

웹 프로그래밍

- CSS -

CSS 등장 배경

- 초기의 HTML은 정보를 표현 하기 위해 나옴
- 외양을 코딩하는 방법이 HTML에 추가 됨
- 콘텐츠가 복잡해짐
- HTML이 정보를 담는데 전념하도록 새로운 언어 도입
- HTML 4.0부터 CSS(Cascading Style Sheet) 도입

CSS 문법

선택자
↓
a { background-color: yellow; font-size: 16px; }
↑
선언 시작

속성명
↑
선언 구분자

속성값
↑
선언 끝

CSS를 삽입하는 방법

- 외부 스타일 시트(external style sheet)
`<link type="text/css" rel="stylesheet" href="external_style.css">`
- 내부 스타일 시트(internal style sheet)
`<style> p { background-color: yellow; } </style>`
- 인라인 스타일 시트(inline style sheet)
`<h1 style="color : red"> head </h1>`

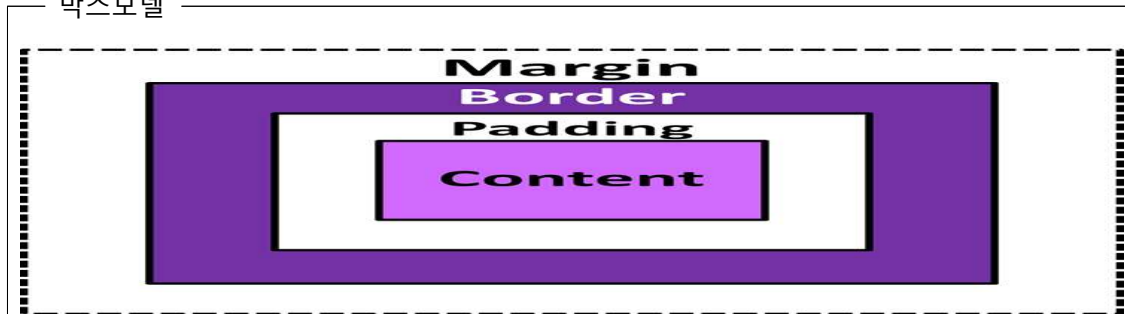
선택자

- 타입 선택자(type selector) : HTML 요소 이름을 사용하는 것, `h1 { color : green; }`
- 전체 선택자(universal selector) : 페이지 안 모든 요소를 선택, `* { color : blue; }`
- 클래스 선택자(class selector) : `.prog { color : red; }, h1.prog { color : blue; }`
- 아이디 선택자(ID selector) : `#web { color : red; }`
- 속성 선택자(attribute selector) : 특정한 속성을 가지는 요소를 선택
title 속성을 가지고 있는 h1 요소 -> `h1[title] { color : blue; }`
- 의사 클래스(pseudo-class) : `a:hover`라고 하면 `<a>` 요소에 클래스 `hover`가 선언 된 것처럼 생각하고 선택자를 만드는 것
- 선택자 그룹 : `h1, h2, h3 { color : red; }`
- 자손 : `div em { color : red; }`
- 자식 : `div>em { color : blue; }`

속성

- 폰트 : 선택자 { font : italic 30px Georgia,serif; }
- 색상 : 선택자 { color : 색상; background-color : 색상; }
- 텍스트스타일 : letter-spacing : 글자 간 간격, line-height : 줄의 높이,
text-align : 수평 정렬을 지정(center, right, justify etc...)
text-shadow : 그림자 효과 { text-shadow : 5px 5px 5px #0f0f0f ; }
word-wrap : break-word; 자동으로 줄 바꿈
- 다단 : { column-count : 2; column-gap : 50px; column-rule : 3px outset black; }

박스모델



박스의 너비

- width와 height 속성으로 콘텐츠 영역의 크기가 설정
- 전체 크기를 계산하려면? → 패딩, 경계, 마진을 더해야한다.

ex) p {

```
width : 150px;  
padding : 10px;  
border : 5px solid red;  
margin : 20px;
```

}

총 너비는 220px

인라인 요소, 블록 요소

- 블록요소 : <h1>, <p>, , , <table>, <pre>, <div>, <form>, <header>, <nav>...
- 인라인요소 : <a>, , , ,
, <input>, ...
- 가운데 정렬하기
`<p style="width : 20%; text-align:center; margin-left : auto; margin-right : auto;">`

테이블

- 자주 사용하는 속성
table { border-collapse: collapse; } table, th, td { border: 2px solid blue; }
- 캡션 위치 caption-side : top or bottom or inherit
- 빈 셀 처리 empty-cells : show or hide or inherit

요소 위치 정하기

- top, bottom, left, right 속성으로 지정 ex) #target { top : 100px; left : 200px; }
- 기준위치인 컨테이너의 경계에서 떨어진 거리로 나타남
- 기준위치는 position 속성으로 지정

position 속성

- 정적 위치 설정(static positioning) : 고정, top,bottom,left,right 속성 영향X
- 상대 위치 설정(relative positioning) : 일반적인 기준위치에서
- 절대 위치 설정(absolute positioning) : 전체 페이지를 기준으로
- 고정 위치 설정(fixed positioning) : 스크롤 내려도 같은 위치에 고정

기타 속성

- display : inline 블록요소를 가로로 배치하기 위해 인라인 요소인 것처럼, 레이아웃을 작성할 때 유용
- z-index : 숫자 숫자 값이 클수록 앞쪽에 위치하게 됨.

레이아웃

- float 속성
#header { background-color: yellow; width: 100%; height: 50px; }
#nav { background-color: red; width: 30%; height: 100px; float: left; }
#content { background-color: blue; width: 70%; height: 100px; float: right; }
#footer { background-color: aqua; width: 100%; height: 50px; clear: both; }
- display : table-cell
header{ background-color: #e3afed; }
nav{ display:table-cell; background-color:#ffd800; }
section{ display:table-cell; background-color:#ffff00; }
aside{ display:table-cell; background-color:#674989; }
footer{ background-color:#954b4b; }

간단한 효과

- 투명도 :

```
img { opacity: 0.4; }  
img:hover { opacity: 1.0; }
```
- 전환 :

```
div { width: 100px; border: 2px solid black; background: lightblue;  
      transition: width 1s; }  
div:hover { width: 200px; }
```
- 변환 :

```
div { width: 50px; height: 50px; border: 1px solid black; margin: 20px; }  
div#box1 { transform: rotate(30deg); background-color: green; }  
div#box2 { transform: skew(30deg, 20deg); background-color: blue; }
```
- 애니메이션 :

```
div { width: 100px; height: 100px;  
      background: red; position: relative;  
      animation: 2s movebox;  
      animation-iteration-count: 5; }
```



```
@keyframes movebox  
{ 0% {left:0px; top:0px; }  
  50% {left:200px; top:0px; }  
 100% {left:0px; top:0px; }
```

메 모