

스타일과 스타일 시트

스타일과 스타일 시트

- 스타일(style) : HTML 문서에서 자주 사용하는 글꼴이나 색상, 정렬, 각 요소들의 배치 방법 등 문서의 겉모습을 결정짓는 내용들
- 스타일 시트(style sheet) : 스타일을 관리하기 쉽도록 한 군데 모아놓은 것

왜 스타일을 사용할까?

- 웹 문서의 내용과 상관없이 디자인만 바꿀 수 있다.
 - 내용과 디자인의 분리 – 웹 표준의 시작
 - 디자인에 영향 없이 내용 수정하거나 내용은 건드리지 않고 디자인만 바꾸는게 가능
- 다양한 기기에 맞춰 탄력적으로 바뀌는 문서를 만들 수 있다.
 - 내용은 그대로 두고, 프린터나 스마트폰 브라우저 등 다양한 기기에 맞는 레이아웃을 만들 수 있다.

스타일과 스타일 시트

스타일 형식



- 선택자 : { 와 } 사이에 정의한 스타일 규칙이 적용될 대상.
- 속성과 속성 값 : '속성 : 속성 값'과 같은 형식으로 함께 표시하며, 속성 /속성 값 쌍이 여럿일 경우에 세미콜론(;)으로 구분

```
p { color: blue; font-size: 16px; } /* 텍스트 단락의 글자 색 파란색, 글자 크기 16px */
```

```
h2 {font-size:20px; color:orange; } /* 2단계 제목의 글자 크기 20px, 글자 색상 주황색 */  
p {color:blue; } /* 텍스트 단락의 글자 색 파란색 */
```

스타일을 표기하는 방법

```
h2{font-size:20px; color:orange;}
```

```
h2{  
  font-size:20px;  
  color:orange;  
}
```

```
h2  
{  
  font-size:20px;  
  color:orange;  
}
```

모두 가능

스타일 주석

- /*와 */ 사이에 주석 내용 입력
- 한 줄 또는 여러 줄을 입력 가능

스타일과 스타일 시트

내부 스타일 시트

- 사용할 스타일을 문서 안에 정리한 것
- 모든 스타일 정보는
 <head> 태그와 </head> 태그 안에서 정의
- <style> 태그와 </style> 태그 사이에 작성

Do it! 내부 스타일 시트 사용하기 예제 05/internal.html

```
<!doctype html>
<html lang="ko">
<head>
  <title>온라인 프로필</title>
  <meta charset="utf-8">
  <style>
    section {
      width:500px; /* 너비 */
      padding:15px; /* 테두리와 내용 사이의 여백 */
      border:5px solid gray; /* 테두리 스타일 */
    }
  </style>
</head>
<body>
  <div id="container">
    <!-- 자기 소개 -->
    <section>
      <h2>Who am I?</h2>
      <p>프론트엔드 웹 기술(Front-end Web Tech.)에 관심이 많습니다. <br>
      현재 제주의 한 시골 마을에서 코딩 중입니다.</p>
    </section>
  </div>
</body>
</html>
```

Who am I?

프론트엔드 웹 기술(Front-end Web Tech.)에 관심이 많습니다.
현재 제주의 한 시골 마을에서 코딩 중입니다.

section {
width:500px; /* 너비 */
padding:15px; /* 테두리와 내용 사이의 여백 */
border:5px solid gray; /* 테두리 스타일 */
}


<h2>Who am I?</h2>
<p>프론트엔드 웹 기술(Front-end Web Tech.)에 관심이 많습니다.

현재 제주의 한 시골 마을에서 코딩 중입니다.</p>

내부 스타일 시트

스타일 시트에 있는 section 스타일
규칙이 적용된 부분

각 스타일 속성은
앞으로 배울
거예요!



스타일과 스타일 시트

외부 스타일 시트

- 여러 웹 문서에서 사용할 스타일을 별도 파일로 저장해 놓고 필요할 때마다 파일에서 가져와 사용
- <style> 태그 없이 <link> 태그만 사용해 미리 만들어 놓은 외부 스타일 시트 파일 연결

style.css

```
h2{
  color:blue; /* 글자 색 */
}
p {
  font-size:0.9em; /* 글자 크기 */
  line-height:2.0; /* 줄간격 */
}
```

```
<head>
  <title>온라인 프로필</title>
  <meta charset="utf-8">
  <style>
    section {
      width:500px; /* 너비 */
      padding:15px; /* 테두리와 내용 사이의 여백 */
      border:5px solid gray; /* 테두리 스타일 */
    }
  </style>
  <link rel="stylesheet" href="css/style.css">
</head>
```

Who am I?

프론트엔드 웹 기술(Front-end Web Tech.)에 관심이 많습니다.
현재 제주의 한 시골 마을에서 코딩 중입니다.

스타일과 스타일 시트

인라인 스타일

- 스타일 시트를 사용하지 않고 스타일을 적용할 대상에 직접 표시
- 스타일을 적용하고 싶은 태그에 style 속성을 사용해 **style="속성: 속성 값;"** 형태로 스타일 적용



```
<body>
  <h1>블루베리와 항산화 효능</h1>
  <p style="color:blue;">블루베리는 항산화제인 안토시아닌과 폴
리페놀을 다량 포함하고 있습니다.</p>
  <p>매사추세츠 보스톤에 있는 USDA 노화에 관한 인류 영양 연
구센터 (the USDA Human Nutrition Research Center on Aging) 의
자료에 의하면
  블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고
합니다. </p>
</body>
```

주요 선택자

전체 선택자

- 페이지에 있는 모든 요소를 대상으로 스타일을 적용할 때 사용
- 다른 선택자와 함께 모든 하위 요소에 한꺼번에 스타일을 적용하려고 할 때 주로 사용

기본형 * {속성:속성 값; 속성:속성 값; ...}

예) * {
margin:0;
padding:0;
}

웹 브라우저마다 기본적으로 설정해 놓은 '패딩' 과 '마진' 의 기본 값이 있습니다. 브라우저마다 그 값이 똑같지 않기 때문에 일반적으로 전체 선택자(*)를 사용해서 패딩과 마진 값을 0으로 리셋합니다.

주요 선택자

태그 선택자

- 문서에서 특정 태그를 사용한 모든 요소에 스타일이 적용됨

기본형 태그 { 스타일 }

블루베리에 관한 연구

블루베리와 항산화 효능

블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.

매사추세츠 보스톤에 있는 USDA 노화에 관한 인류 영양 연구센터 (the USDA Human Nutrition Research Center on Aging) 의 자료에 의하면 블루베리는 과일 중에서 가장 항산화 작용이 뛰어난 과일이라고 합니다.

블루베리와 노화

USDA 인류 영양 연구센터(the USDA Human Nutrition Research Center) 실험실에서 신경과학자들은 쥐들에게 블루베리를 먹임으로써 노화에 의한 인지능력의 손실을 예방해 준다는 사실을 발견하였습니다.

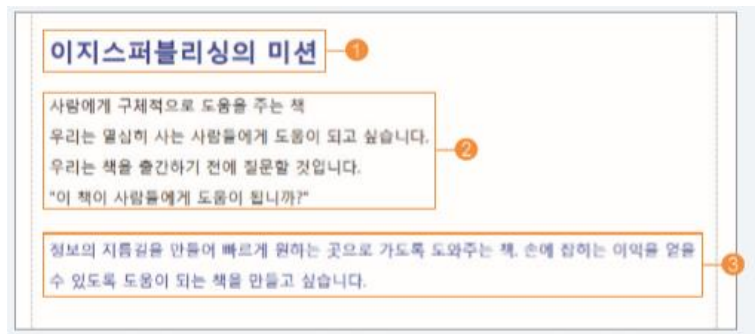
```
<head>
  <title>블루베리(Blueberry)</title>
  <style>
    h2 {
      color:blue;
    }
    p{
      font-size:12px;
      margin-left:20px;
    }
  </style>
</head>
<body>
  <h1>블루베리에 관한 연구</h1>
  <h2>블루베리와 항산화 효능</h2>
  <p>블루베리는 항산화제인 안토시아닌과 폴리페놀을 다량 포함하고 있습니다.</p>
  <p>매사추세츠 보스톤에 있는 ..... 과일이라고 합니다. </p>
  <h2>블루베리와 노화</h2>
  <p>USDA 인류 영양 연구센터..... 사실을 발견하였습니다. </p>
</body>
```

주요 선택자

클래스(class) 선택자

문서 안에서 여러 번 반복할 스타일이라면 클래스 선택자로 정의.

마침표(.) 다음에 클래스 이름 지정



```
<style>
  .bluetext {
    color:blue;
  }
</style>
```

① .bluetext 클래스 선택자 지정

```
<h2 class="bluetext">이저스퍼블리싱의 미션</h2>
<p>사람에게 구체적으로 도움을 주는 책<br>
.....
<p class="bluetext">정보의 지름길을 만들어 빠르게 원하는 곳으로 가도록 도와
주는 책. 손에 잡히는 이익을 얻을 수 있도록 도움이 되는 책을 만들고 싶습니다.
```

② 클래스 선택자 지정되지 않음

③ .bluetext 클래스 선택자 지정

사용해도 다르게 나타낼 수 있습니다!

주요 선택자

id 선택자

문서 안에서 한번만 사용한다면 id 선택자로 정의.
파운드(#) 다음에 id 이름 지정

class 선택자와 id 선택자의 차이

공통점 : 요소의 특정 부분에만 스타일 적용

차이점

- 클래스 선택자 : 문서 안에서 여러 번 반복
- id 선택자 : 문서 안에서 한번만 사용



```
<style>
  #container {
    width:600px; /* 너비 */
    padding:15px; /* 테두리와 내용 사이의 여백 */
    border:1px dotted gray; /* 테두리 스타일 */
  }
</style>
```

container id 선택자 적용

```
<div id=" c ">
  <h1>회사 소개</h1>
```

주요 선택자

그룹 선택자

- 같은 스타일을 사용하는 선택자를 한꺼번에 정의
- 쉼표(,)로 구분해 여러 선택자를 나열

기본형 이름1, 이름2, ...

```
h1 {  
  text-align:center; /* 가운데 정렬 */  
}  
h2 {  
  text-align:center; /* 가운데 정렬 */  
}
```

→

```
h1, h2 {  
  text-align:center; /* 가운데 정렬 */  
}
```

캐스캐이딩 스타일 시트(CSS)

캐스캐이딩의 의미

- 캐스캐이딩(Cascading) : '위에서 아래로 흐른다'는 뜻
- 선택자에 여러 스타일이 적용될 때 스타일 충돌을 막기 위해 '위에서 아래로 흐르며 적용되는' 방법을 선택

캐스캐이딩의 원칙

- ① 스타일 우선순위 - 스타일 규칙의 중요도, 적용 범위에 따라 우선순위가 결정되고
우선순위에 따라 위에서 아래로 스타일 적용
- ② 스타일 상속 - 태그들의 포함 관계에 따라 부모 요소의 스타일을 자식 요소로, 위에서 아래로 전달

※ 스타일 시트에서 '캐스캐이딩'은 가장 기본적인 개념이기 때문에 일반적으로 '스타일 시트'는 '캐스캐이딩 스타일 시트(css)'와 같은 의미로 사용됨

캐스캐이딩 스타일 시트(CSS)

*. 스타일 우선 순위

1. 얼마나 중요한가에 따라

시스템에서 만든 스타일. 사용자가 제어할 수 없음

=>

브라우저의 기본 스타일

=>

웹 사이트를 만들 때 제작자가 만든 스타일

2. 얼마나 한정지을 수 있는가에 따라



3. 소스 순서에 따라

- 중요도가 같다면 소스 순서에 따라 결정
- 소스에서 나중에 온 스타일이 먼저 온 스타일을 덮어씀

캐스캐이딩 스타일 시트(CSS)

*. 스타일 우선 순위

이지스퍼블리싱의 미션

사람에게 구체적으로 도움을 주는 책,
우리는 열심히 사는 사람들에게 도움이 되고 싶습니다.

우리는 책을 출간하기 전에 질문할 것입니다.

"이 책이 사람들에게 도움이 됩니까?"



CSS3와 CSS 모듈

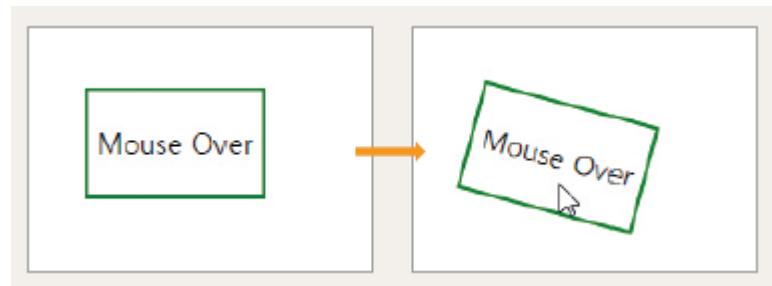
CSS3란

- CSS1 → CSS2 → **CSS3**
- CSS3부터는 배경이나 글꼴, 박스 모델 등 수십 개 기능을 주제별로 규약을 따로 만듦 → "CSS 모듈"
- CSS3는 한번에 표준 규약이 결정되지 않음.

CSS3와 브라우저 접두사

- 표준 규약이 아닌 속성들은 브라우저에 따라 다른 방식으로 지원됨
→ 속성 이름 앞에 접두사(prefix)
- 표준 규약이 완성된 속성도 옛날 버전의 모던 브라우저 사용자를 고려하기 위해 브라우저 접두사를 붙여 사용하기도 함

접두사	설명
-webkit-	웹킷 방식 브라우저용(사파리, 크롬 등)
-moz-	게코 방식 브라우저용(모질라, 파이어폭스 등)
-o-	오페라 브라우저
-ms-	마이크로소프트 인터넷 익스플로러



```
<style>
  .box:hover {
    -webkit-transform: rotate(15deg);
    -moz-transform: rotate(15deg);
    -o-transform: rotate(15deg);
    -ms-transform: rotate(15deg);
    transform: rotate(15deg);
  }
</style>
</head>

<body>
  <div class="box">Mouse Over</div>
</body>
```