**CSS**

CSS(Cascading Style Sheet)는 HTML과 같은 마크업 언어로 작성된 문서에 색상, 폰트, 각 요소의 위치 변경, 백그라운드 및 간단한 애니메이션(transition) 등의 효과를 주어 보다 보기 좋게 꾸며 주는 역할을 합니다.

**1. HTML에서 CSS를 적용하는 방법**

**1.1 HTML 요소 속성으로써 적용하는 방법**

HTML의 각 태그에서 style 속성으로 적용하는 방법이 있습니다.

<p style=”color: red”>빨간색</p>

속성으로써 적용을 하게 되면 적용의 우선 순위가 가장 높으므로 바로 적용됩니다.

다만 우선순위가 가장 높고, HTML의 요소 안에 분산되어 산재하므로 유지보수에 어려움이

있을 수 있습니다.

CSS 적용 순서

요소에서 style 속성(inline 타입) > #id(클래스명) > .class(클래스명) > tag(태그명)

**1.2 HTML <style>~</style> 요소로 적용하는 방법**

같은 HTML 문서에 <style> 태그 안에 스타일을 적용하는 방식 입니다.

예)

<!DOCTYPE html>

<html>

<head>

<meta charset='utf-8'>

<style>

p { color : red; }

</style>

</head>

<body>

<p>빨간색 글씨</p>

</body>

</html>

같은 문서에 CSS가 HTML 요소로 함께 존재하게 되는 형태입니다. 여러페이지에 공통 스타일이 있는 경우 중복해서 작성해야 하는 불편함이 있고, 중복된 소스가 분산되어 산재하므로

유지 보수에 어려움이 따를 수 있습니다.

**1.3 외부 파일로 적용하는 방법**

CSS를 외부 파일로 따로 분리하여 작성하는 방식 입니다.

<link rel=”stylesheet” type=”text/css” href='css 외부 파일 경로'>

예)

<!DOCTYPE html>

<html>

<head>

<meta charset='utf-8'>

<link rel=”stylesheet” type=”text/css” href=”css/style/.css”>

</head>

<body>

<p>빨간색 글씨</p>

</body>

</html>

외부파일로 따로 분리하여 작성하는 경우 공통 요소에 대한 스타일을 한번만 정의 할수 있습니다.

웹브라우저는 한번 다운로드 받은 외부 파일은 캐싱처리를 하므로 브라우저의 렌더링 속도에 이점이 있습니다.

그러나 캐싱처리를 하므로 수정한 CSS 요소가 바로 반영되지 않는 다는 문제가 있습니다.

**2. 선택자**

HTML요소에 스타일을 입힐려면 우선 특정 요소를 선택하고 선택한 요소에 스타일을 지정하여 꾸며야 합니다.

특정 요소를 선택하게 해 주는 것을 선택자라고 합니다.

**2.1 선택자의 종류**

**2.1.1 태그 선택자**

div, span, p와 같이 태그 이름으로 요소를 선택하는 방법 입니다.

태그 이름이므로 복수의 요소를 선택할 수 있습니다.

<style>

p { color: red; }

</style>

**2.1.2 클래스 선택자**

클래스 속성에 지정된 이름으로 요소를 선택하는 방법입니다.

클래스는 개념적으로 복수의 요소에 적용하기 위한 속성이므로 복수의 요소를 선택할 수 있습니다.

클래스 선택자는 점(.)으로 시작하며 클래스명을 입력합니다.

<style>

.selected { color: red }

</style>

<ul>

<li>스파크

<li class='selected'>모닝</li>

<li>레이</li>

</ul>

**2.1.3 아이디 선택자**

아이디 속성으로 지정된 이름으로 요소를 선택하는 방법입니다.

아이디는 개념적으로 1개의 요소에 적용하기 위한 속성입니다. 그러나 CSS에서는 여러개가 선택될 수 있으나 적용 우선순위에 영향을 받으므로 반드시 1개의 요소만 적용하도록 합니다.

아이디 선택자는 샵(#)으로 시작하며 아이디명을 입력합니다.

<style>

#title { color: red }

</style>

<h1 id='title'>제목</h1>

**2.1.4 스타일 적용 우선 순위**

style 속성에 적용(inline 타입) > #id명 > #class명 > tag이름

스타일은 가장 많은 요소에 적용될수 있는 범위의 선택자가 가장 우선순위가 작고, 적용 범위가 작을 수록 스타일 우선순위가 높습니다.

**2.1.5. 조상 자손 선택자**

스타일을 적용할 요소의 범위를 좁혀서 적용하기를 원할때 사용할 수 있습니다.

왼쪽에 먼저 나열되는 요소가 상위 요소로 인식됩니다.

예)

p 태그 아래 li 요소가 모두 선택됩니다.

<style>

p li { color :red; }

</style>

<p>

<ul>

<li>스파크</li>

<li>모닝</li>

<li>레이</li>

</ul>

<ol>

<li>스파크</li>

<li>모닝</li>

<li>레이</li>

</ol>

</p>

**2.1.6 부모 자식 선택자**

상기 예시의 경우 p 태그 아래 li로 되어 있는 모든 태그가 적용이 됩니다. 그러나 ul 요소 하위 li만 적용해야 하는 경우도 있는데, 그때는 > 결합자 를 사용하여 바로 하위 요소를 선택할 수 있습니다.

<style>

p ul > li { color: red; }

</style>

**2.1.7 그룹 선택자**

여러 요소를 선택하여 동일 속성을 선택하는 방법

<style>

ul, ol { color: red; }

.car, .truck { color: blue; }

#nav li, #footer li { font-size: 20px; }

</style>

**2.1.8 가상 클래스선택자**

선택자에 추가하는 키워드로, 선택한 요소가 특별한 상태여야 적용될 수 있는 선택자를 의미 합니다.

예)

:link - 방문한 적이 없는 링크

:visited - 방문한 적이 있는 링크

:hover - 마우스를 롤오버 했을 때

:active - 마우스를 클릭했을 때

:read-only 읽기 전용 상태일때

:not(선택자) { 특정 포함되지 않은 요소만 선택)

:nth-child 특정 순서에 있는 요소를 선택할 때

:checked radio, checkbox에서 선택된 요소

:after - 하위 요소 가장 끝에

:before - 하위 요소 바로 앞에

선택자를 모두 암기는 불가 다만 쉽게 찾아서 쓸 수 있도록 하기 참고 URL을 한번씩 연습 해 볼 것

선택자 Reference

<https://developer.mozilla.org/ko/docs/Web/CSS/Reference#%EC%84%A0%ED%83%9D%EC%9E%90>

**2.1.9 결합자**

결합자니는 “A는 B의 자식”, “A는 B와 인접한 요소” 처럼 두 개 이상의 선택자 끼지 관계를 형성합니다.

**2.1.9.1 인접 형제 결합 A + B**

요소 A와 B가 같은 부모를 가지며 B가 A를 바로 뒤따라야 하도록 지정합니다.

**2.1.9.2 일반 형제 결합 A ~ B**

요소 A와 B가 같은 무모를 가지며, B가 A를 뒤따라야 하도록 지정합니다. 그러나 B가 A의 바로 옆에 위치해야 할 필요는 없습니다.

**2.1.9.3 자식 결합자 A > B**

요소 B가 A의 바로 밑에 위치해야 하도록 지정합니다.

**2.1.9.4 자손 결합자**

요소 B가 A의 밑에 위치해야 하도록 지정합니다. 그러나 B가 A의 바로 아래에 있을 필요는 없습니다.

**2.2 스타일 상속(inherit)**

스타일은 가장 효율적인 방식으로 브라우저에서 적용이 됩니다. 보통 상위 요소가 하위 요소의 스타일에 영향을 주나 모든 속성에 해당되지는 않습니다.

예)

color 속성은 상속됩니다.

border 속성은 상속되지 않습니다.

**3. 속성**

스타일 각각의 효과는 속성이라고 합니다. 속성은 약 250개 정도가 있다고 하나 실제 개발 환경에서 모두 익히고 작업을 하는 경우는 거의 없습니다. 사용 빈도수가 낮은 경우는 바로 떠올려서 적용하기 역시 어려울 수 있습니다.

따라서 속성에 대한 공부는 가장 많이 쓰는 속성 위주로 연습을 하시면 됩니다.

**3.1 폰트**

**3.1.1 font-size**

글자 크기를 지정하는 속성 입니다. 주요 단위는 px, em, rem입니다.

rem

<html> 태그에 적용된 font-size에 따라 상대적으로 크기가 결정됩니다.

px

모니터상의 화소 하나의 크기에 대응되는 단위 입니다. 고정된 값이기 때문에 이해하기

쉽습니다.

em

부모태그에 지정된 font-size에 따라 상대적으로 크기가 결정됩니다.

**3.1.2 color**

글꼴의 컬러를 지정할 수 있습니다.

색상을 지정하는 방법

hex 코드(16진수 코드) 적용방식

예) p { color: #ff0000; }

색상명으로 적용하는 방식

예) p { color: red; }

rgb방식으로 적용하는 방식

빛의 3원색인 빨강,녹색,파랑의 수치로 적용하는 방법이며

색상의 범위는 각각 0~255(256개)씩 조합하여 색상을 구성합니다.

(16,777,216개 색상)

p { color: rgb(255, 0, 0); }

**3.1.3 text-align**

텍스트 정렬 방향을 지정할 수 있습니다.

left - 왼쪽 정렬

right - 오른쪽 정렬

center - 중앙 정렬

justify - 양쪽 정렬

**3.1.4 line-height**

행간 높이를 지정할 수 있습니다. 기본값은 1.2 입니다.

**3.1.5. font-weight**

텍스트의 굵기를 지정할수 있습니다.

normal(정상), bold(굵게)와 같이 텍스트로 속성을 지정하거나

100~900 범위의 숫자로 굵기를 지정할 수 있습니다.

(다만 폰트가 지원을 하는 숫자의 굵이여야 적용이 됩니다.)

**3.1.6 font-family**

글꼴을 지정할 수 있는 속성입니다.

글꼴을 지정하는 방법

p { font-family: 폰트명1, 폰트명2, 폰트명3 }

폰트는 적용가능 폰트를 왼쪽부터 우선 순위를 가지고 적용됩니다.

즉 폰트명1이 적용된다면 폰트명2는 적용되지 않습니다.

2개 이상 단어로 구성된 폰트명은 큰따옴표(”)로 감싸서 설정합니다.

예) p { font-family : “Sans Serif”, Verdana, “Times New Roman”; }

폰트는 사이트를 이용하는 사용자의 컴퓨터에 자체적으로 보유하고 있는 시스템 폰트와 웹폰트로 구분해서 생각할 수 있습니다.

시스템에 설치하는 폰트는 웹 사용자에 따라 보유하고 있을 수도 있고 없을 수도 있으므로 통일성을 위해서 또는 라이센스가 있는 보기 좋은 폰트 사용을 위해 웹 폰트를 사용하기도 합니다.

**웹폰트 사용하는법**

<https://fonts.google.com/>

예)

<https://fonts.google.com/specimen/Noto+Sans+KR?preview.text_type=custom&selection.family=Noto+Sans+KR:wght@100;300;400;500;700;900&sidebar.open=true>

<link>방식

<link rel="preconnect" href="https://fonts.gstatic.com">  
 <link href="https://fonts.googleapis.com/css2? family= **Noto+Sans+KR:wght@100;300;400;500;700;900 &** display=swap" rel="stylesheet">

import 방식

@import url('https://fonts.googleapis.com/css2? family= **Noto+Sans+KR:wght@100;300;400;500;700;900 &** display=swap');

font-family: 'Noto Sans KR', sans-serif

**3.2 공간**

**3.2.1 인라인 레벨 요소(Inline level Element)**

줄개행을 하지 않습니다.

공간을 지정할 수 없습니다. 요소 안에 있는 내용만큼의 공간만 차지합니다.

위 아래 공백(margin)을 지정할 수 없으나, 내부 공백(padding)은 지정할 수 있습니다.

대표적으로 <span>태그는 inline-level 요소 입니다.

**3.2.2 블록 레벨 요소(block Level Element)**

항상 줄개행을 합니다.

공간을 지정할 수 있습니다. 즉, width, height(너비와 높이)를 가질 수 있습니다.(CSS에서 지정)

아래 위 또는 왼쪽 오른쪽에 공백(margin)을 지정할 수 있습니다.

대표적으로 <div> 태그는 block-level 요소 입니다.

**3.2.3 인라인 블록 요소(Inline-Block Level Element)**

block-level, inline-level 외에도 이 둘의 속성을 모두 가지고 있는 inline-block-level 요소도 있습니다.

각 요소 자체에 자연적으로 있는 속성은 아니며, style 지정을 하여 적용할 수 있습니다.

또한 block-level 속성은 style 지정을 통해 inline, inline-block level 속성으로 변경이 가능합니다.

inline-level 속성 역시 style 지정을 통해 inline-block, block-level 속성으로 변경할 수 있습니다.

**3.2.4 display 속성**

diplay 속성을 사용하여 block, inline, inline-block, 또는 none 속성(안보임처리)를 지정하여

공간 속성을 변경할 수 있습니다.

예)

<style>

p { display: inline; }

span { display: block; }

.section { display: none; }

</style>

**3.3 레이아웃**

**3.3.1 box-sizing**

box-sizing 속성을 설정하지 않는 다면 모든 요소는 기본 content-box 속성을 가집니다.

content-box

내용 기준이 되며, 각 요소의 기준 너비, 높이는 보더(border)와 padding이 더해집니다.

border-box

보더가 기준이 되며 보더 기준으로 기준 너비, 높이가 결정됩니다.

**3.3.2 포지션**

position 속성은 문서상의 배치하는 방법을 지정합니다.(top, right, bottom, left, z-index)

아무 속성을 지정하지 않는다면 기본 값은 static 입니다.

**static**

요소를 일반적인 문서 흐름에 따라 배치합니다. top, right, bottom, left, z-index 속성은 적용되지 않습니다.

**relative**

요소를 일반적인 문서 흐름에 따라 배치하고 자기자신을 기준으로 top, right, bottom, left의 값에 따라 오프셋(offset)을 적용합니다.

z-index에 따라 요소의 층위를 지정할 수 있습니다.

**absolute**

요소를 일반적인 문서 흐름에서 제거하고, 페이지 레이아웃 공간도 배정하지 않습니다.

대신 가장 가까운 위치 지정 조상요소에 대해 상대적으로으로 배치합니다(조상 위치 기준으로 top, right, bottom, left 값 지정)단, 조상 중 위치 지정요소가 없다면 가장 상위 블록을 기준으로 삼습니다.

조상요소를 지정하는 방법은 상위 요소에 position: relative; 속성을 부여 하면 됩니다.

z-index에 따라 요소의 층위를 지정할 수 있습니다.

**fixed**

요소를 일반적인 문서 흐름에서 제거하고 페이지 레이아웃에 공간도 배정하지 않습니다.

대신 뷰포트의 초기 컨테이닝 블록을 기준으로 삼아 배치합니다(즉, 브라우저에서 보이는 영역 기준으로 top, right, bottom, left 배치)

**3.3.3 float**

왼쪽 또는 오른쪽 방향에 따라 흘러가듯이 배치

left - 왼쪽 방향으로 흘러가듯이 배치

right - 오른쪽 방향으로 흘러가듯이 배치

none - 초기값이며 흘러가는듯한 배치를 하지 않음

float 속성은 clear를 해주지 않는다면 지정하지 않아도 다음 요소에 영향을 줄 있으므로 반드시 clear 처리합니다.

예)

<style>

ul.menu > li { float: left; }

ul.menu:after { clear: left; content: ''; display: block; }

</style>

<ul class='menu'>

<li>메뉴1</li>

<li>메뉴2</li>

<li>메뉴3</li>

</ul>

**3.3.4 margin**

margin 속성은 네 방향 바깥 여백 영역을 설정 합니다.

margin-top, margin-right, margin-bottom, margin-left의 단축 속성 입니다.

적용예시)

/\* 네 면 모두 적용 \*/

margin: 10px;

/\* 세로방향 | 가로 방향 \*/

margin: 10px 20px;

/\* 위 | 가로방향 | 아래 \*/

margin: 10px 20px 5px;

/\* 위 | 오른쪽 | 아래 | 왼쪽 \*/

margin: 10px 5px 15px 6px;

3.3.5 padding

요소 내부의 빈 공간을 추가합니다.

padding-top, padding-right, padding-bottom, padding-left의 단축 속성 입니다.

적용예시)

/\* 네 면 모두 적용 \*/

padding: 10px;

/\* 세로방향 | 가로 방향 \*/

padding: 10px 20px;

/\* 위 | 가로방향 | 아래 \*/

padding: 10px 20px 5px;

/\* 위 | 오른쪽 | 아래 | 왼쪽 \*/

padding: 10px 5px 15px 6px;

**3.3.6 다단(multi column)**

신문과 같이 긴 텍스트를 단을 나누어 보기 좋게 출력 할 수 있습니다.

column-count 다단 갯수

column-width - 다단별 너비

column-gap - 다단 사이의 여백

column-rule-width - 구분선 두께

column-rule-style - dotted(점선)|solid(직선)|thick(두꺼운 직선) - 다단에 구분선을 넣는 경우

column-rule-color - 구분선 색상

**3.3.7 media query**

미디어쿼리는 다양한 장비(미디에)에 따른 화면 사이즈에 적응하기 위해 특정 break-point 기준에 따라 CSS 속성을 다르게 적용하는 방법 입니다.

예)

<style>

@media all and (max-width: 400px) { // 화면사이즈 400px이하 적용

};

@media all and (max-width: 720px) { // 화면사이즈 720px 이하 적용

};

@media all and (max-width: 1024px) { // 화면사이즈 1024px 이하 적용

}

</style>

**3.4 그래픽**

**3.4.1 background**

배경색 또는 배경 이미지를 지정하는 속성 입니다.

**3.4.1.1 background-color**

요소의 배경색상을 지정합니다.

예)

body { background-color: blue; }

**3.4.1.2 background-image**

배경을 이미지로 채우게 됩니다. 이미지 경로를 설정하면 좌우, 상하 반복(repeat-x, repeat-y) 속성이 기본적으로 적용되므로 요소를 가득 채우게 됩니다.

예)

body { background-image: url(”img/photo.jpg”); }

**3.4.1.3 background-repeat**

배경이미지의 반복 속성을 지정합니다.

no-repeat 반복 없음

repeat-x - 좌우 방향으로 반복

repeat-y - 상하 방향으로 반복

예)

body {

background-image : url(”img/photo.jpg”);

background-repeat : no-repeat;

}

3.4.1.4 background-attachment

배경이미지를 스크롤할지 고정할지 여부를 지정

fixed - 고정

scroll; 스크롤

예)

body {

background-image : url(”img/photo.jpg”);

background-repeat : no-repeat;

background-attachment : scroll;

}

**3.4.1.5 background-position**

배경이지미의 위치를 지정할 수 있습니다.

background-position: 좌중우(left|right|center) 상중하(top|bottom|center);

background-position: 100px 100px; (좌측에서 100px 이동, 위에서 아래로 100px 이동)

예)

body {

background-image : url(”img/photo.jpg”);

background-repeat : no-repeat;

background-position: right top;

}

**3.4.1.6 background 단축형**

background: 색상 이미지 반복여부 스크롤여부 위치

예)

body {

background: #ffffff url(”img/photo.jpg”) no-repeat right top;

}

**3.4.2 overflow**

요소 안에 있는 컨텐츠의 크기가 영역에 비해 클 경우 통제하는 속성

**visible** - 기본속성이며, 컨텐츠 영역이 상위 영역에 비해 클 경우 영역 밖에 겹쳐지게 출력이 됩니다.

**hidden** - 컨텐츠 영역중 넘어서는영역을 감춤니다.

**scroll** - 컨텐츠 영역이 넘어설 경우 스크롤바 생성

**auto** - 컨텐츠 영역이 영역을 넘어서지 않으면 아것도 발생하지 않으나 넘어설 경우 스크롤바를 생성

**overflow-y**: auto - 상하 위치 기준으로 스크롤바 생성

**overflow-x**: auto - 좌우 위치 기준으로 스크롤바 생성

**3.4.3 transition**

CSS 효과를 특정 지연을 주어 부드럽게 전환될 수 있도록 합니다.

transition-delay - CSS 속성이 적용되기전 지연시간을 지정합니다.

예) transition-delay : 1s;

transition-duration

전환효과 진행시간

transition-property - CSS 속성

transition-property : width; - 가로 너비가 변할 경우 전환효과 발생

transition-timing-function - 애니메이션 효과

(linear, ease, ease-in, ease-out, ease-in-out)

transition 축약 적용

transition : property(속성) duration(지연시간), timing-function(애니메이션효과), delay(전환발생전 대기시간)

예)

div {

transition: width 2s ease-out 1s;

}