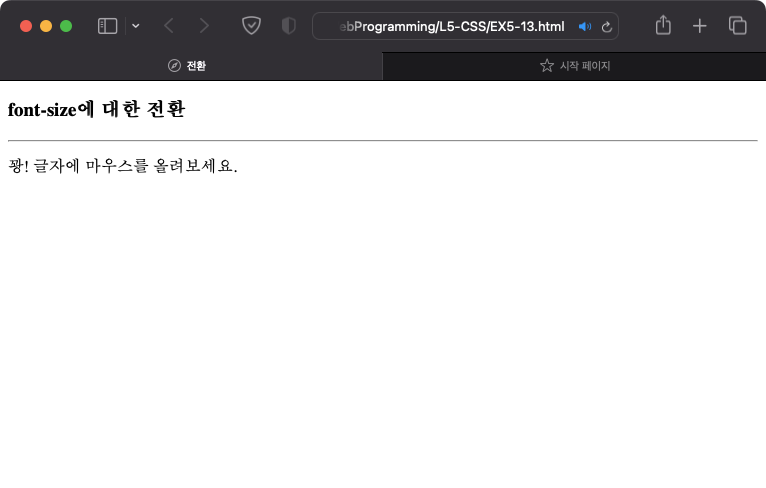
**<h3>**font-size에 대한 전환**</h3>**

**<hr>**

**<p><span>**꽝!**</span>** 글자에 마우스를 올려보세요.**</p>**

**</body>**

**</html>**

****

<그림 13-1> 예제 5-13 결과

CSS에서 전환(transition) 효과를 사용하여 텍스트를 꾸밀 수 있다. 전환이란 HTML에 적용된 CSS의 속성을 서서히 변화시키는 애니메이션 효과를 의미한다. <그림 13-1>에서 <span>태그의 ‘transition’의 값에서 전환 속성을 font-size로 하여 글자 크기를 변화할 것을 암시하였고 5s로 하여 5초에 걸처 서서히 변화하도록 지시하였다. 그리고 ‘hover’ 클래스를 사용하여 마우스를 올리면 500% 크기로 커지게 설정하였다.

1. **예제 5-14 다양한 변환 사례**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>**다양한 변환 사례**</title>**

**<style>**

div {

display : inline-block;

padding : 5px;

color : white;

background : olivedrab;

}

div#rotate { transform: rotate(20deg); }

div#skew { transform: skew(0deg,-20deg); }

div#translate { transform: translateY(100px); }

div#scale { transform: scale(3,1); }

div#rotate:hover { transform: rotate(80deg); }

div#skew:hover { transform: skew(0deg,-60deg); }

div#translate:hover { transform: translate(50px, 100px); }

div#scale:hover { transform: scale(4,2); }

div#scale:active { transform: scale(1,5); }

**</style>**

**</head>**

**<body>**

**<h3>**다양한 Transform**</h3>**

아래는 회전(rotate), 기울림(skew), 이동(translate), 확대/축소(scale)가 적용된 사례이다.

또한 마우스를 올리면 추가적 변환이 일어난다.

**<hr>**

**<div id="rotate">**rotate(20deg)**</div>**

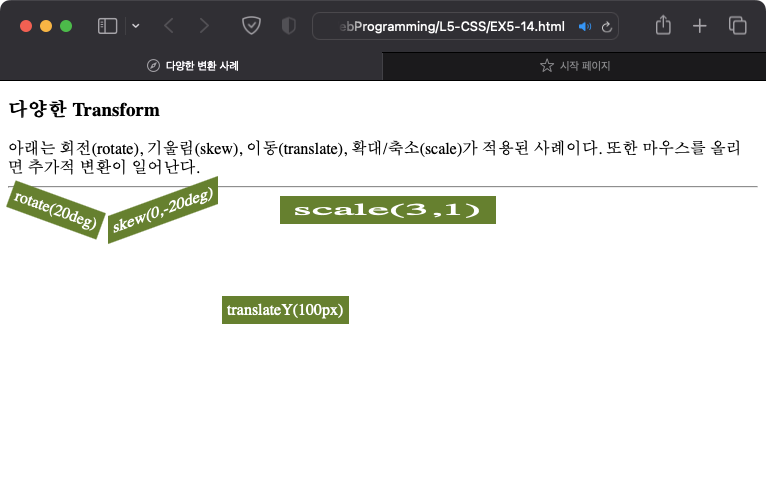
**<div id="skew">**skew(0,-20deg)**</div>**

**<div id="translate">**translateY(100px)**</div>**

**<div id="scale">**scale(3,1)**</div>**

**</body>**

**</html>**

****

<그림 14-1> 예제 5-14 결과

CSS에서 변환(transform) 효과를 사용하여 텍스트나 이미지를 꾸밀 수 있다. <그림 14-1>에서 사용된2차원 변환 함수로 위치 이동(translate), 확대/축소(scale), 회전(rotate), 기울임(skew)이 있다. <그림 14-1>에서 ‘rotate(20deg)’는 처음에 20만큼 회전되었고 마우스를 올리면 80만큼 회전한다. ‘skew(0,-20deg)’는 처음에 -20만큼 기울여져 있고 마우스를 올리면 -80deg만큼 기운다.

‘translateY(100px)’는 처음에 Y축으로 100만큼 이동하였고 마우스를 올리면 X축으로 50px만큼 더 이동한다. 끝으로 ‘scale(3,1)은 처음에 x축으로 3만큼, y축으로 1만큼 크기가 늘어져 있고 마우스를 올리면 x축으로 4만큼, y축으로 2만큼으로 바뀐다. 그리고 마우스 클릭을 하면 x축으로 1만큼, y축으로 5만큼 바뀌게 된다.

1. **참고문헌**

* W3schools, (n.d.), “CSS Tutorial”, <https://www.w3schools.com/css/default.asp>
* TCPSCHOOL, (n.d.), “CSS”, <http://www.tcpschool.com/css/intro>