

## Section 03

## CSS

## DHTML의 기초

- HTML + CSS + JavaScript + DOM 의 합성어
- DHTML의 등장 배경
  - 웹의 이용이 일반인에게 급속도로 확산
  - 일방적인 문서전달 체제가 갖는 정적인 특성 배제
  - 사용자와 상호작용할 수 있는 동적요소의 필요성
  - 일반 사용자들이 어렵지 않게 동적인 효과를 거둘 수 있는 기술이 요구
    - CGI, 자바애플릿은 일반인이 다루기에는 어려움이 따름
- DHTML의 역사
  - 1997년 HTML 4.0 버전부터 지원되기 시작
  - 2001년 DOM 레벨 1과 2, CSS(Cascading Style Sheet) 레벨 1과 2의 표준이 채택
  - 2004년 DOM L3 표준 채택; 2005년 12월 CSS L3 발표

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## DHTML의 기초 – 특 징

- 서버에 의지하거나 추가적인 프로그램에 의존하지 않으면서 브라우저에 의해 사용자에게 동적인 기능을 제공
  - 마우스나 브라우저 자체의 이벤트에 의해 웹 페이지의 내용을 변경가능
- 스크립트 언어를 이용
  - 브라우저에 나타나는 내용이나 외형을 사용자 임의대로 조절
  - 브라우저에 의해 실행되므로 서버에 부담을 주지 않음
  - 이벤트(Event)에 대해 빠른 반응을 보이는 장점을 가짐
- DHTML은 문서 객체 모델(DOM: Document Object Model)을 정의
  - 문서를 객체화
  - 스크립트 언어를 통해 객체에 접근하여 내용 변경 가능
  - 스타일 시트(Style Sheet)로 문서의 외형을 정의

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## DHTML의 기초 – 문제점

- 브라우저에서의 호환성
  - 브라우저 기술이 표준화보다 앞서 발전
  - 브라우저마다 독자적으로 동적인 기능이 추가
    - 넷스케이프사는 커뮤니케이터 4.0 버전에서 Layer 태그
    - W3C는 Layer 태그를 무시하고 CSS(Cascading Style Sheet)에 기반을 둔 DHTML을 채택
    - 마이크로소프트사의 인터넷 익스플로러 4.0 버전에서 W3C의 표준인 DOM 레벨 1에 기반한 CSS를 지원
  - 브라우저마다 문서 객체 모델과 스크립트의 처리가 다름
    - 실제 스크립트를 작성해 보면 많은 부분에서 차이가 발생
  - 웹 문서 작성시 두 종류의 브라우저에서 모두 오류 없이 실행되는 스크립트를 작성이 요구

---

---

---

---

---

---

---

---

---

---

## Section 03 CSS

## DHTML의 구성 요소

- 문서 객체 모델
  - DOM (Document Object Model)
- 스타일 시트 (Style Sheet)
- 레이어 (Layer)
- 스크립트 언어 (Script Language)

---

---

---

---

---

---

---

---

## Section 03 CSS

## 문서 객체 모델 (1)

- 넷스케이프사의 자바스크립트에서 사용되기 시작한 개념
- 브라우저와 웹 페이지의 구성요소를 다루기 위해 사용
  - 브라우저의 내장 객체와 브라우저에서 표현되는 웹 페이지의 텍스트, 이미지, 폼과 같은 각 요소에 이름을 지정하여 이들 각각을 객체로 정의

---

---

---

---

---

---

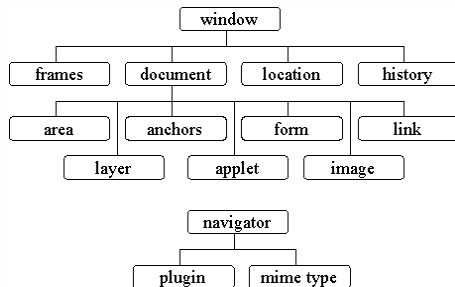
---

---

## Section 03 CSS

## 문서 객체 모델 (2)

- 웹 브라우저의 계층적 객체 구조




---

---

---

---

---

---

---

---

## Section 03

## CSS

## 문서 객체 모델 (3)

- window 객체
  - 최상위 레벨의 객체
  - 문서와 데이터를 주고 받는 작업이 모두 이 객체를 통해 일어남
  - frame, document, history, location 등의 하위 객체를 포함
    - 자바스크립트는 document라는 이름을 사용하여 문서에 접근
    - document 객체의 하위 객체에 접근하기 위해서는 상위 객체와 하위 객체 이름 사이에 점(.)으로 연결한 이름을 사용
      - 문서에 포함된 이미지에 접근하기 위해 document.image와 같이 사용

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 문서 객체 모델 (4)

- 넷스케이프와 익스플로러간에 차이점이 존재
  - 넷스케이프에서는 자바 스크립트 언어에서 DOM 객체를 정의함
  - 익스플로러에서는 스크립트 언어와는 독립적인 객체로 DOM을 정의함
- 넷스케이프와 익스플로러의 버전 6부터는 모두 W3C의 표준인 DOM 레벨 1과 2를 완전히 지원함

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 시트 (1)

- 문서의 외형을 다양하게 표현하기 위해 생긴 개념
  - 페이지 레이아웃과 폰트, 색상, 배경 및 웹 페이지 상의 다른 요소들의 스타일을 세부적으로 제어
  - 웹 페이지의 스타일을 문서의 내용과 분리시켜 스타일 정의 부분의 수정으로 전체 문서의 스타일을 변경
  - 스타일을 별도의 문서 파일로 유지하여 하나의 문서만 수정하고도 한꺼번에 여러 페이지의 외형 수정 가능

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 시트 (2)

- HTML에서는 문서가 브라우저에서 표현되는 방식인 스타일을 지정하기 위하여 태그와 속성을 이용
  - 내용이나 문단마다 각각 스타일을 정의해 주어야 하므로 동일한 내용을 반복하는 불편
  - 문서의 내용과 스타일이 혼재되어 HTML 소스를 이해하는데도 어려움이 존재
- 문서의 물리적 스타일 정보를 논리적인 내용으로부터 분리
  - 문서의 내용이 알아보기 용이
  - 반복되는 스타일 정의로 인한 불편이 해소
  - 스타일의 변경이 용이
  - 스타일 시트는 여러 문서에도 반복 적용이 가능
    - 일관성 있는 웹 문서 제작이 용이

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 시트 (3)

- 종류
  - CSS(Cascading Style Sheet)
    - 속성과 속성값을 이용하여 스타일을 정의
  - JSSS(JavaScript Style Sheet)
    - 자바스크립트로 스타일 정보를 제어하는 방법
- CSS 방법을 사용하는 것이 보편적
  - JSSS는 넷스케이프 4.0 이상의 버전에서 지원
  - 익스플로러에서는 JSSS를 사용할 수 없음

---

---

---

---

---

---

---

---

## Section 03

## CSS

## Layer (1)

- 웹 페이지의 내용을 표현하기 위한 투명한 캔버스
  - 이미지를 삽입 가능
  - 텍스트 등을 표현 가능
- 위치를 결정하는 요소를 포함
  - 값을 다르게 설정하면 많은 레이어를 포개어 놓을 수 있음

---

---

---

---

---

---

---

---

## Section 03

## CSS

## Layer (2)

- 종류
  - CSS(Cascading Style Sheet) 레이어
    - <DIV> 태그나 <SPAN> 태그를 사용
    - 스타일 시트를 사용해서 만든 것이기 때문에 넷스케이프와 익스플로러 양쪽 모두에서 사용이 가능
  - <LAYER> 태그를 이용한 레이어
    - <LAYER> 태그는 넷스케이프의 자체 태그이기 때문에 익스플로러에서는 지원하지 못함
- 웹 문서를 양쪽 브라우저에서 모두 읽을 수 있도록 하려면 CSS 레이어를 이용하는 것이 바람직

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스크립트 언어

- 스크립트는 어떠한 객체에 대해 그것의 상태를 바꾸기 위해 그에 대한 특성이나 행동에 대해 기술하는 역할을 수행
  - 마우스의 움직임, 폼의 입력 등 웹 페이지에서 일어나는 이벤트를 받아들여 문서 객체 모델로 정의된 객체의 상태를 동적으로 변경
- 종류
  - JavaScript - 넷스케이프에서 지원
  - JScript - 인터넷 익스플로러에서 지원
  - 비주얼 베이직 스크립트(VBScript)
  - JavaScript와 JScript는 문법과 기능면에서 거의 다를 바 없기 때문에 작성한 웹 페이지를 익스플로러와 넷스케이프 두 브라우저에서 모두 사용 가능하도록 하기 위해 JavaScript를 사용하는 것이 일반적

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## DHTML 문서 작성

- 스타일 시트를 이용한 웹 문서 작성
- 레이어를 이용한 웹 문서 작성
- 스크립트 언어를 이용한 웹 문서 작성

---

---

---

---

---

---

---

---

---

---

## Section 03 CSS

### CSS – 개요

- Cascading Style Sheets의 약자: 계단형 스타일시트
- 정의
  - CSS는 문서의 모양을 정의하는 속성을 모아 놓은 언어
  - 문서의 외형을 구체적으로 명시하기 위한 문장의 집합
- 역사
  - CSS의 표준화 작업과 신기술 및 팁을 제공하는 W3C에서 1996년 12월 **CSS레벨1** 발표
  - 1998년 5월 **CSS레벨2** 발표
  - 2005년 12월 **CSS 레벨3** 발표
    - 2007년 6월 **CSS3 Working Draft** 발표

---

---

---

---

---

---

---

---

## Section 03 CSS

### CSS – 장점 (1)

- 스타일의 변경과 웹 페이지의 일관성을 유지가 용이
  - <H3> 태그에 이탤릭 효과와 글자색으로 녹색을 지정하기 위해서 HTML에서는 모든 <H3>태그에 대해 일일이 다음과 같이 설정
 

```
<H3> <i> <FONT color=green> 녹색 이탤릭 </FONT> </i> </H3>
```
  - 스타일 시트를 사용하면 다음과 같이 <H3>의 스타일을 선언
 

```
H3 { font-style : italic ; color : green }
```
  - 한 번만 정의해 주면 웹 페이지에 포함된 모든 <H3> 태그에 같은 스타일을 적용

---

---

---

---

---

---

---

---

## Section 03 CSS

### CSS – 장점 (2)

- 스타일 시트를 사용하게 되면 문서의 내용과 스타일 정보가 분리
  - 문서의 내용을 알아보기 쉽게 한다
  - 똑같은 내용의 스타일 정보를 반복해서 작성해야 하는 번거로운 작업을 피할 수 있다
  - 스타일 시트 부분만을 수정하여 문서 전체의 스타일을 변경 가능하다
  - 스타일의 변경이 보다 쉽고 일관성 있게 이루어진다

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CSS - 개요

## ■ 특징

- 글꼴의 크기나 모양, 컬러, 마진 등 특정 이름으로 미리 지정해 놓고 사용하는 기능
- 문서들의 일관성을 쉽게 유지할 수 있고, 문서의 구조와 스타일이 분리되어 논리적으로 더 나은 문서의 작성이 가능
- 문서의 유형을 정의할 때 사용되는 추상적인 개념
- 웹 페이지의 텍스트를 다양하게 변화

---

---

---

---

---

---

---

---

## Section 03

## CSS

## HTML 예제

```
<HTML>
<HEAD>
<TITLE>CSS이해하기</TITLE>
</HEAD>
<BODY>
<H3><FONT COLOR=navy>CSS 이해하기</FONT></H3>
<FONT color=blue size=2>1. CSS를 이용하면 웹문서를 좀
더 시각적으로 표현할 수 있다.</FONT><br>
<FONT color=blue size=2>2. CSS를 이용하면 문서의 구조
를 쉽게 파악할 수 있게 된다.</FONT>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CSS로 표현된 예제

```
<HTML>
<HEAD> <TITLE>CSS이해하기</TITLE>
<STYLE type="text/css">
  H3 {color : navy}
  FONT {color: blue; font-size: 10pt}
</STYLE>
</HEAD>
<BODY>
<H3>CSS 이해하기</H3>
<FONT>1. CSS를 이용하면 웹문서를 좀더 시각적으로 표현할 수 있
다.</FONT> <br>
<FONT>2. CSS를 이용하면 문서의 구조를 쉽게 파악할 수 있게 된
다.</FONT>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## CSS – 정의 방법 3가지

- <BODY>태그 내부 삽입형: in-line style
  - 문서 내의 각 태그 요소에 직접 스타일 시트를 정의
  - 형식 : <태그 style="속성:값; 속성:값; ">
- 문서 <HEAD> 내 지정형: embedded/global style
  - 문서의 <HEAD> 부분에 스타일 시트를 정의
  - 형식:
    - <style type="text/css">스타일 내용 정의</style>
    - <STYLE>태그 와 @import 명령어를 사용하여 외부 스타일시트 연결
- 외부 문서 삽입형: linked/external style
  - 스타일 시트를 별도의 파일로 저장하여 다른 문서에서도 같은 스타일 시트를 사용할 수 있도록 하는 것
  - 형식 : <head> 부분에 정의
    - <link rel="stylesheet" href="스타일 외부 파일 전체경로" type="text/css">

---

---

---

---

---

---

---

---

## Section 03 CSS

## 인라인 스타일 시트 – inline

- 각 태그 내에 STYLE 속성으로 정의

```

<html>
<head>
<title>스타일 정의</title>
</head>
<body>
<h1 style="font-family:굴림; font-style:italic;..."> CSS </h1>
<p style="font-size:20pt; text-align:center"> 스타일 시트 연습
</p>
</body>
</html>

```

{tag style="속성 : 속성값;  
속성 : 속성값"}

---

---

---

---

---

---

---

---

## Section 03 CSS

## 전역 스타일 시트 – embedded/global

- <HEAD> 태그 내에 <STYLE> 태그를 이용하여 정의

```

<html>
<head>
<title>스타일 정의</title>
<style type="text/css">
h1 {font-family:굴림; font-style:italic; ...}
P {font-family:휴먼매직체; font-size:20pt; ...}
</style>
</head>
<body> .....</body>
</html>

```

tag {속성 : 속성값;  
속성 : 속성값}

---

---

---

---

---

---

---

---



## Section 03 CSS

## 전역 스타일 시트 - 예제

```
<HTML>
<HEAD> <TITLE> 스타일 시트 사용법 </TITLE>
<STYLE TYPE = "TEXT/CSS">
  H2 {COLOR:RED; ALIGN:CENTER; MARGIN-TOP:50;}
  P {COLOR:BLUE; ALIGN:RIGHT; MARGIN-LEFT:50;}
</STYLE>
</HEAD>
<BODY>
  <H2> STYLE IMPORT 이용한 연결 방법 </H2>
  <P> STYLE IMPORT 이용한 연결 방법 </P>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 전역 스타일 시트 - @IMPORT

```
stylelink.css
H2 {COLOR:RED; ALIGN:CENTER; MARGIN-TOP:50;}
P {COLOR:BLUE; ALIGN:RIGHT; MARGIN-LEFT:50;}

<HTML>
<HEAD> <TITLE> 스타일 시트 사용법 </TITLE>
<STYLE TYPE = "TEXT/CSS"> @IMPORT URL(stylelink.css);
</STYLE>
</HEAD>
<BODY>
  <H2> STYLE IMPORT 이용한 연결 방법 </H2>
  <P> STYLE IMPORT 이용한 연결 방법 </P>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 외부 스타일 시트 - external/linked

- 정의된 스타일 시트를 파일로 저장 후 각 문서에서 사용
- 스타일 시트 파일이름 : \*.CSS
- 주로 <HEAD> 태그 내에서 <LINK> 태그를 이용하여 선언

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="my.css">
</head>
<body> ..... </body>
</html>
```

my.css

```
h1 {font-family:굴림; font-style:italic; }
P {font-family:휴먼매직체; font-size:20pt; }
```

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CSS 예제 1- cascading

```

<HTML>
<HEAD>
  <TITLE> CSS 연습 </TITLE>
  <LINK REL=STYLESHEET TYPE="text/css" HREF="test.css">
  <STYLE TYPE="text/css">
    H3 { color : red }
  </STYLE>
</HEAD>
<BODY>
  <H3> H3는 무슨색일까? 빨간색이면 맞다. </H3>
  <H3 STYLE="color:green"> 그런데 여기는 H3가 녹색입니다. </H3>
</BODY>
</HTML>

```

H3 {color : navy}

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CSS 예제- cascading (우선순위)

- Inline/global/external
  - inline > global or external
- global/external
  - 가장 나중에 지정한 것이 적용

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CSS 예제 2- cascading

```

<HTML>
<HEAD><TITLE>CSS 이해하기</TITLE>
<link rel="stylesheet" type="text/css" href="test.css">
<STYLE type="text/css">
  H3 {color : red}
</STYLE>
</HEAD>
<BODY>
<H1>CSS 이해하기: 우선순위 연습 </H1>
<h3>1. CSS를 이용하면 웹 문서를 좀더 시각적으로 표현할 수 있
다.<br></h3>
<h3>2. CSS를 이용하면 문서의 구조를 쉽게 파악할 수 있게 된다.</h3>
</BODY>
</HTML>

```

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 정의 기본 형식

- 스타일의 정의는 스타일을 적용할 대상과 대상에 적용할 속성과 속성값으로 구성
  - 스타일을 적용할 대상을 선택자(Selector)라 하며 일반적으로 HTML 태그들이 선택자로 사용
  - 선택자를 지정하고 '{'와 '}' 안에 속성과 속성값으로 이루어진 스타일 정보를 기술하는 선언부 지정
  - 속성과 속성값은 콜론(:)으로 연결하여 정의한다.

**선택자 { 속성 : 속성값 }**

- 예 : <P> 태그에 포함되는 내용을 빨간색으로 보이도록 <P> 태그의 스타일을 정의

**P { color : red }**

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 정의 유형

H1	{color: blue}
선택자	속성 값

- 선택자와 선언부로 구성
  - 선택자(selector) : 'H1'
  - 선언부(declaration) : 'color: blue'
- 선언부
  - 속성(property/attribute) : 'color'
  - 값(value) : 'blue'
- 모든 HTML 요소가 선택자가 될 수 있음

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 스타일 시트에서 자주 사용되는 속성

- color : 글씨 색 - 16진수, 색상 이름, rgb(red,green,blue)
- background-color : 배경 색
- background-image : 배경 이미지 - url("배경이미지의 위치")
- font-family : 글자 체 - 굴림, 바탕, arial,...
- font-size : 글자 크기 - 숫자pt
- font-weight : 글자 굵기 - normal, bold, bolder, lighter
- font-style : 글자 모양 - normal, italic, oblique
- text-indent : 들여 쓰기 - 숫자, 백분율
- text-align : 정렬 - left, center, right
- text-decoration : 글자 꾸밈 - none, underline, overline, line-through,...
- border-width : 경계선 굵기 - 숫자pt
- border-color : 경계선 색
- border-style : 경계선 유형 - solid, double, groove, ridge, inset, outset

---

---

---

---

---

---

---

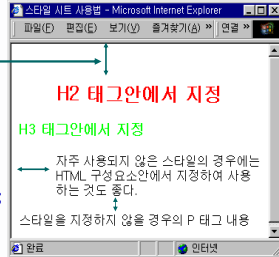
---

## Section 03 CSS

## 스타일 지정 예

H2 {COLOR:RED; ALIGN:CENTER; MARGIN-TOP:50; }  
 P {COLOR:BLUE; ALIGN:RIGHT; MARGIN-LEFT:50;}

margin-top=50px;

margin-left=50px;  
margin-bottom=15px;

## Section 03 CSS

## 스타일 지정 방법: 1. 선택자의 그룹화

- 선택자의 요소들을 comma(,)를 사용하여 그룹화

```
<STYLE TYPE="text/css">
H1 {COLOR:RED; FONT-WEIGHT:BOLD; SIZE:15pt;}
H2 {COLOR:RED; FONT-WEIGHT:BOLD; SIZE:15pt;}
H3 {COLOR:RED; FONT-WEIGHT:BOLD; SIZE:15pt;}
</STYLE>

↓

<STYLE TYPE="text/css">
H1,H2,H3 {COLOR:RED; FONT-WEIGHT:BOLD; SIZE:15pt;}
</STYLE>
```

```
H1 { background : yellow }
P { background : yellow }
```



```
H1, P { background : yellow }
```

## Section 03 CSS

## 선언부의 그룹화

- 선언부의 요소들을 세미콜론(;)을 사용하여 그룹화

```
H1 { font-weight: bold;
font-size: 12pt;
line-height: 14pt;
font-family:
helvetica;
font-variant:
normal;
font-style: normal; }
```



```
H1 {font-weight: bold;
font-size: 12pt;
line-height: 14pt;
font-family: helvetica;
font-variant: normal;
font-style: normal; }
```

## Section 03 CSS

## 스타일 지정 방법: 2. 속성 상속/계승

- 속성을 지정하는 태그가 어떠한 태그 안에서 사용되는가에 따라 스타일을 지정할 수 있도록 함
- 요소들 사이의 공백으로 구분함

```
H1 { color: blue }
EM { color: red }
```



H1 요소의 하위로 나오는 EM 요소의 글자색만 변경

```
H1 { color: blue }
H1 EM { color: red }
```

## Section 03 CSS

## 선택자의 속성계승 예제

```
<head><title>인터넷 활용</title>
<style type="text/css">
  H3 {background:yellow; color:blue }
  H3 EM {background:navy; color:white }
  EM {color:red}
</style>
</head>
<body>
  <EM>스타일은 이탤릭체의 빨간색 글자로 표현됩니다.</EM>
  <H3>이 경우는 <EM>로만 선택자에 의한 스타일이 </EM>적용됩니다.</H3>
</body>
```

## Section 03 CSS

## 3. CLASS 속성을 이용한 스타일 정의

- 클래스를 사용하면 원래의 태그 스타일이 있더라도 그 스타일을 적용하지 않고 클래스 스타일로 정의한 속성이 적용
  - 특정 태그에만 적용되는 클래스의 스타일을 정의하려면 선택자로 스타일을 적용하고자 하는 태그의 이름을 쓰고, 점(.)과 클래스 이름을 사용
  - 형식: **선택자.클래스명 { 속성 : 속성값 }**
- 특정 태그에만 적용되는 CLASS 스타일 정의:
  - 예) <P> 태그 중 클래스 속성이 blue인 태그에만 스타일을 지정

```
P.blue { background : blue }
```

## Section 03 CSS

## 3. CLASS 속성을 이용한 스타일 정의(Cont'd)

- 클래스로 지정된 스타일을 적용하기 위해 HTML 문서의 <BODY> 태그 내에서 다음과 같은 형식으로 사용
- 형식:

<태그명 CLASS = "클래스 이름"> ~ </태그명>

- 예) 태그가 적용되는 범위의 배경색으로 파란색을 지정

<P CLASS = "blue">이 단락은 CLASS 속성을 이용한 태그입니다.</P>

---

---

---

---

---

---

---

---

## Section 03 CSS

## 3. CLASS 속성을 이용한 스타일 정의(Cont'd)

- 모든 태그에 적용되는 클래스 스타일을 적용하려면 다음과 같이 선택자에 태그 이름 없이 마침표(.)와 클래스 이름만을 사용
- <BODY> 태그 안에 위치하는 어떤 태그에서도 태그의 CLASS 속성값으로 클래스 이름만 지정해주면 해당 스타일을 사용 가능
- 형식:

. 클래스명 { 속성 : 속성값 }

---

---

---

---

---

---

---

---

## Section 03 CSS

## CLASS 속성을 이용한 스타일 정의 - 예제1

- 태그에 적용되는 스타일과 모든 태그에 적용되는 스타일을 비교

```
<HEAD>
<TITLE> 클래스 스타일 만들기 </TITLE>
<STYLE TYPE = "text/css">
  P { background : yellow; color : blue }
  P.boldtxt { font-weight : bold }
  .redtxt { color : red }
  .greentxt { color : green }
</STYLE>
</HEAD>
```

---

---

---

---

---

---

---

---

## Section 03

## CSS

## CLASS 속성을 이용한 스타일 정의 - 예제1

```
<BODY>
<P> 이 텍스트는 스타일의 적용을 받아 글자가 파란색으로
표시됩니다.
<P CLASS="redtxt"> 같은 태그이지만 글자가 빨간색으로
표시됩니다.
<P CLASS="greentxt"> 녹색의 글자도 사용할 수 있습니다.
<H2 CLASS="redtxt"> P 태그가 아니라도 빨간글자가 적용
됩니다.</H2>
<H2 CLASS="greentxt"> P 태그가 아니라도 녹색글자가 적용
됩니다.</H2>
<P CLASS="boldtxt"> P 태그에 굵은 글자도 사용할 수 있습니다.
<H2 CLASS="boldtxt"> P 태그에만 적용되는 클래스이므로
적용되지 않습니다.</H2>
</BODY>
```

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 클래스를 이용한 스타일 정의-예제2

- 클래스별 스타일 지정
- 하나의 태그에 여러 가지 종류의 서로 다른 스타일 적용하고자 할 때

```
<html>
<head> <title>스타일 정의</title>
<style type="text/css">
  p.red{font-size:25pt; color:red}
  p.blue{font-size:20pt; color:blue}
</style></head>
<body>
<p class="blue"> 스타일 시트를 이용한 다양한 문서 제작</p>
<p class="red"> 클래스를 이용한 스타일 정의 </p>
</body>
</html>
```

---

---

---

---

---

---

---

---

---

---

## Section 03

## CSS

## 클래스를 이용한 스타일 정의-예제3

- 사용 예) <HTML>
 

```
<HEAD>
  <TITLE> CLASS 속성 설명 </TITLE>
  <STYLE TYPE="text/css">
    H1.italic { font-style: italic }
  </STYLE>
</HEAD>
<BODY>
  <H1> 보통 머리글 </H1>
  <H1 CLASS=italic> 이탤릭체 머리글
</H1>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

---

---

## Section 03 CSS

## 클래스를 이용한 스타일 정의-예제4

- 하나의 스타일을 여러 가지 태그에 스타일 적용

```
<html>
<head> <title>스타일 정의</title>
<style type="text/css">
    .red{font-size:25pt; color:red}
    .blue{ font-size:20pt; color:blue}
</style>
</head>
<body>
<p class="blue"> 스타일 시트를 이용한 다양한 문서제작</p>
<h1 class="red"> 클래스를 이용한 스타일정의</h1>
</body>
</html>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 클래스를 이용한 스타일 정의-예제5

- 사용 예)

```
<HTML>
<HEAD>
<TITLE> CLASS 속성 설명</TITLE>
<STYLE TYPE="text/css">
    .italic { font-style: italic }
</STYLE>
</HEAD>
<BODY>
<H1 CLASS=italic> 이탤릭체 머리글</H1>
<P CLASS=italic>이탤릭체로 지정된 문단 부분</P>
</BODY>
</HTML>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 클래스를 이용한 스타일 정의 - 예제6

```
<head><title>인터넷 활용과 웹 디자인</title>
<style type="text/css">
    H2 {background:navy; color:white }
    P.title {font-size:15pt; background:rgb(215,200,240)}
    P.intro {font-size:13pt; background:rgb(255,255,255)}</style>
</head><body>
<h2>광의 계획서 소개</h2>
<p class=title>광의 개요</p><p class=intro> 인터넷의 기본 프로토콜인
TCP/IP 프로토콜을 배워 인터넷의 기초를 이해하여, 인터넷에서 활용하여 사
하며 HTML과 자바 스크립트를 중심으로 홈페이지 제작 기술을 익힌다.</p>
<p class=title>교 재</p>
<p class=intro>인터넷 활용과 웹 디자인, 홍길동외 3인, 길동출판사, <br>
웹 프로그래밍, 홍길동 저, 길동출판사 </p>
<p class=title>평가 방법</p>
<p class=intro> 중간고사 30%<br> 기말평가 30%<br>
출 석 20%<br> 과 제 물 15%<br>
실 습 5%<br> 계 100% </p></body>
```

---

---

---

---

---

---

---

---



## Section 03 CSS

## 4. ID 속성을 이용한 스타일 정의

- 스타일 시트를 정의할 때, 개별적으로 고유한 이름을 붙여서 스타일을 정의
  - 이미 지정된 스타일 속성 이외에 또 다른 속성을 지정하여 사용 가능
  - CSS 구문에서는 다음의 형식과 같이 '#' 기호를 사용하여 ID 이름과 스타일을 정의
  - 형식: **#ID이름 { 속성 : 속성값 }**
  - HTML 태그에서는 ID 이름을 사용하여 해당 태그에 대한 스타일을 지정
    - 사용되는 ID 이름은 대소문자를 구분하여 사용해야 하며 스타일은 한 번만 지정 가능
  - 형식: **<태그명 ID = "ID이름"> ~ </태그명>**

---

---

---

---

---

---

---

---

## Section 03 CSS

## ID를 이용한 스타일 정의-예제

- class스타일 정의와 동일

```

<html> <head>
<title>스타일 정의</title>
<style type="text/css">
#red{font-size:25pt; color:red}
#blue{ font-size:20pt; color:blue}
</style>
</head>
<body>
<p id="blue"> 스타일 시트를 이용한 다양한 문서제작</p>
<h1 id="red"> 클래스를 이용한 스타일정의 </h1>
</body> </html>
  
```

p#red{font-size:25pt; color:red}  
p#blue{font-size:20pt; color:blue}

---

---

---

---

---

---

---

---

## Section 03 CSS

## ID/Class 속성을 이용한 스타일 정의-예제

- ID/class 속성을 이용한 스타일의 선언 방법과 스타일을 태그에 적용하기 위한 사용 방법을 보여줌

```

<HEAD><TITLE>클래스 스타일 만들기</TITLE>
<STYLE TYPE ="text/css">
P { background : yellow; color : blue; }
.boldtxt { font-weight : bold; }
#once { font-style : italic;}
</STYLE>
</HEAD>
<BODY>
<H2 ID = "once"> 이 스타일은 ID 속성을 이용한 스타일입니다.</H2>
<H2 CLASS = "boldtxt"> H2 태그에도 굵은 글자가 적용됩니다.</H2>
<P CLASS = "boldtxt"> P 태그에 굵은 글자도 사용할 수 있습니다.
</BODY>
  
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 4. ID를 이용한 스타일 정의(Cont'd)

- 일반적으로 **class** 속성은 태그를 분류하여 같은 태그라도 다른 스타일을 지정하고자 할 때 사용
- ID 속성은 웹페이지 전체에서 단 한번만 사용될 스타일을 지정할 때 사용
  - 하나의 ID 이름은 문서에서 단 한번만 사용

---

---

---

---

---

---

---

---

## Section 03 CSS

## CSS - 클래스 선택자와 아이디 선택자

구분	클래스 선택자	아이디 선택자
시작기호	.(마침표)	#(hash)
속성명	class	id
역할	문서에서 주로 사용되는 서식을 클래스명으로 지정하여 문서에 통일감을 준다.	스크립트로 아이디에 지정된 스타일의 속성을 변경할 수 있어 DHTML로 가는 관문 역할을 한다.
주요용도	문서를 꾸며준다.	레이어의 동작을 제어한다.
주 사용 태그	<p> <div> <span> 등	<div> 등
적용범위	여러 개의 태그에 적용 가능하다.	하나의 태그에만 적용하여 사용한다.
공통점	이름은 사용자가 임의로 지정한다. 태그에 제한 없이 어느 태그에서나 사용 가능하다.	

---

---

---

---

---

---

---

---

## Section 03 CSS

## CLASS와 ID 속성 사용 (예)

```
<HTML><HEAD>
<TITLE> CLASS와 ID를 사용한 스타일 지정 </TITLE>
<STYLE TYPE="text/css">
P.bold { font-size: 20pt; font-weight: bold }
P.italic { font-size: 20pt; font-style: italic }
B.bold { font-weight: bold }
B#pastoral { color: green; font-size: 50pt; font-style: italic }
</STYLE>
</HEAD>
<BODY>
<P CLASS=bold> 이것은 크고 굵은 글씨체의 문단입니다. </P>
<P CLASS=italic> 이것은 크고 이탤릭체의 문단입니다.</P>
<P> 이것은 일반적인 문장인데, <B ID=pastoral> Bold </B> 요소는 가운데 <B CLASS=bold>강조</P>되게 됩니다. </P>
<P CLASS=bold> 이것은 또 다른 크고 굵은 글씨체의 DIV입니다. </P>
</BODY></HTML>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 5. 가상 선택자 (Pseudo selector)

- 지금까지의 선택자 정의로도 다양한 스타일들에 충분하나, 일부 불편적인 효과를 해결하지 못하고 있다.
  - 적절히 쓰면 불필요한 클래스를 사용하지 않아도 되는 방법
- HTML 자원에는 없지만 CSS1 레벨에서 양식화 과정에 외부 정보를 허용하도록 확장하기 위해 **가상 클래스 (Pseudo-classes)** 와 **가상 엘리먼트 (Pseudo-elements)** 개념을 소개
  - HTML 자원의 외부 정보를 허용하기 위하여 CSS선택자(selector)에 사용

---

---

---

---

---

---

---

---

## Section 03 CSS

## 5. 가상 선택자 (..cont'd)

- **가상/의사 선택자란**
  - 실제로는 존재하지 않는 요소나 특수한 상황에서만 발생하는 동적인 요소를 CSS 명령으로 제어하는 기법 즉, 클래스가 정의된 것처럼 간주
- 즉, 가상 클래스 선택자는
  - 웹 문서의 소스에는 실제로 존재하지 않지만 필요에 의해 임의로 가상의 선택자를 지정하여 사용하는 것
- 예) a:link -
  - <a> 요소에 클래스 link가 선언된 것처럼 생각하고 선택자를 만드는 것

---

---

---

---

---

---

---

---

## Section 03 CSS

## 5. 가상 선택자(..cont'd)

- **:link** 가상클래스
  - 아직 방문한 적이 없는 연결에 적용
- **:visited** 가상클래스
  - 사용자가 방문한 적이 있는 연결에 적용
- 예)
  - **a:link** - 방문되지 않은 링크의 스타일
  - **a:visited** - 방문된 링크의 스타일
  - **a:hover** - 마우스가 위에 있을 때의 스타일
  - **a:active** - 마우스로 클릭되는 때의 스타일

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 정의를 위한 CSS 기본 속성

- 글꼴 (font)
- 문단 (text)
- 색과 배경 (color)

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 정의를 위한 CSS 속성: Module

- 선택자 (selectors)
- 박스 모델 (Box Model)
  - Borders : 스타일과 색상
  - Margin : elements 안에서 콘텐츠 영역의 크기 설정(요소 크기)
  - Padding : 콘텐츠와의 경계 사이 결정
  - Backgrounds : 요소의 배경 설정
- 텍스트 효과 (Text Effect)
- 2차원/3차원 변환 (2D/3D Transformation)
- 애니메이션 (Animation)
- 다중 컬럼 레이아웃 (Multiple Column Layout)
- 사용자 인터페이스 (User Interface)

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 정의를 위한 CSS 속성: UI

- 화면에 출력될 element들에 디자인 요소를 추가하는 속성
  - 커서의 모양을 변경
  - 리스트의 형태를 변경 및 이미지 삽입
  - 문서의 배경색, 배경이미지
  - 스크롤바(Scroll bar)의 색상 변경
  - 보임, 안보임 조정
- 관련속성
  - cursor
  - scrollbar
  - Classification(목록) : List-style-type, list-style-image, list-style-position, list-style
  - Display
  - Background-color, background-image, background-attachment, background-repeat, background

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 시트의 속성 - 글꼴 (font)

- 폰트와 관련된 속성을 이용하여 글자체, 크기, 모양, 굵기 등의 스타일을 지정

속성	속성값	의미
font-family	글자체 이름 (예: Sans-Serif, 굴림)	글자체
font-style	normal, italic	글자 모양
font-size	pt 단위의 숫자	글자 크기
font-weight	normal, bold, bolder, lighter	글자 굵기

## Section 03 CSS

## font size 설정

- 폰트의 단위
  - pt - 포인트 (1포인트 : 0.72인치)
  - px - 픽셀
  - % - 퍼센트
  - em - 배수(scale factor)
    - em을 쓰는 이유는 IE8과 같이 zoom기능이 없는 브라우저에서 브라우저의 글꼴 크기 옵션을 조절했을 때 정확하게 하기 위한인데 최근의 브라우저들이 모두 zoom 기능을 지원하고 있어 em 대신 px를 사용
  - 키워드 - xx-small, x-small, small, medium, large, x-large, xx-large

```
body {
  font-size: 20pt;
}
```

↑                      ↑  
폰트 크기를 나타내는 속성    20포인트 크기를 의미한다.

## Section 03 CSS

## font-style 설정

- 이탤릭체 여부 : normal, italic, oblique

```
h1#s1 { font-style: normal; } heading 1
h1#s2 { font-style: italic; } heading 1
```

↑  
이탤릭체로 지정

## Section 03 CSS

## font-weight 설정

- font-weight : 볼드체 여부(normal, bold)

```
h1#s1 { font-weight: normal; } heading 1
h1#s2 { font-weight: bold; } heading 1
```

↑  
볼드체로 설정

---

---

---

---

---

---

---

---

## Section 03 CSS

## 글꼴 관련 스타일 속성의 사용법 (1)

- 글자색 지정: **color** 속성
  - p{color: orange}
  - p{color: #FFA500}
  - p{color: rgb(255,165,0)}
  - 글자색을 오렌지색으로 표현
- 글자체 (글꼴) 지정: **font-family** 속성
  - p{font-family: 돋움, 궁서, 바탕}
  - 돋움체가 없으면 궁서체, 궁서체가 없으면 바탕체로 표현
  - CSS Parser는 앞의 글꼴부터 읽으며 글꼴이 사용자PC에 없을 경우 다음 글꼴을 읽음

---

---

---

---

---

---

---

---

## Section 03 CSS

## 글꼴 관련 스타일 속성의 사용법 (2)

- 글자크기 지정: **font-size** 속성
  - 예) p{font-size: 20pt}
  - 사용법 : {font-size : 속성값}
    - 절대사이즈= xx-small | x-small | small | medium | large | x-large | xx-large
      - Font 요소의size 속성값이1~7 : 한 단계 커지거나 작아질 때 마다 1.2배율(Scaling factor)가 적용
    - 상대사이즈= larger | smaller
    - 비율(percentage) : 상위가 없을 경우 기본 크기(12pt)를 기준으로 100분율 상대크기
    - 1 pt (Points) = 1/72 inches (1 인치를 72 등분한 단위)

---

---

---

---

---

---

---

---

## Section 03 CSS

## 글꼴 관련 스타일 속성의 사용법 (3)

- 글자굵기 지정: **font-weight** 속성
  - p{font-weight: bold}
  - 글자를 진하게
    - 사용법: { font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400(normal) | 500 | 600 | 700(bold) | 800 | 900 }
- 글꼴모양 지정: **font-style** 속성
  - p{font-style: italic}
  - 글자를 기울임
    - 사용법: { font-style: normal(정상) | italic(이태릭) | oblique(기운것) }
- **font-variant** 속성
  - 사용법: { font-variant : normal | small-caps(작은 대문자 형태)

## Section 03 CSS

## 글꼴 관련 스타일 속성의 사용법 (4)

- font 속성 : font에 관한 속성을 한꺼번에 지정하는 단축형 (short hand) 속성
  - font-style, font-variant, font-weight, font-size, line-height, font-family등을 한꺼번에 지정할 수 있는 단축형 속성
  - 속성값: (반드시 1, 2, 3 순서대로작성)
    1. font-style, font-variant, font-weight 중 하나 또는 전부
    2. font-size/line-height (font-size만 쓸 경우 "/" 생략가능)
    3. font-family
  - 예)
    - {font : italic small-caps bold 12px/1.3em Arial}
    - {font : bold 12px/1.3em Verdana}
      - /\*bold체, 12px 글씨 크기에 행 간격1.3em, 글꼴 Verdana\*/
    - {font : italic 12px Tahoma}
      - /\*기울임체, 12px 글씨 크기, 글꼴 Tahoma\*/
    - {font : 12px/1.5em bold sans-serif}
      - /\*bold가 순서에 맞지 않아 이후의 sans-serif 글꼴은 무시

## Section 03 CSS

## 스타일 시트의 속성 - text

- 텍스트의 모양, 배치, 간격 등을 자유롭게 조절

속성	속성값	의미
text-align	left, center, right, justify	정렬 기준
text-indent	숫자 또는 백분율	들여 쓰기
vertical-align	baseline, bottom, middle, text-bottom, text-top, top, sub, super	세로 정렬
text-decoration	none, underline, overline, line-through, blink	글자 꾸밈
text-spacing	숫자	글자 간격
word-spacing	숫자	단어 간격

## Section 03 CSS

## 스타일 시트의 속성 - 텍스트

- 예제: 텍스트 관련 속성 중 text-decoration 속성을 이용하여 속성값을 center로 지정하여 가운데 정렬을 수행하고 text-decoration 속성의 속성값을 line-through로 지정하여 문자열에 취소선을 적용

```
<HEAD>
<STYLE TYPE = "text/css">
  H2 { text-align : center}
  P { text-decoration : line-through}
</STYLE>
</HEAD>
<BODY>
  <H2>가운데 정렬</H2>
  <P>취소선 적용</P>
</BODY>
```

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 시트의 속성 - 색과 배경

- 스타일 시트를 이용하면 배경색 뿐만 아니라 이미지를 가져와 배경 이미지로 지정 가능

속성	속성값	의미
color	색의 이름, 또는 rgb(red, green, blue)	색
background-color	배경색	
background-image	url("배경 이미지의 URL")	배경이미지

---

---

---

---

---

---

---

---

## Section 03 CSS

## 스타일 시트의 속성 - font &amp; color

- 예제: <H2> 태그에 대해서는 font-weight 속성을 이용하여 파란색의 글자로 굵게 표시하였고, <P> 태그에 대해서는 빨간색 글자로 20포인트의 크기로 문자열을 출력

```
<HEAD>
<STYLE TYPE = "text/css">
  H2 { color : blue; font-weight : bold}
  P { color : red; font-size : 20pt}
</STYLE>
</HEAD>
<BODY>
  <H2>파란색 굵은 글자체</H2>
  <P>크기가 20인 빨간 글자</P>
</BODY>
```

---

---

---

---

---

---

---

---