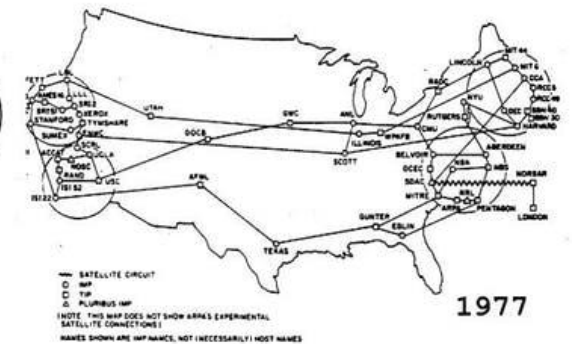
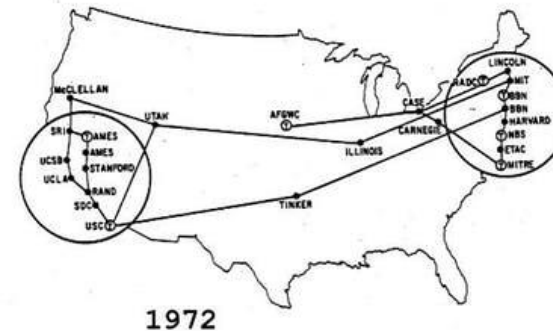
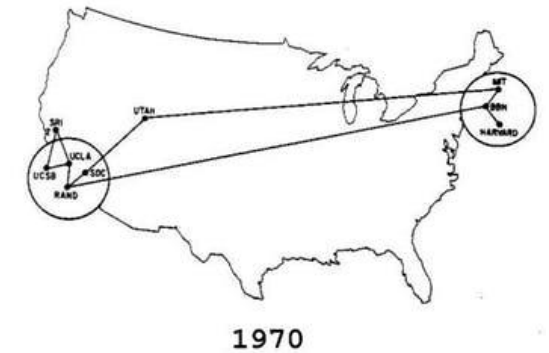
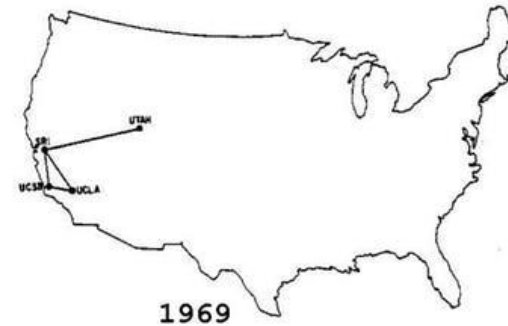


# 인터넷과 웹 소개

# 인터넷과 웹

## ▶ 인터넷의 시작

- ▶ 예전부터 과학자들은 정보가 연결되어 있는 세계를 꿈꿔 왔음
- ▶ 냉전 시대인 **1960년대**, 소련에서 최초로 인공위성 발사에 성공하자 위기를 느낀 미국은 신기술 연구를 위한 **ARPA** 부서를 창설(미 국방부 주도)
- ▶ **1969년 ARPANET** 개발
- ▶ **ARPANET의 주요 목적**
  - ▶ 정보 교환과 연구 역량 극대화 도모
  - ▶ 네트워크를 분산, 하나가 단절되더라도 나머지에 영향이 없도록 하고자 함
- ▶ 군사 목적의 **ARPANET**은 대학, 기업과 연계 발전해 나감
- ▶ 이 중, 민간 연구용이 현재의 인터넷으로 발전

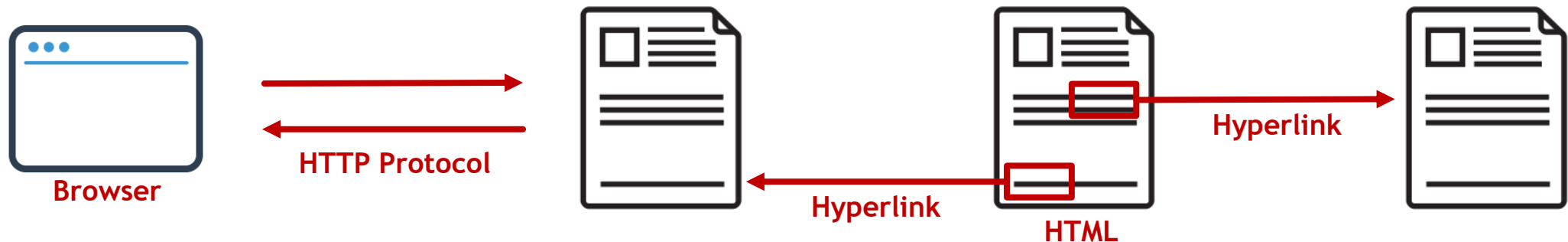


연도별 ARPANET 연결 상황

# 인터넷과 웹

## ▶ World Wide Web(WWW)의 발명

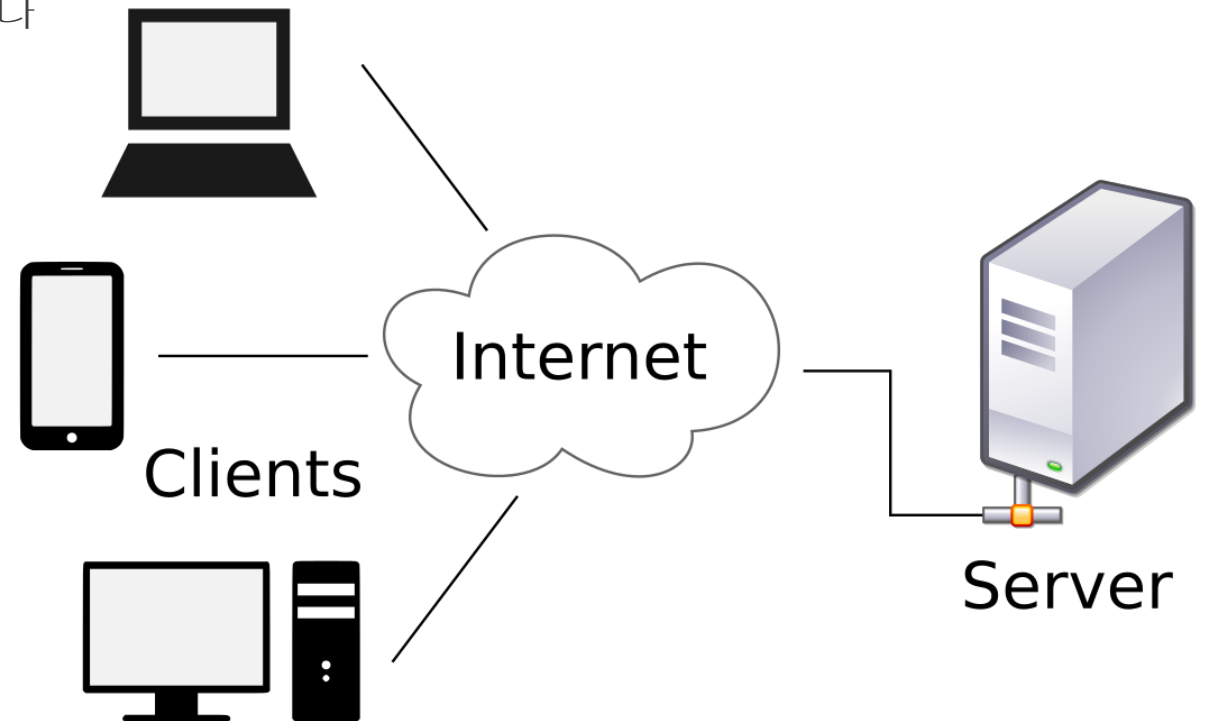
- ▶ 유럽입자물리연구소(CERN)의 팀 버너스-리(Tim Berners-Lee)가 고안 1969년 ARPANET 개발
- ▶ Hyperlink 개념을 도입한 문서 형식인 HTML(HyperText Markup Language) 도입, 인터넷 상의 논문을 연결
  - ▶ Hyperlink : 문서와 문서 사이를 링크로 연결, 빠르게 이동할 수 있는 개념
- ▶ 문서에 주소를 붙이는 URL(Uniform Resource Locator) 개념 확립
- ▶ 인터넷에 연결된 컴퓨터 사이에서 메시지를 주고 받을 수 있는 통신 규약 HTTP 고안
- ▶ WWW : 전 세계 인터넷상의 콘텐츠를 연결, 공유할 수 있는 인터넷 기반 서비스



# 인터넷과 웹

## : 서버와 클라이언트

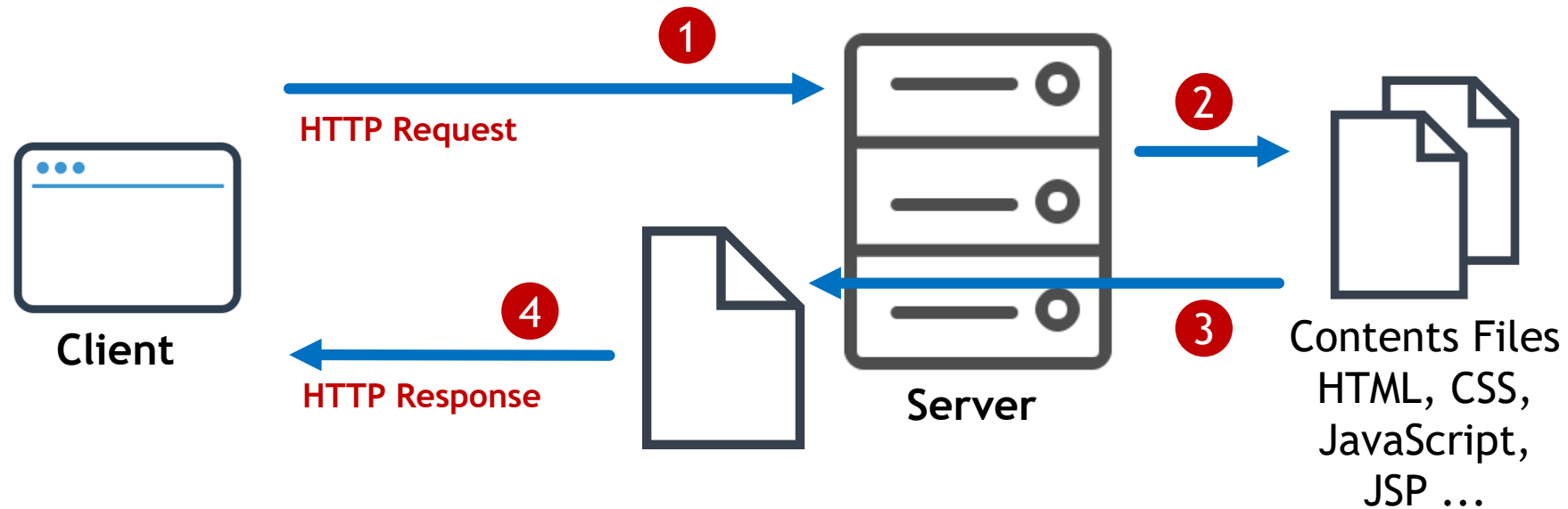
- ▶ 인터넷 상에 연결된 컴퓨터들은 역할에 따라 서버(Server)와 클라이언트(Client)로 구분된다
  - ▶ 클라이언트 : 정보를 요청하는 컴퓨터
  - ▶ 서버 : 다른 컴퓨터로부터 요청을 받아 정보 혹은 서비스를 제공하는 컴퓨터
- ▶ 서버 컴퓨터도 다른 컴퓨터로 요청을 보낼 수 있다  
이 경우, 해당 서버 컴퓨터도 클라이언트가 된다
  - ▶ 성능상의 구분이 아니며 역할에 따른 상대적 구분



# 웹의 작동 원리

## : 요청과 응답

- ▶ ① 클라이언트(브라우저)가 서버상에 있는 파일의 위치를 지정하여 요청
- ▶ ② 서버는 클라이언트가 요청한 파일을 내부에서 검색(혹은 동적으로 작성)
- ▶ ③ 서버는 클라이언트의 요청에 적합한 웹 리소스를 클라이언트로 전송
- ▶ ④ 클라이언트는 서버로부터 전송 받은 HTML을 기반으로 문서의 구조를 렌더링하여 화면에 표시

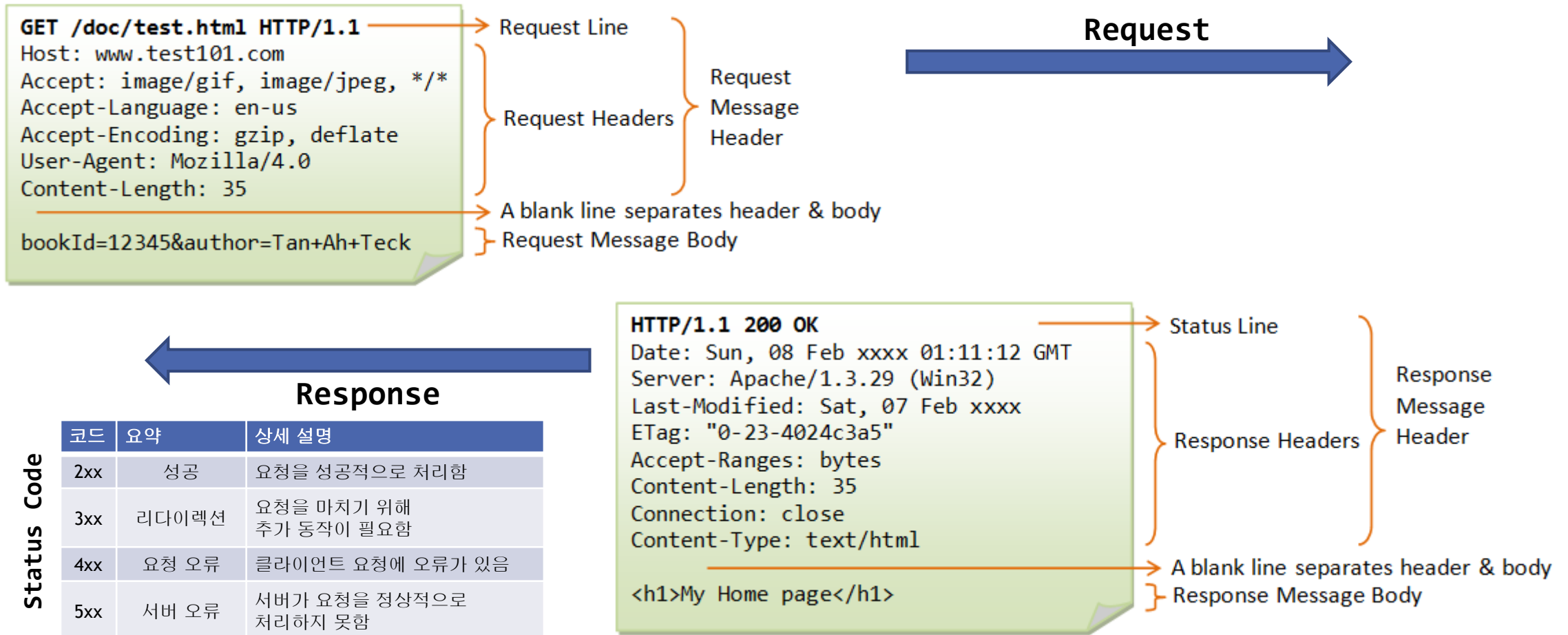


# 웹의 작동 원리

## : 요청과 응답

- ▶ 브라우저의 주소창에 웹 주소를 입력하면 일어나는 일들

- ▶ [https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)



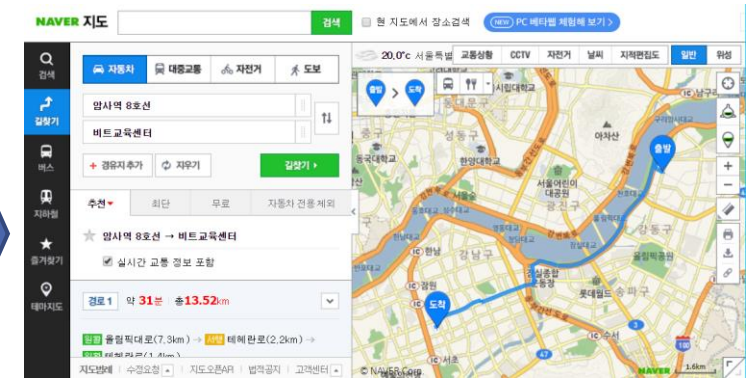
# 웹의 진화

- ▶ 단순히 문서 및 연구자료를 빠르게 교환하고자 고안되었던 웹은 대중화되면서 다양한 요구사항을 반영하며 진화하기 시작

<http://info.cern.ch> - home of the first website

From here you can:

- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)



## 하이퍼텍스트 문서

- 기본적인 서식과 하이퍼링크 포함한 텍스트 문서

## 웹 페이지

- 다양한 리소스(이미지, 오디오 등) 제공
- 시각적으로 풍부한 레이아웃
- 사용자와의 상호작용은 없음

## 웹 애플리케이션

- 사용자와의 상호 작용
- 복잡한 처리 가능
- 데스크탑 응용프로그램 수준으로 진화

# 웹 브라우저 전쟁, 그리고 웹 표준의 대두

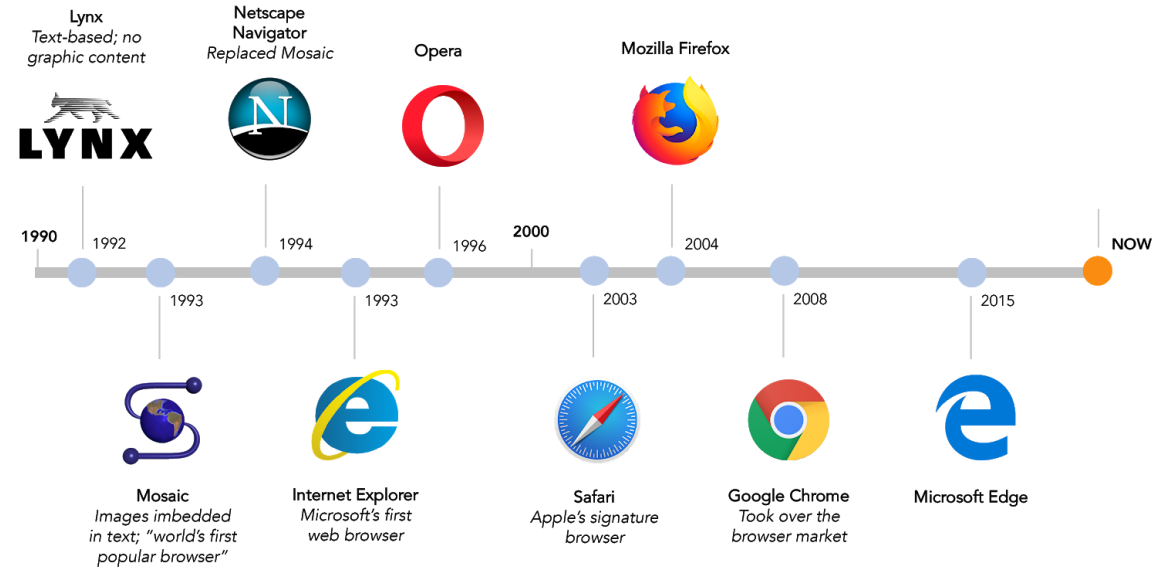
- ▶ 초창기 웹 브라우저 시장은 넷스케이프가 독주
- ▶ Microsoft가 Windows 98에 인터넷 익스플로러(IE)를 기본 탑재하면서 웹 브라우저 시장 독점
  - ▶ IE 6의 경우, 전 세계 웹 브라우저의 90% 이상 점유
- ▶ IE의 독점적 지위에 따른 웹 환경의 폐해(Active X 등)



- ▶ 사파리, 파이어폭스, 오페라 등 다양한 브라우저 영향력 확대
- ▶ 특히, 스마트폰 시대를 맞으며 IE의 영향력 하락



- ▶ 특정 웹 브라우저에 종속되는 웹 페이지는 더 이상 설 자리가 없음을 의미
- ▶ 웹 페이지 제작 기술에 표준의 필요성이 대두





# 웹 표준 정의

## ▶ W3C (World Wide Web Consortium)

- ▶ 웹 기술의 표준화를 담당하는 기구
- ▶ <http://www.w3.org>
- ▶ HTML, CSS, XML 등 웹과 관련된 다양한 기술에 대해 연구, 논의하며 관련 기술 표준을 제정
- ▶ 웹 표준은 W3C의 목표와 비전에 따라 연구한 결과를 웹에 반영시키기 위한 권고 사항

The screenshot shows the W3C website with the following elements:

- Header:** W3C logo, tagline "Leading the web to its full potential", a language selector set to "국가별 W3C", and a "Go" button.
- Navigation:** A horizontal menu with links for "STANDARDS", "PARTICIPATE", "MEMBERSHIP", and "ABOUT W3C".
- Left Sidebar:**
  - WEB AND INDUSTRY** (with a dropdown arrow):
    - Automotive
    - Entertainment (TV and Broadcasting)
    - Publishing
    - Web Commerce
    - Web Payments
    - Web of Data
    - Web and Telecommunications
    - Web of Things
  - WEB FOR ALL** (with a dropdown arrow)
- Main Content Area:**
  - News Item:** "EMVCo, FIDO Alliance, and W3C Form Interest Group to Enhance Security and Interoperability of Web Payments" dated "17 April 2019 | Archive". The text states: "The FIDO Alliance, EMVCo, and the World Wide Web Consortium (W3C) announced today the creation of a new Interest Group for organizations to collaborate on a vision for Web payment security and interoperability. Participants in the Web Payment Security Interest Group will define areas of collaboration and identify gaps between existing technical specifications in order to increase compatibility among different technologies." Below this, it mentions: "The Web Payment Security Interest Group charter defines a scope of activities that includes formulation of a vision for web payment security, development of use cases, gap analysis, liaisons with other organizations, and identification of standardization opportunities for each organization. The Interest Group does not publish specifications. Technical work is carried out in other groups within each organization."
- Right Sidebar:**
  - ABOUT W3C** (with a dropdown arrow): "The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web."
  - DONATIONS** (with a dropdown arrow): "Become a Friend of W3C: support the W3C mission and free developer tools."
  - W3C BLOG** (with a dropdown arrow): "W3C honored with Emmy®"

# 웹 표준 정의

- ▶ 특정 브라우저에서만 사용하는 비표준 기술은 배제
- ▶ W3C에서 정한 표준 기술
- ▶ 역할의 분담
  - ▶ HTML : 문서의 구조와 데이터를 담당
  - ▶ CSS : 문서의 표현 영역(디자인, 스타일)을 담당
  - ▶ JavaScript : 문서의 동적인 작동을 담당
- ▶ 요약

- HTML과 CSS를 웹의 표준으로 사용하자
- HTML은 구조를 잡고, CSS는 디자인을 담당하도록 역할을 분리하자



# 웹 표준 정의

## ▶ HTML과 CSS를 웹의 표준으로 사용하자

- ▶ 웹 표준에 따라 웹 페이지를 작성하게 되면 표준을 따르는 브라우저에서는 모두 같은 정보를 얻게 되므로 사용자가 가장 큰 혜택을 보게 된다
- ▶ IE, 파이어폭스, 크롬, 사파리 등 서로 다른 브라우저에서도 같은 화면을 보여주는 것, 즉 브라우저의 벽을 허물어주는 개념을 '**크로스 브라우징(Cross Browsing)**'이라 한다.

## ▶ HTML은 구조를 잡고, CSS는 디자인을 담당하도록 분리하자

- ▶ 정보의 구조는 **HTML**이 담당하고 **CSS**는 레이아웃과 디자인을 담당하도록 분리
- ▶ 하나의 콘텐츠 소스를 다양한 장비와 브라우저에서 사용할 수 있다는 의미

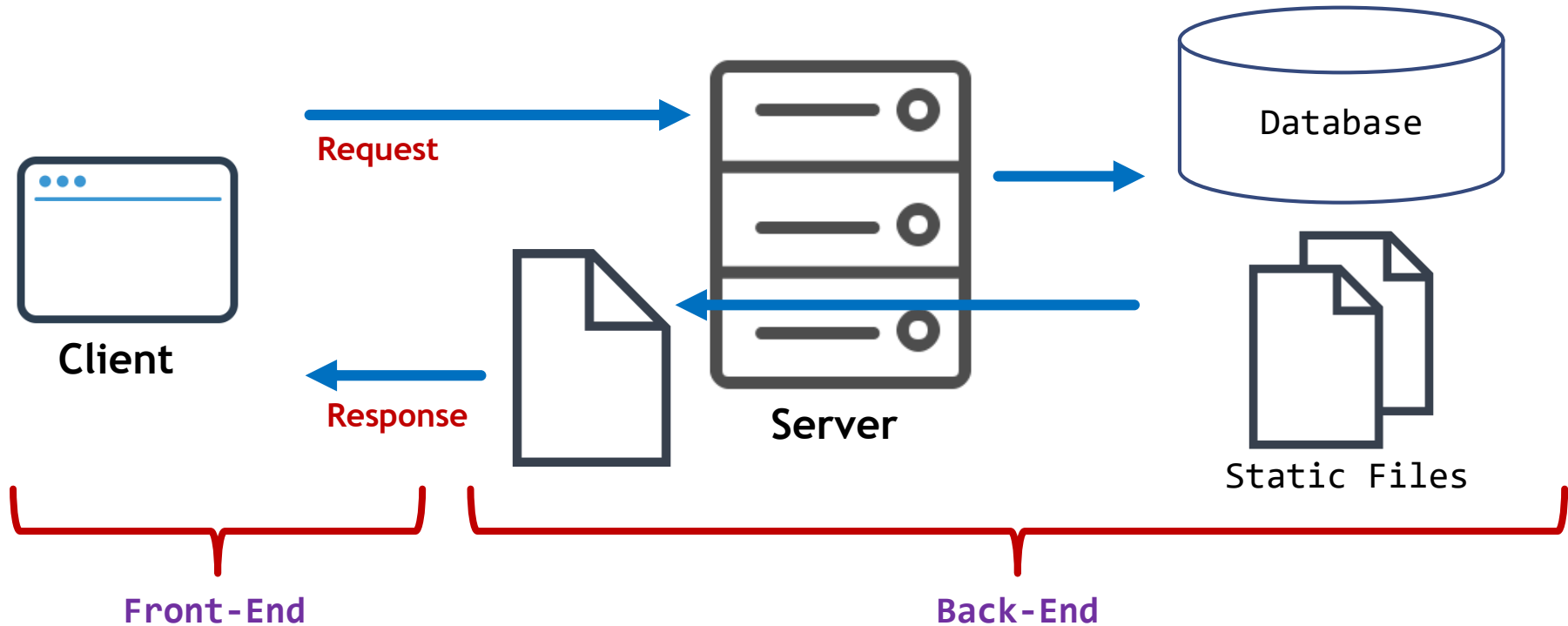
## ▶ 웹 접근성

- ▶ 어떤 장비, 어떤 브라우저를 사용하던지 해당 사이트를 얼마나 볼 수 있는가를 가늠하는 척도
- ▶ 장애 여부에 상관 없이 어떤 사용자도 해당 사이트 사용에 어려움이 없어야 한다는 의미
- ▶ 웹 표준을 지켜 작성한 **HTML** 문서는 웹 접근성을 준수하는데 비표준 문서들보다 용이하다
- ▶ 표현을 **CSS**로 분리한 **HTML** 문서는 웹 접근성을 준수하는데 표현과 정보가 섞여 있는 문서들보다 용이하다

# 웹 프로그래밍

## : 프론트엔드와 백엔드

- ▶ 웹 프로그래밍은 크게 프론트엔드(Front-End) 개발과 백엔드(Back-End) 개발로 구분
  - ▶ 분야별로 필요 기술과 요구사항이 다름



- 사용자가 직접 브라우저로 활용하는 영역
- 웹 콘텐츠 구조화, 레이아웃, 디자인
- HTML, CSS, JavaScript

- 사용자의 요청에 따라 데이터를 처리하고 응답하는 영역
- 운영체제, 네트워크, 데이터베이스 기술
- 웹 프레임워크 기술
- Java, PHP, Python, Node.js 등

# 개발환경의 구축

## : IDE 설치

### ▶ Visual Studio Code : <https://code.visualstudio.com/>

- ▶ 권장 설치 확장 프로그램 : Live Server

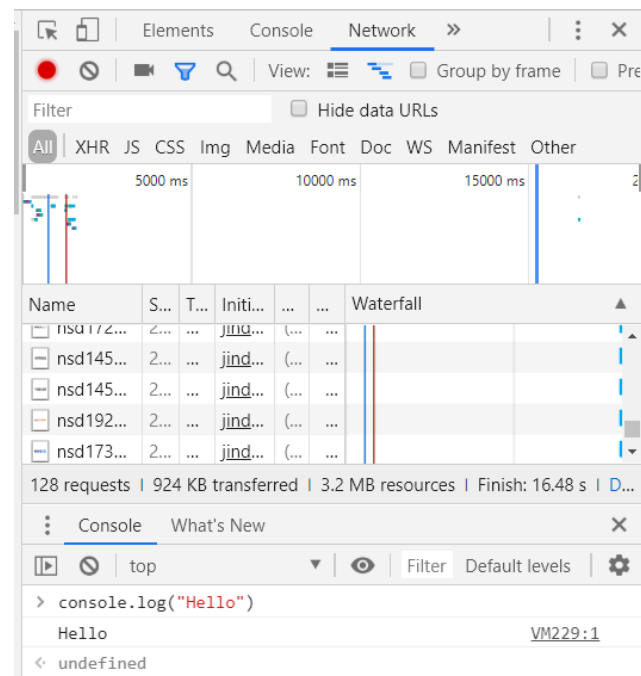
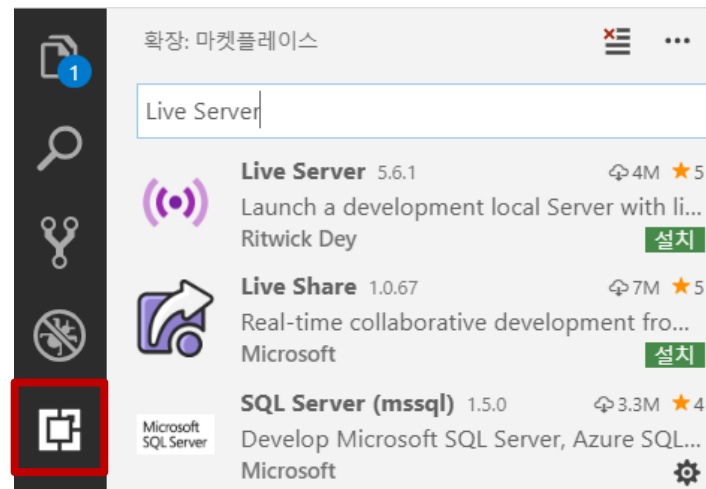
### ▶ 그 외 추천 IDE

- ▶ JetBrains WebStorm
- ▶ Atom
- ▶ Bracket 등

### ▶ 웹 브라우저와 개발자 도구(Developer Tool)

#### ▶ 추천 브라우저: Google Chrome

- ▶ 웹 표준을 가장 빠르게, 잘 반영하는 브라우저
- ▶ 개발자 도구를 이용, 빠르고 편리한 웹 프론트 개발을 지원
  - ▶ Windows의 경우, F12를 이용하여 개발자 도구를 열 수 있음

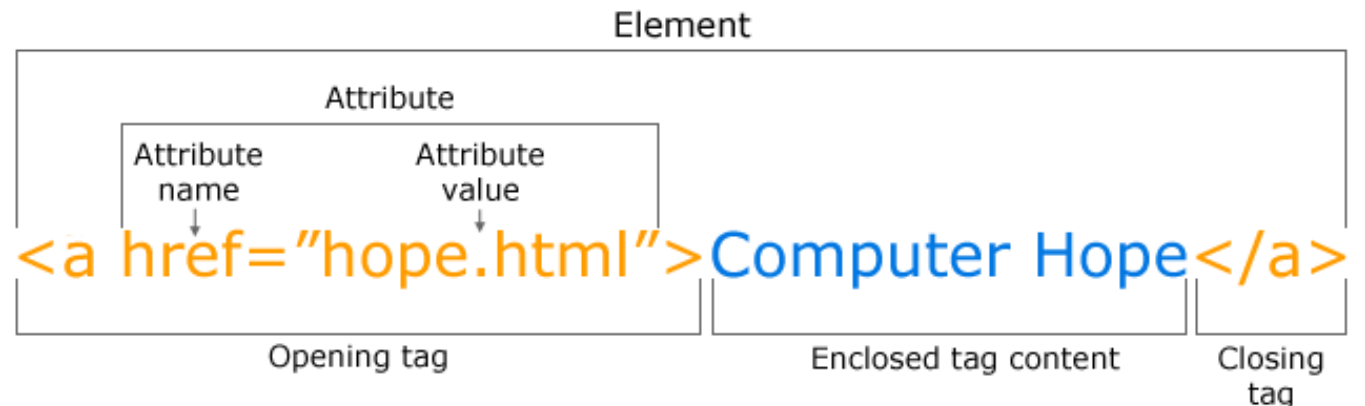


# HTML 기본 구조와 작성법

# HTML 기본 용어

## : 태그, 요소, 속성

- ▶ 태그
  - ▶ < > 사이에 오는 단어나 문자
  - ▶ 브라우저에게 포함된 콘텐츠의 구조와 의미에 관해 알려준다
- ▶ HTML 엘리먼트는 세 부분(여는 태그, 콘텐츠, 닫는 태그)으로 구성
  - ▶ 콘텐츠가 없는 엘리먼트의 경우, 닫는 태그 없이 단독으로 사용하기도 함 : 예) <br/>
  - ▶ 대부분 엘리먼트는 속성을 가지며 중첩이 가능함
- ▶ 속성(Attribute)
  - ▶ 태그의 동작을 제어하기 위해 여는 태그 안에 사용되는 특수 용어
  - ▶ 속성은 주로 속성명=속성값의 쌍으로 기술된다



# HTML 페이지 기본 구조

- ▶ `!DOCTYPE html`
  - ▶ 문서가 HTML5 형식임을 브라우저에 알림
- ▶ `html`
  - ▶ 모든 HTML 페이지의 루트 요소
- ▶ `head`
  - ▶ HTML 문서의 요약 정보 및 문서에서 사용될 스타일시트와 자바스크립트 명시
- ▶ `body`
  - ▶ 렌더링 되어 실제 브라우저에 표시될 문서의 본문

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Structure</title>
  </head>
  <body>
    <!-- 주석 -->
    <h1>HTML</h1>
    <p>기본적인 HTML 문서 구조입니다.</p>
  </body>
</html>
```



# HTML 기본 태그

## : 텍스트 관련 태그

### ▶ Heading Tag : <h1> ~ <h6>

- ▶ 문서의 제목을 표현할 때 사용
- ▶ h1은 가장 중요한 헤더 정보에, h6은 가장 낮은 중요도의 헤더 정보에 사용

...  
<h1>첫 번째로 중요한 헤더 정보</h1>  
<h2>두 번째로 중요한 헤더 정보</h2>  
<h3>세 번째로 중요한 헤더 정보</h3>  
<h4>네 번째로 중요한 헤더 정보</h4>  
<h5>다섯 번째로 중요한 헤더 정보</h5>  
<h6>여섯 번째로 중요한 헤더 정보</h6>  
...



**첫 번째로 중요한 헤더 정보**

**두 번째로 중요한 헤더 정보**

세 번째로 중요한 헤더 정보

네 번째로 중요한 헤더 정보

다섯 번째로 중요한 헤더 정보

여섯 번째로 중요한 헤더 정보

# HTML 기본 태그

## : 텍스트 관련 태그

### ▶ Paragraph Tag : <p>

- ▶ 문서의 본문(문단)을 표현할 때 사용

<p>문단은 p 태그로 표시할 수 있습니다.</p>



문단은 p 태그로 표시할 수 있습니다.

### ▶ 개행(줄 바꾸기) : <br>

- ▶ 기본적으로 HTML은 개행(줄 바꾸기)을 표시하지 않음.
- ▶ 줄 바꿈을 표시하기 위해서는 br 태그를 이용

<p>br 태그는 줄바꿈을<br> 표시할 수 있습니다.</p>



br 태그는 줄바꿈을 표시할 수 있습니다.

### ▶ 수평줄 표시 : <hr>

<hr>

- ▶ hr 태그는 다른 태그 내부에서는 사용하지 않도록 합니다 -> 웹 표준 위반



---

# HTML 기본 태그

## : 텍스트 관련 태그

## ▶ 특수 문자의 표기

- ▶ HTML 태그 내에 연속된 공백이 있어도 브라우저는 공백 한 개로 렌더링  
-> 공백을 모두 표시하기 위해서는 `&nbsp;`를 사용해야 함

<p>여러 개의 공백 표시</p>

[illegible]

## ▶ HTML 특수 문자의 종류

특수 문자	출력 문자
&nbsp;	공백
&lt;	<
&gt;	>
&amp;	&
&copy;	©
&quot;	"



여러 개의 공백 표시

여러 개의 공백 표시

# HTML 기본 태그

## : 텍스트 관련 태그

### ▶ 텍스트 모양 관련 태그

- ▶ 웹 페이지의 글자에 형태와 의미를 부여

태그	설명
b	굵은 글자
i	기울어진 글자
small	작은 글자
sub	아래 첨자
sup	위 첨자
ins	밑줄 글자
del	가운데 줄이 그어진 글자(취소선)

### ▶ 텍스트 강조 태그 : em, strong

태그	설명
em	텍스트 강조
strong	더 강한 텍스트 강조

```
<p><b>글자 모양을 다양하게</b></p>
<p><i>글자 모양을 다양하게</i></p>
<p><small>글자 모양을 다양하게</small></p>
<p>글자 <sub>모양</sub>을 다양하게</p>
<p><ins>글자 모양을 다양하게</ins></p>
<p><del>글자 모양을 다양하게</del></p>
<hr>
<p><em>글자 모양을 다양하게</em></p>
<p><strong>글자 모양을 다양하게</strong></p>
```

글자 모양을 다양하게

*글자 모양을 다양하게*

글자 모양을 다양하게

글자 모양을 다양하게

글자 모양을 다양하게

~~글자 모양을 다양하게~~

*글자 모양을 다양하게*

**글자 모양을 다양하게**

# HTML 기본 태그

## : 하이퍼링크 태그 - A

### ▶ a : 하이퍼링크(Hyperlink)

- ▶ 사용자의 선택(클릭)에 따라 관련된 특정 정보로 이동

```
<a href="http://www.google.com">구글</a>
```

이동할 위치

출력 문자열

구글

Google

Google Search

I'm Feeling Lucky

- ▶ href 속성 : 링크 클릭시 이동할 위치를 지정하는 속성
- ▶ 외부 링크 : 링크할 외부 웹 서버의 주소와 페이지이 위치를 모두 기술한다
  - ▶ 예) <https://movie.naver.com/movie/running/current.nhn>
- ▶ 내부 링크 : 현재 페이지와 동일한 서버 내에 위치하므로 서버 주소는 명시할 필요 없음
  - ▶ 절대 경로 : 최상위(ROOT) 디렉터리를 기준으로 경로를 지정
  - ▶ 상대 경로 : 현재 페이지를 기준으로 트리 경로를 지정

# HTML 기본 태그

## : 목록 태그

- ▶ 여러 항목의 목록을 나열하는 태그
  - ▶ ul : 순서가 없는 목록(Unordered List)
  - ▶ ol : 순서가 있는 목록(Ordered List)
- ▶ li : ul, li 하부에 위치하는 목록의 항목 태그(List Item)

순서 없는 목록

```
<ul>  
  <li>사과</li>  
  <li>바나나</li>  
  <li>오렌지</li>  
</ul>
```

- 사과
- 바나나
- 오렌지

순서 있는 목록

```
<ol>  
  <li>사과</li>  
  <li>바나나</li>  
  <li>오렌지</li>  
</ol>
```

1. 사과
2. 바나나
3. 오렌지

- ▶ 목록 태그는 현대 웹 어플리케이션에서 내비게이션 메뉴를 만들 때도 널리 사용

---

홈   사회   정치   경제   국제   문화   **IT**   랭킹   연재   포토   TV

---

# HTML Basic

## : Table Tags

### ▶ table 태그

- ▶ 표를 만들 때 사용하는 태그
- ▶ 테이블 캡션 : caption
- ▶ 테이블 헤더 : thead
- ▶ 테이블 본문 : tbody
- ▶ 테이블 푸터 : tfoot

### ▶ 테이블 태그는 계층 구조로 작성

- ▶ tr 태그 (Table Row)
  - ▶ 테이블의 한 열(row)을 표현
- ▶ td 태그 (Table Data)
  - ▶ tr 태그 내에서 하나의 행(column)을 표현
- ▶ th 태그 (Table Header)
  - ▶ td 대신 사용. 제목 셀로 사용

성적표

이름	나이	점수
철수	23	80
영희	22	90
평균		85

thead

tbody

tfoot

```
<table border="1">
  <caption>성적표</caption>
  <thead>
    <tr>
      <th>이름</th>
      <th>나이</th>
      <th>점수</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>철수</td>
      <td>23</td>
      <td>80</td>
    </tr>
    ...
  </tbody>
  <tfoot>
    <tr>
      <td>평균</td>
      <td></td>
      <td>85</td>
    </tr>
  </tfoot>
</table>
```

# HTML Basic

## : Table Tags

- ▶ 열(TR)과 행(TD)의 확장
  - ▶ rowspan : 열의 확장
  - ▶ colspan : 행의 확장

colspan="2"

cell(1, 1)	
cell(2, 1)	cell(2, 2)

```
<table border="1">
  <tr>
    <td colspan="2">cell(1, 1)</td>
  </tr>
  <tr>
    <td>cell(2, 1)</td>
    <td>cell(2, 2)</td>
  </tr>
</table>
```

rowspan="2"

cell(1, 1)	cell(1, 2)
	cell(2, 2)

```
<table border="1">
  <tr>
    <td rowspan="2">cell(1, 1)</td>
    <td>cell(1, 2)</td>
  </tr>
  <tr>
    <td>cell(2, 2)</td>
  </tr>
</table>
```



# HTML 기본 태그

## : 이미지 태그

### ▶ **img** : 웹 페이지 내에 이미지를 삽입하는 태그

- ▶ **img** 태그는 태그 내부에 별도 콘텐츠 영역이 없이 단독 태그로 사용
- ▶ 이미지를 보지 못하는 시각장애인은 스크린 리더로 웹 페이지를 듣게 되는데 이때, **img** 태그에 설정된 **alt** 속성의 내용으로 어떤 이미지인지 이해
- ▶ 또한, 검색엔진 역시 **img** 태그의 **alt** 속성을 이용, 웹 페이지의 이미지가 어떤 이미지인지 식별할 수 있게 됨

표시할 이미지의 위치


```
  

```



대체 텍스트

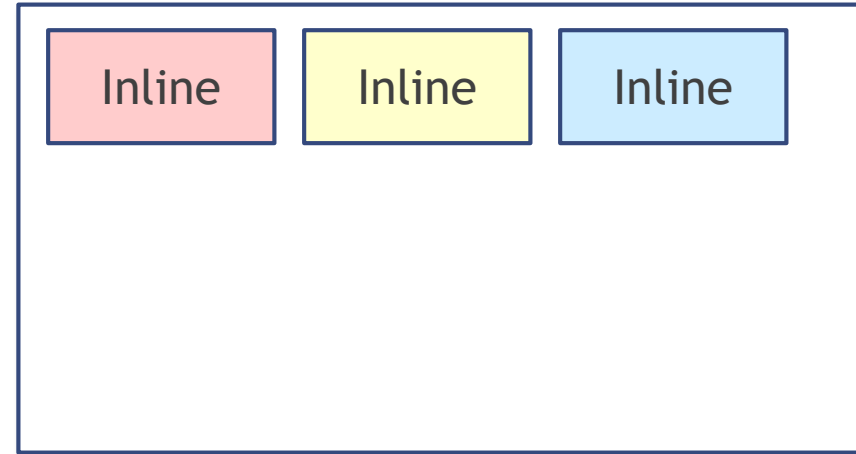
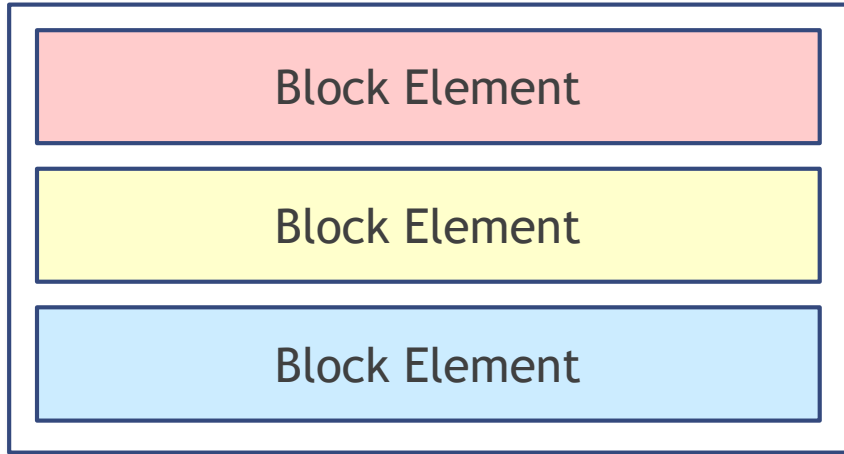


 그림이 없습니다

# HTML 기본 태그

## : 공간 분할 태그

- ▶ HTML 요소가 공간을 점유하는 방식 : block vs inline



- ▶ 모든 시각적인(공간을 점유하는) HTML 요소들은 block 혹은 inline 방식으로 영역을 차지한다.
- ▶ div와 span은 임의의 작은 구역을 설정할 때 사용하는 태그
  - ▶ div : Block 요소로 줄 바꿈이 일어난다.
  - ▶ span : inline 요소로 줄 바꿈이 일어나지 않는다.  
문맥의 흐름이 유지된다.

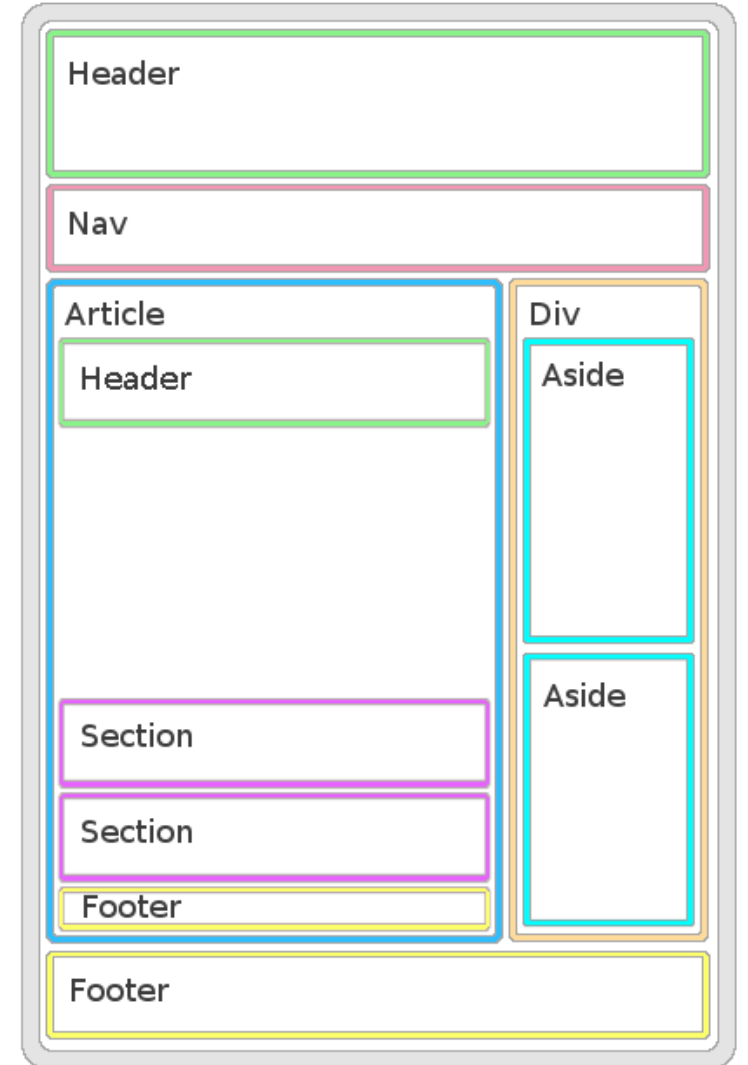
블록 형식 태그	인라인 형식 태그
div 태그 h1 ~ h6 태그 p 태그 목록 태그 테이블 태그 form 태그	span 태그 a 태그 input 태그 글자 형식 태그

# HTML 기본 태그

## : 시맨틱 태그

- ▶ 시맨틱 검색(Semantic Search) : 검색 로봇이 검색어 의미를 스스로 분석하고 추리해 원하는 정보를 더 정교하게 찾아주는 검색 방식(지능형 웹)
- ▶ 시맨틱 마크업(Semantic Markup) : HTML 태그를 적절히 사용하여 문서 안의 내용이 담고 있는 의미가 무엇인지 표현할 수 있도록 구조를 작성하는 것
- ▶ 시맨틱 태그(Semantic Tag) : 검색 로봇에 웹 페이지를 의미론적으로 해석할 수 있도록 도움을 주는 HTML5의 태그들

태그	설명
header	헤더
nav	내비게이션
aside	사이드에 위치하는 공간
section	여러 중심 내용을 감싸는 공간
article	글 내용이 들어가는 공간
footer	푸터



# HTML 기본 태그

## : 입력 양식(Form) 태그

- ▶ 입력 양식 : 사용자에게 정보를 입력 받는 공간
  - ▶ 기본적으로 **Form**은 웹 프론트(HTML)에서 서버로 데이터를 전송하기 위해 사용
- ▶ Form의 작성 : form 태그

```
<form method="POST"
      action="search.php">
  <input type="text" name="keyword" />
  <input type="submit" value="검색" />
</form>
```



- ▶ **method** 속성 : 서버로 데이터를 보낼 때 사용할 전송 방식(GET, POST)  
기본 값은 **GET**
- ▶ **action** 속성 : 폼으로 전송하는 데이터를 전달 받아 처리할 서버의 페이지 주소  
생략하면 현재 페이지로 데이터를 전송

# HTML 기본 태그

## : 입력 양식(Form) 태그

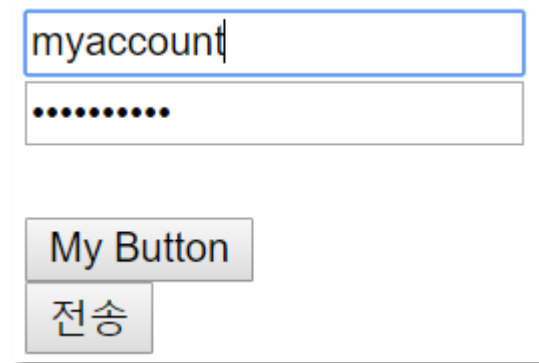
### ▶ input 태그

- ▶ 전송할 데이터를 입력하기 위한 입력 영역을 만드는 태그. form 태그 안쪽에 기술됨
- ▶ type 속성에 따라 다른 형태의 입력 영역이 표시

속성명	설명
type	입력 영역의 종류 (기본값: text)
name	전송할 데이터의 이름

type	설명
text	텍스트 입력 영역
password	암호 입력 영역(입력 값을 가림)
hidden	숨겨진 입력 영역
file	파일 정보 선택 영역
submit	전송 버튼
button	사용자 정의 버튼

```
<form method="POST" action="profile.jsp">  
<input type="text" name="userid" /><br/>  
<input type="password" name="password" /><br/>  
<input type="hidden" name="secret_key" /><br/>  
<input type="button" value="My Button" /><br/>  
<input type="submit" value="전송" />  
</form>
```



myaccount

.....

My Button

전송

# HTML 기본 태그

## : 입력 양식(Form) 태그

### ▶ input 태그 : checkbox와 radio

type	설명
checkbox	체크 박스 (중복 선택 가능)
radio	라디오 버튼 (단일 선택만 가능)

#### ▶ checkbox (type="checkbox")

```
<input type="checkbox" name="pet" value="dog" checked>개  
<input type="checkbox" name="pet" value="cat" checked>고양이  
<input type="checkbox" name="pet" value="bird">새
```

☒ 개 ☒ 고양이 ☐ 새

#### ▶ radio (type="radio")

```
<input type="radio" name="fruit" value="apple">사과  
<input type="radio" name="fruit" value="grape">포도  
<input type="radio" name="fruit" value="orange" checked>오렌지
```

☐ 사과 ☐ 포도 ☒ 오렌지

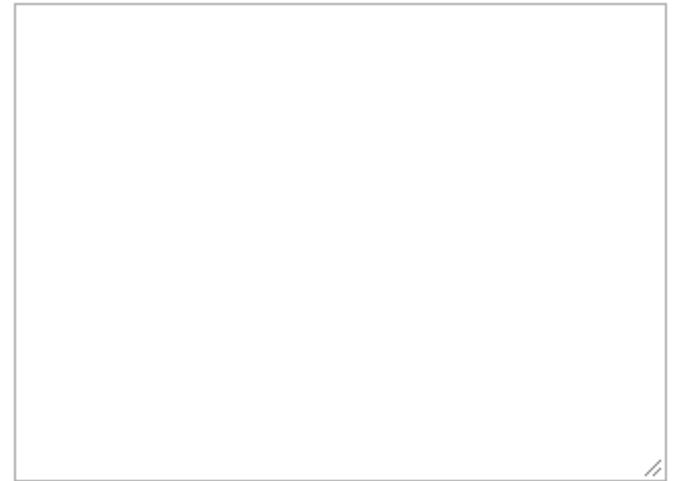
# HTML 기본 태그

## : 입력 양식(Form) 태그

### ▶ textarea 태그

- ▶ 장문의 텍스트(여러 줄)를 전송하고자 할 때 사용

속성명	설명
rows	열 수
cols	행 수



```
<textarea name="content" rows="10" cols="30"></textarea>
```

### ▶ select 태그

- ▶ 여러 개의 선택 옵션(option) 가운데서 하나를 선택

```
<select name="telecom">  
  <option value="S">SKT</option>  
  <option value="K">KT</option>  
  <option value="L">LGT</option>  
  <option value="E">기타</option>  
</select>
```



# HTML 기본 태그

## : 입력 양식(Form) 태그

- ▶ **fieldset** 태그
  - ▶ 입력 양식의 그룹 지정
- ▶ **legend** 태그
  - ▶ 입력 양식 그룹의 제목 지정
- ▶ **label** 태그
  - ▶ 입력 양식의 컨트롤 레이블(LABEL)을 표시

```
<form>  
  <fieldset>  
    <legend>로그인</legend>  
    <label for="account">계정</label>  
    <input type="text" name="account">  
    <label for="password">비밀번호</label>  
    <input type="submit" value="로그인">  
  </fieldset>  
</form>
```



로그인

계정  비밀번호



CSS3로 웹 페이지 스타일링하기

# CSS(Cascading Style Sheet)의 이해

## ▶ Cascading이 의미하는 것

- ▶ '계단형'의 의미로 스타일 적용시 규칙에 따라 우선 순위가 있고, 우선 순위에 따라 최종 렌더링될 스타일이 정해짐

## ▶ CSS가 필요한 이유

1. HTML에 직접 스타일을 적용함으로써 생기는 HTML 문서 자체의 무거움을 줄일 수 있다
2. 하나의 스타일 시트로 다수의 페이지에 같은 속성을 적용함으로써 작업 시간을 단축
3. 웹 표준의 원칙 : HTML 마크업은 구조를 잡는데 사용, CSS로 디자인을 결정

## ▶ 가장 기본적인 스타일의 적용 : HTML 태그에 style 속성으로 지정

```
<h1 style="color: yellow; background-color: black">Inline Styling</h1>
```

- ▶ 가장 쉬운 방법이지만,  
웹 표준 준수에는 가장 좋지 않은 방법



**Inline Styling**

# CSS(Cascading Style Sheet)의 이해

## ▶ 내부 스타일 시트

- ▶ HTML의 head 영역 안쪽에 <style> 태그를 이용, 스타일을 적용

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS: Example</title>
    <style>
      h1 {
        color: yellow;
        background-color: black;
      }
    </style>
  </head>
  <body>
    <h1>Page CSS</h1>
  </body>
</html>
```

```
h1 {
  color: yellow;
  background-color: black;
}
```

스타일 속성      스타일 값

- h1 태그를 찾아
- 색상을 노란 색으로
- 배경색을 검은 색으로 적용

**Page CSS**

# CSS(Cascading Style Sheet)의 이해

## ▶ 외부 스타일 시트

- ▶ 여러 사람이 협업하거나 프로젝트의 규모가 클 때는 스타일시트를 별도 외부 파일로 분리하는 것이 효율적
- ▶ 웹 표준의 측면에서 가장 추천하는 방식

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS: Example</title>
    <link rel="stylesheet"
          href="style.css" />
  </head>
  <body>
    <h1>External CSS</h1>
  </body>
</html>
```

style.css

```
@charset "utf-8";
h1 {
  color: yellow;
  background-color: black;
}
```

스타일 속성      스타일 값

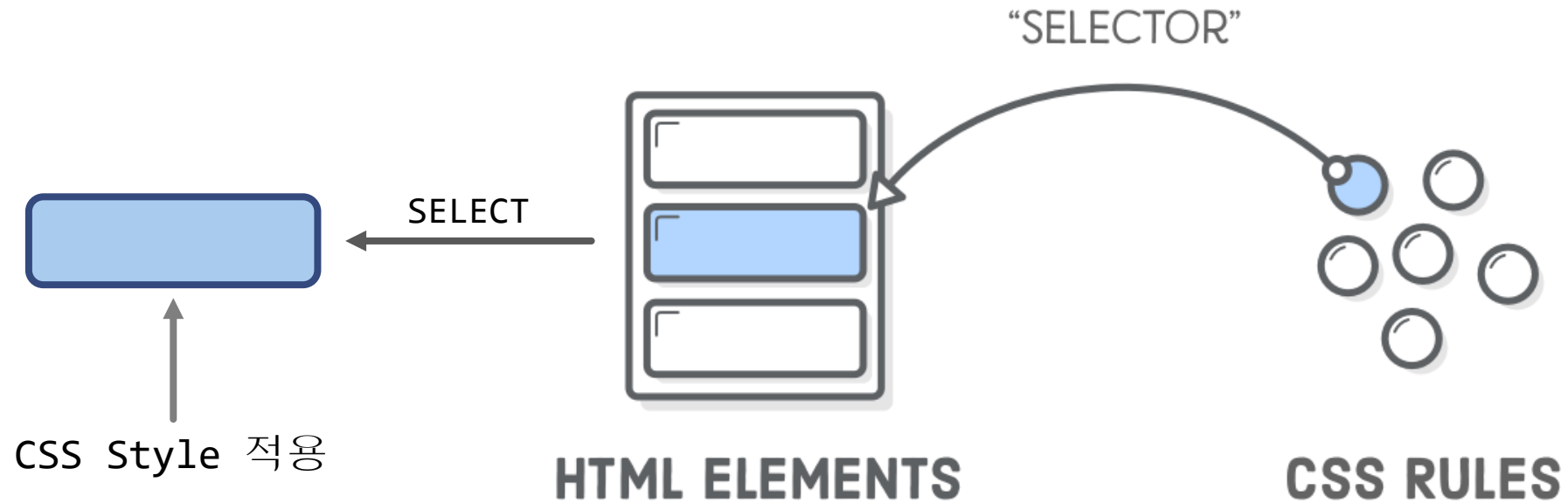
- h1 태그를 찾아
- 색상을 노란 색으로
- 배경색을 검은 색으로 적용

**External CSS**

# CSS3 선택자

## : 선택자 개요

- ▶ CSS 선택자(Selector) : 특정한 HTML 태그를 선택할 때 사용
  - ▶ 선택자를 사용, 특정 HTML 태그를 선택하면 조건에 맞는 태그에 스타일이나 스크립트를 적용할 수 있음
  - ▶ 선택자의 활용은 스타일시트 뿐 아니라 JavaScript에서도 사용하므로 웹 프론트엔드 개발시 꼭 이해해야 함



# CSS3 선택자

## : 기본 선택자

### ▶ 기본 선택자의 종류

종류	선택자	설명
전체 선택자	*	HTML 문서 내 모든 태그 선택
태그 선택자	태그명	HTML 페이지 내 모든 지정 태그
아이디 선택자	#아이디	지정한 아이디를 id 속성값으로 가지고 있는 태그
클래스 선택자	.클래스	지정한 클래스를 class 속성에 포함하고 있는 태그

### ▶ 전체 선택자

- ▶ HTML 문서 내 모든 태그를 선택하므로 페이지 전체에 공통적인 특성 값을 부여할 수 있다

- 페이지 내 모든 태그를 찾아
- 공백을 모두 제거
- 기본 콘텐츠 색상을 회색으로



```
<style>
  * {
    margin: 0;
    padding: 0;
    color: grey;
  }
</style>
```

# CSS3 선택자

## : 기본 선택자

### ▶ 태그 선택자

#### ▶ HTML 페이지 내 모든 지정 태그

- 페이지 내 모든 h1 태그를 찾아
- 글꼴을 볼드(bold)체로
- 색상을 빨간색(red)로

**Heading 1**



```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 {
        font-weight: bold;
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
  </body>
</html>
```

참고: 여러 개의 선택자를 동시에 지정할 때는 ,로 연결

```
h1, h2, h3, h4, h5, h6 {
  font-weight: bold;
  color: red;
}
```

# CSS3 선택자

## : 기본 선택자

### ▶ 아이디 선택자

- ▶ 지정한 아이디를 **id** 속성값으로 가지고 있는 태그
- ▶ 선택하고자 하는 아이디는 # 기호를 부여

- 사과
- 바나나
- 오렌지
- 포도



```
...  
<style>  
#apple { color: red; }  
#banana { color: yellow; }  
#orange { color: orange; }  
  
.selected {  
    text-decoration: underline;  
}  
</style>  
...  
<ul>  
    <li id="apple">사과</li>  
    <li id="banana">바나나</li>  
    <li id="orange">오렌지</li>  
    <li id="grape"  
        class="selected">포도</li>  
</ul>  
...
```

### ▶ 클래스 선택자

- ▶ 지정한 클래스를 **class** 속성에 포함하고 있는 태그
- ▶ 선택하고자 하는 클래스는 . 기호를 부여



# CSS3 선택자

## : 속성 선택자

### ▶ 속성 선택자

선택자	설명
선택자[속성명]	지정한 속성이 설정되어 있는 모든 태그
선택자[속성명=속성값]	지정한 속성이 지정한 속성 값으로 설정된 모든 태그

- input 태그 중 type 속성이 text인 태그 선택
- input 태그 중 type 속성이 password인 태그 선택



```
...  
<style>  
  input[type="text"] {  
    background-color: red;  
  }  
  input[type="password"] {  
    background-color: blue;  
  }  
</style>  
...  
<form>  
  <input type="text" />  
  <input type="password" />  
</form>  
...
```

# CSS3 선택자

## : 자식 선택자와 후손 선택자

### ▶ 자식 선택자와 후손 선택자

#### ▶ DOM Tree의 포함 관계를 이용한 하위 요소의 선택

종류	선택자	설명
후손 선택자	선택자A 선택자D	선택자A(조상)의 내부에 포함된 선택자D(후손)를 선택
자식 선택자	선택자P > 선택자C	선택자P(부모)의 바로 아래 노드들 중 선택자C(자식)를 선택

**d-sel > h1**

**d-sel > d-sel-child > h1**

**c-sel > h1**

선택되지 않음. Why? → **c-sel > c-sel-child > h1**



```
...
<style>
  #d-sel h1 {
    color: red;
  }
  #c-sel > h1 {
    color: red;
  }
</style>
...
<div id="d-sel">
  <h1>d-sel > h1</h1>
  <div id="d-sel-child">
    <h1>
      d-sel > d-sel-child > h1
    </h1>
  </div>
</div>
<div id="c-sel">
  <h1>c-sel > h1</h1>
  <div id="c-sel-child">
    <h1>
      c-sel > c-sel-child > h1
    </h1>
  </div>
</div>
...
```

# CSS3 선택자

## : 가상 클래스(Pseudo Class)

- ▶ **Pseudo Class** : 선택될 요소의 특별한 상태를 지정하는 선택자에 추가된 키워드
  - ▶ 가상 클래스는 단독으로 사용되지 않으며 선택자를 이용 선택된 요소의 상태 값을 의미함
- ▶ **반응 선택자** : 사용자의 반응으로 생성되는 특정한 상태를 선택

형태	설명
:active	사용자가 마우스로 클릭한 상태
:hover	사용자가 마우스 커서를 올린 상태
:link	링크 상태(미방문: A Tag Only)
:visited	링크 상태(이미 방문: A Tag Only)

```
...
<style>
  a:hover { color: red; }
  a:active { font-weight: bold; }
  a:link { color: blue; }
  a:visited { color: grey; }
</style>
...
<a href="#">반응 선택자</a>
...
```

# CSS3 선택자

## : 가상 클래스(Pseudo Class)

- ▶ 상태 선택자 : 입력 양식의 상태를 선택하기 위해 사용

형태	설명
:checked	checkbox, radio 태그가 체크된 상태
:focus	입력 양식에 초점이 맞춰져 있는 상태
:enabled	입력 가능한 input 태그의 상태
:disabled	입력 불가능한 input 태그의 상태

사용 가능 입력 양식:

사용 불가 입력 양식:

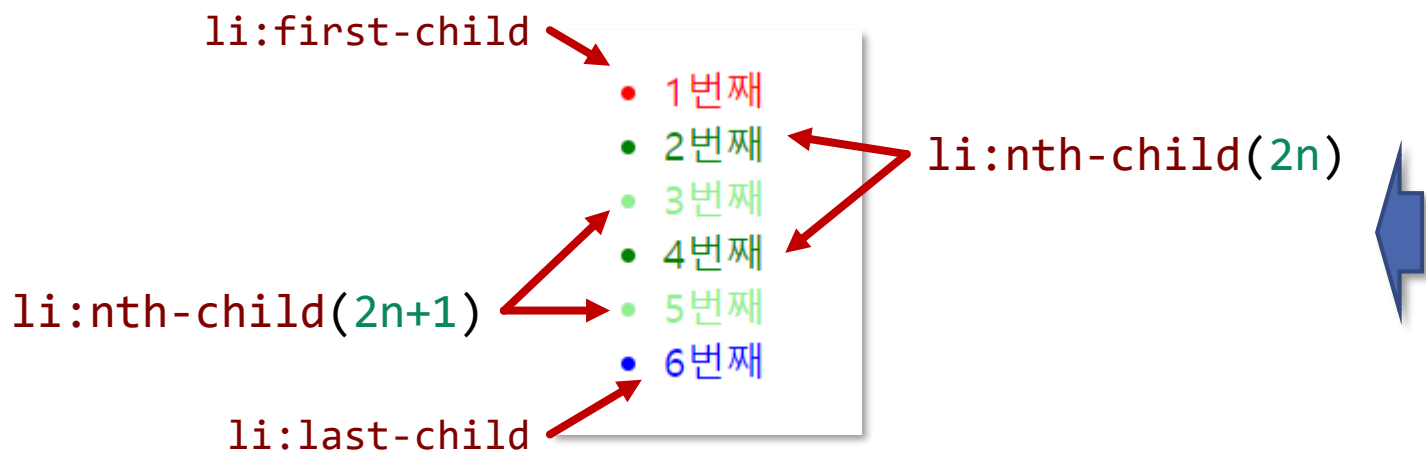
```
...
<style>
  input:enabled {
    background-color: white;
  }
  input:disabled {
    background-color: gray;
  }
  input:focus {
    background-color: orange;
  }
</style>
...
사용 가능 입력 양식:
  <input /><br>
사용 불가 입력 양식:
  <input disabled="disabled" />
...
```

# CSS3 선택자

## : 가상 클래스(Pseudo Class)

▶ 구조 선택자 : 특정 위치에 있는 태그를 선택할 때 사용

형태	설명
:first-child	형제 관계에서 첫 번째로 등장하는 태그 선택
:last-child	형제 관계에서 마지막 등장하는 태그 선택
:nth-child(수열)	형제 관계에서 앞에서 수열 번째로 등장하는 태그 선택
:nth-last-child(수열)	형제 관계에서 뒤에서 수열 번째로 등장하는 태그 선택



```
...
<style>
  li:nth-child(2n) {
    color: green;
  }
  li:nth-child(2n+1) {
    color: lightgreen;
  }
  li:first-child { color: red; }
  li:last-child { color: blue; }
</style>

...
<ul>
  <li>1번째 </li>
  <li>2번째 </li>
  <li>3번째 </li>
  <li>4번째 </li>
  <li>5번째 </li>
  <li>6번째 </li>
</ul>
...
```

# CSS3 단위

## : 키워드와 url

- ▶ 키워드(Keyword) : W3C에서 미리 정의한 단어
  - ▶ 키워드를 스타일 값으로 지정하면 키워드에 해당하는 스타일이 자동으로 적용
  - ▶ 예: 색상 키워드 ([https://developer.mozilla.org/ko/docs/Web/CSS/color\\_value](https://developer.mozilla.org/ko/docs/Web/CSS/color_value))
- ▶ url : 이미지, 글꼴 파일 등 네트워크상에 있는 리소스를 불러올 때 사용



```
...  
<style>  
body {  
    background-image: url('Landscape.jpg');  
}  
</style>  
...
```

# CSS3 단위

## : 크기

- ▶ CSS3에서 가장 많이 사용하게 되는 단위
  - ▶ %, em, px, cm, mm, inch 등 종류도 다양
  - ▶ 글꼴, 위치, 크기 등을 지정할 때 널리 사용
- ▶ px 단위
  - ▶ 브라우저의 기본 font-size 속성은 16px이며 절대값으로 지정

	단위	설명
상대 단위	%	백분율 단위
	em	배수 단위
절대 단위	px	픽셀 단위

```
<ul>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
</ul>
```

- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.

```
<style>
  li:nth-child(1) {}
  li:nth-child(2) { font-size: 16px; }
  li:nth-child(3) { font-size: 24px; }
  li:nth-child(4) { font-size: 32px; }
</style>
```

# CSS3 단위

## : 크기

- ▶ % 단위
  - ▶ 기본으로 설정된 크기를 기준으로 상대적인 단위를 지정
- ▶ em 단위
  - ▶ 기본으로 설정된 크기를 기준으로 배수를 이용 단위를 지정

	단위	설명
상대 단위	%	백분율 단위
	em	배수 단위
절대 단위	px	픽셀 단위

```
<ul>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
  <li>CSS로 크기를 지정해 봅니다.</li>
</ul>
```



```
<style>
  li:nth-child(1) {}
  li:nth-child(2) { font-size: 1.0em; }
  li:nth-child(3) { font-size: 1.5em; }
  li:nth-child(4) { font-size: 2.0em; }
</style>
```



```
<style>
  li:nth-child(1) {}
  li:nth-child(2) { font-size: 100%; }
  li:nth-child(3) { font-size: 150%; }
  li:nth-child(4) { font-size: 200%; }
</style>
```



- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.
- CSS로 크기를 지정해 봅니다.



# CSS3 단위

## : 색상

- ▶ 키워드로 지정하기
  - ▶ CSS3 색상 키워드를 이용: [https://developer.mozilla.org/ko/docs/Web/CSS/color\\_value](https://developer.mozilla.org/ko/docs/Web/CSS/color_value)
- ▶ Red, Green, Blue 색상의 조합
  - ▶ alpha 값은 0.0(완전 투명) ~ 1.0(완전 불투명)

형태	설명
<code>rgb(red, green, blue)</code>	red, green, blue 색상을 조합
<code>rgba(red, green, blue, alpha)</code>	red, green, blue 색상을 조합, alpha로 투명도 조정
<code>#000000</code>	16진수 표기법. red, green, blue 각각 2자리를 차지



#000000  
`rgb(0, 0, 0)`



#7F7F7F  
`rgb(127, 127, 127)`



#FFFFFF  
`rgb(255, 255, 255)`

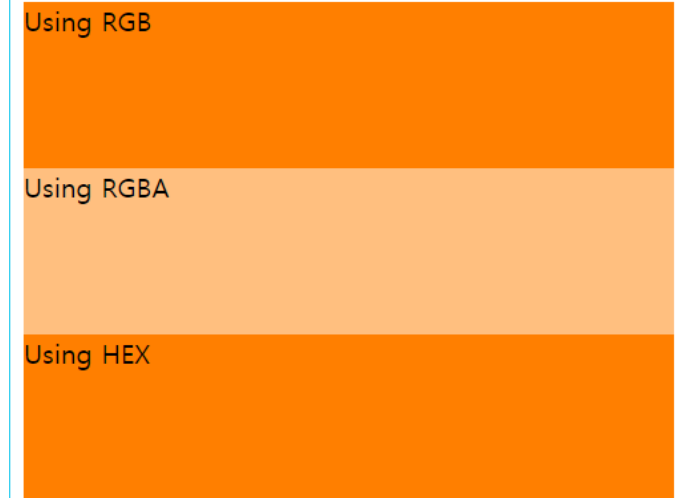


# CSS3 단위

## : 색상

- ▶ rgb, rgba, HEX 코드를 이용한 색상의 변경

```
<style>
  div#rgb {
    background-color: rgb(255, 127, 0);
  }
  div#rgba {
    background-color: rgba(255, 127, 0, 0.5);
  }
  div#hex {
    background-color: #FF7F00;
  }
</style>
...
<div id="rgb">Using RGB</div>
<div id="rgba">Using RGBA</div>
<div id="hex">Using HEX</div>
```

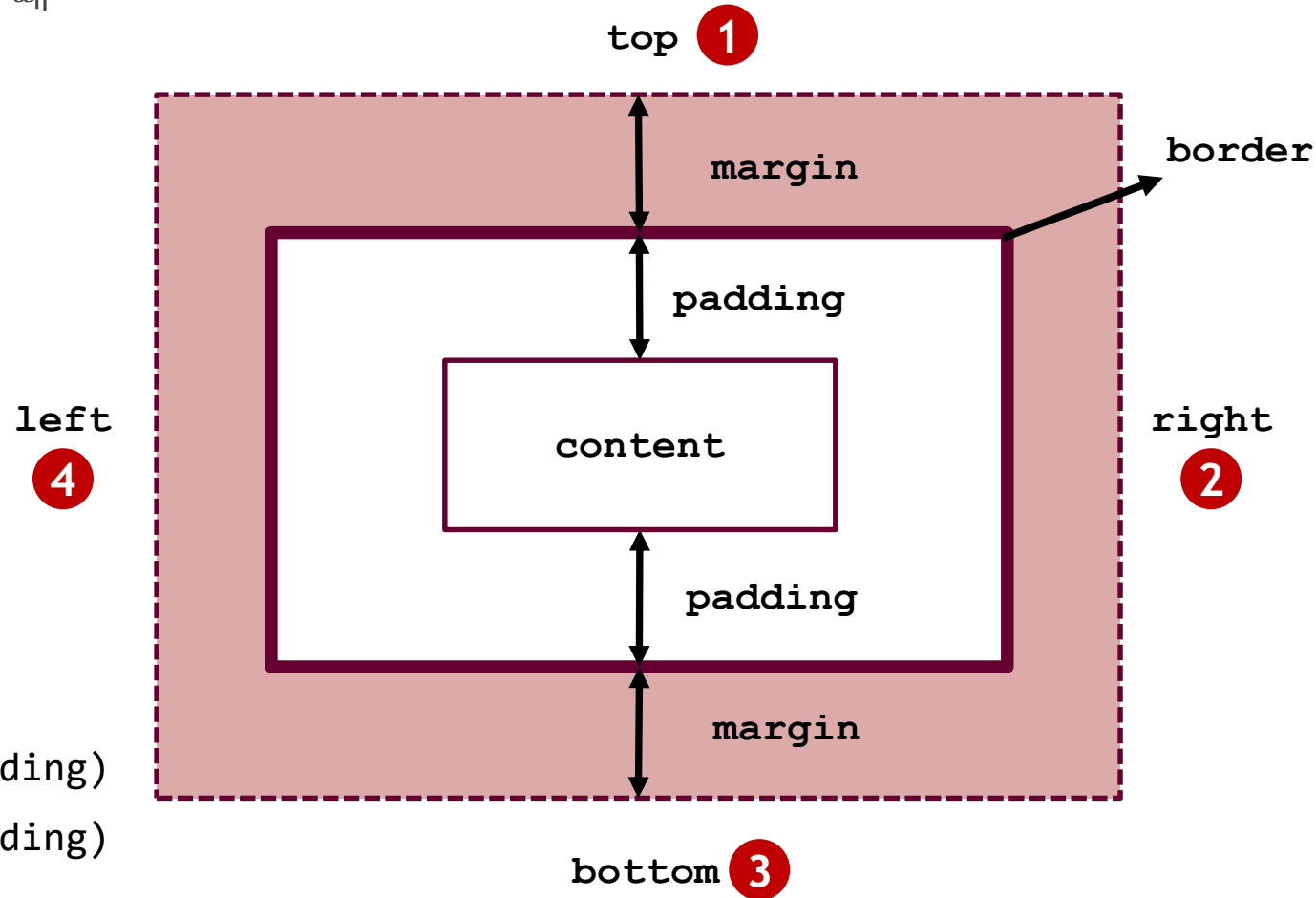


# CSS3 박스 모델

- ▶ 웹 문서를 이루는 요소들은 페이지에 배치될 때 박스 형태의 구조를 이룸
  - ▶ 박스 모델을 잘 이해해야 요소들을 원하는 위치에 적절하게 배치할 수 있음
- ▶ padding 영역 :  
요소의 콘텐츠와 테두리 사이의 영역  
(내부 공간 지정)
- ▶ margin 영역 :  
요소의 테두리와 외부 요소 사이의 영역  
(외부 공간 지정)

전체 너비 =  $\text{width} + 2 * (\text{margin} + \text{border} + \text{padding})$

전체 높이 =  $\text{height} + 2 * (\text{margin} + \text{border} + \text{padding})$



# CSS3 박스 모델

: margin, padding

- ▶ margin, padding에 값을 1개 입력할 경우

- ▶ 4방향 모두 적용

- ▶ 특정 방향의 margin과 padding 지정

접미어	설명
-left	왼쪽
-right	오른쪽
-top	위
-bottom	아래

```
padding-top: 10px;  
padding-right: 10px;  
padding-bottom: 10px;  
padding-left: 10px;
```

```
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
margin-left: 10px;
```

```
...  
<style>  
  div {  
    background-color: red;  
    padding: 20px;  
    margin: 20px;  
    text-align: center;  
  }  
</style>  
...  
<div>  
  Content 1  
</div>  
<div>  
  Content 2  
</div>  
...
```



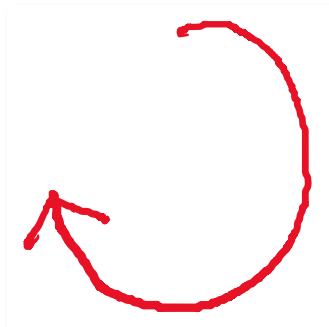
# CSS3 박스 모델

## : margin, padding

- ▶ margin, padding에 값을 4개 입력할 경우
  - ▶ -top을 시작으로 시계방향  
: 위 오른쪽 아래 왼쪽
  - ▶ 각 값은 공백으로 구분한다
- ▶ margin, padding에 값을 2개 입력할 경우
  - ▶ 첫 번째 값 : 위-아래
  - ▶ 두 번째 값 : 좌-우

```
...  
<style>  
  div {  
    background-color: red;  
    /* 위 오른쪽 아래 왼쪽 */  
    padding: 10px 20px 30px 100px;  
    margin: 0 10px 0 10px;  
    text-align: center;  
  }  
</style>  
...
```

```
...  
    /* 위-아래 좌-우 */  
    padding: 0 20px;  
    margin: 0 20px;  
...
```



# CSS3 박스 모델

## : border

### ▶ border : 테두리를 결정하는 속성

#### ▶ border를 결정하는 기본 스타일 3가지

속성	설명
border-width	테두리 두께
border-style	테두리 형태
border-color	테두리 색상

```
...  
<style>  
    div.box {  
        width: 200px; height: 200px;  
        background-color: black;  
        border-width: 5px;  
        border-style: dashed;  
        border-color: yellow;  
    }  
</style>  
...
```



#### ▶ border-width, border-style, border-color 속성을 한꺼번에 입력할 수 있음

```
...  
<style>  
    div.box {  
        ...  
        border: 5px dashed yellow;  
    }  
</style>  
...
```



# CSS3 박스 모델

## : border

- ▶ border-radius : 모서리를 둥글게 만드는 스타일

```
...  
<style>  
    img.round { border-radius: 20px; }  
</style>  
...  
  
...
```



- ▶ 모서리별 다른 둥글기를 적용할 수 있음

```
...  
<style>  
    img.round {  
        border-radius: 10px 20px 30px 40px;  
    }  
</style>  
...
```



# CSS3 박스 모델

## : display

### ▶ display

- ▶ HTML 요소가 공간을 점유하는 방식을 결정 (HTML 공간 분할 태그 항목 참조)
- ▶ display 속성의 키워드

키워드	설명
none	요소를 화면에서 감추기
block	요소를 블록 형식으로 표시
inline	요소를 인라인 형식으로 표시
inline-block	요소를 인라인 블록 형식으로 표시

### ▶ inline vs inline-block

	inline	inline-block
width, height 속성	X	O
margin 속성	좌우만 적용	상하좌우 모두 적용

### Inline elements

inline	inline	inline	inline
--------	--------	--------	--------

### Inline-block elements

inline-block	inline-block	inline-block	inline-block
--------------	--------------	--------------	--------------



# CSS3 박스 모델

## : visibility

### ▶ display vs visibility

- ▶ display는 공간 점유 방식을 결정하는 속성
- ▶ visibility는 시각적으로 표시 여부를 결정하는 속성 (visible, hidden)

```
#penguin {  
  display: none;  
}
```



```
  
<h1>Penguin</h1>
```



```
#penguin {  
  visibility: hidden;  
}
```

Penguin



Penguin

Penguin

# CSS3 박스 모델

## : background

### ▶ 배경을 지정하는 CSS 속성

속성	설명
background-image	배경 이미지 지정
background-size	배경 이미지의 크기 지정
background-repeat	배경 이미지의 반복 형태 지정
background-attachment	배경 이미지의 부착 형태 지정
background-position	배경 이미지의 위치 지정
background-color	배경 이미지의 색상 지정

### ▶ background-image : 배경 이미지 지정

```
...  
<style>  
  body {  
    background-image: url("Landscape.jpg");  
  }  
</style>  
...
```



CSS3에서는 여러 개의 이미지  
동시 적용도 가능(콤마로 구분)



# CSS3 박스 모델

## : background

### ▶ 배경을 지정하는 CSS 속성

- ▶ background-size : 배경 이미지의 크기

```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-size: 100%;
  }
</style>
```



- ▶ 두 개의 값으로 너비와 높이를 다르게 지정 가능



```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-size: 100% 150px;
  }
</style>
```

# CSS3 박스 모델

## : background

### ▶ 배경을 지정하는 CSS 속성

- ▶ background-repeat : 배경 이미지 반복 설정 (기본값 : repeat)

```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-repeat: no-repeat;
  }
</style>
```



- ▶ repeat-x, repeat-y로 값을 설정하면 x축 방향, y축 방향으로의 반복 여부만 지정도 가능



```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-repeat: repeat-y;
  }
</style>
```

# CSS3 박스 모델

## : background

### ▶ 배경을 지정하는 CSS 속성

#### ▶ background-repeat : 배경 이미지 반복 설정 (기본값 : repeat)

- ▶ fixed로 값을 지정하면 화면이 스크롤 되어도 배경은 스크롤되지 않음

```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-repeat: repeat-y;
    background-attachment: fixed;
  }
</style>
```



#### ▶ background-position: 배경 이미지가 표시될 이미지의 위치

```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: bottom;
  }
</style>
```



# CSS3 박스 모델

## : background

- ▶ 배경을 지정하는 CSS 속성
  - ▶ background: 배경 관련 속성을 일괄 지정



```
<style>
  body {
    background-image: url("image/Landscape.jpg");
    background-position: bottom;
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-color: lightgray;
  }
</style>
```

image position / size repeat attachment color

```
<style>
  body {
    background: url("image/Landscape.jpg") bottom / 100% no-repeat fixed lightgray;
  }
</style>
```





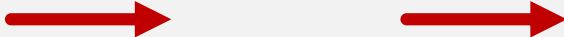
# CSS3 주요 속성

## : 글자 속성

### ▶ font-family 속성 : 글꼴을 지정하는 속성

- ▶ 한 단어로 된 글꼴은 따옴표를 사용하지 않지만, 두 단어 이상의 글꼴 이름은 따옴표 사용을 추천
- ▶ 사용자의 환경에 따라 개발시 사용한 글꼴이 사용자의 컴퓨터에 설치되어 있지 않을 가능성  
-> 복수 개의 글꼴(비슷한 형태의 글꼴)을 콤마(,)를 이용하여 명시(대체 글꼴)
- ▶ 다양한 사용자의 글꼴 상황을 개발자가 알 수 없으므로 font-family 속성의 맨 마지막에는 대부분의 컴퓨터에 탑재되어 있는 일반 글꼴(generic-family) 글꼴을 명시

font-family: "후보 글꼴1", "후보 글꼴2", **generic-family**  
sans-serif;



AaBbCc

Serif

AaBbCc

Sans-serif

AaBbCc

monospace

# CSS3 주요 속성

## : 글자 속성

- ▶ **font-size** 속성 : 글꼴의 크기
  - ▶ 브라우저의 기본 글꼴 크기는 16px
  - ▶ 크기 단위(px, em, %) 혹은 키워드(small, medium, large 등)로 지정 가능
- ▶ **font-style** 속성 : 글꼴의 기울기 조정
  - ▶ normal, italic, oblique 등의 키워드
- ▶ **font-weight** 속성 : 글꼴의 두께
  - ▶ 수치로 입력 : 400 = 일반 글자의 두께로 출력, 700 = 두껍게 출력.  
두께를 지원하지 않는 글꼴은 지정할 수 없음
  - ▶ 키워드로 입력 : normal, lighter, bold, bolder 등
- ▶ **text-align** 속성 : 글자의 정렬을 지정
  - ▶ 키워드로 입력 : left, right, center 등
- ▶ **text-decoration** 속성 : 선으로 글자를 꾸미는 속성
  - ▶ 키워드로 입력 : none, line-through, underline, overline 등



# CSS3 주요 속성

## : 위치(position) 속성

- ▶ 웹 프론트엔드 개발시 요소의 위치를 결정
  - ▶ 절대 위치 좌표 : 요소의 x 좌표와 y 좌표를 설정, 절대 위치를 지정
  - ▶ 상대 위치 좌표 : 요소를 입력한 순서 혹은 요소의 포함 관계에 의해 위치가 지정
  - ▶ 기본적으로 웹 브라우저는 상대 위치 좌표로 요소의 위치를 결정, 렌더링
- ▶ position 속성 : HTML 요소의 위치 설정 방법을 변경하는 속성

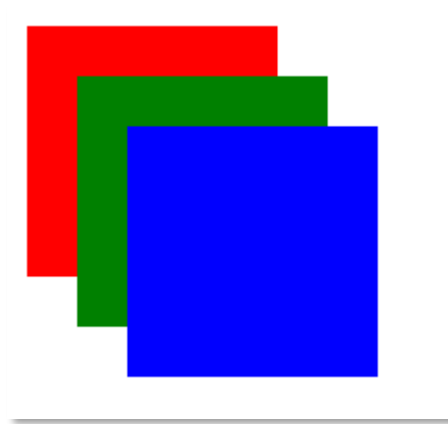
키워드	설명
static	상대 위치/좌표 설정
relative	초기 위치에서 상하좌우로 이동
absolute	절대 위치/좌표 설정
fixed	화면을 기준으로 절대 위치/좌표 설정

# CSS3 주요 속성

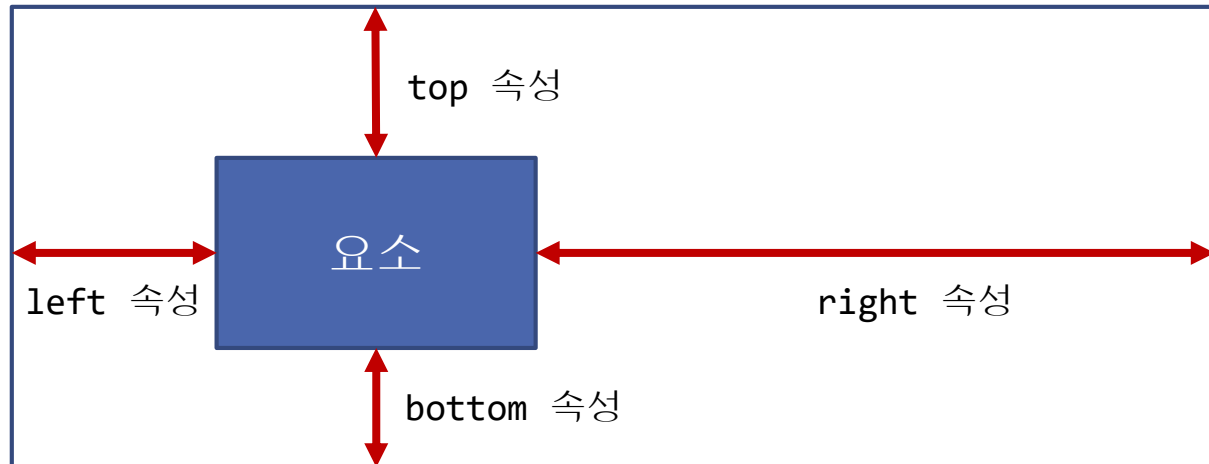
## : 위치(position) 속성

### ▶ absolute 좌표계

- ▶ left
- ▶ top
- ▶ right
- ▶ bottom



### ▶ z-index : 화면에 보이는 순서(z 축) 조정



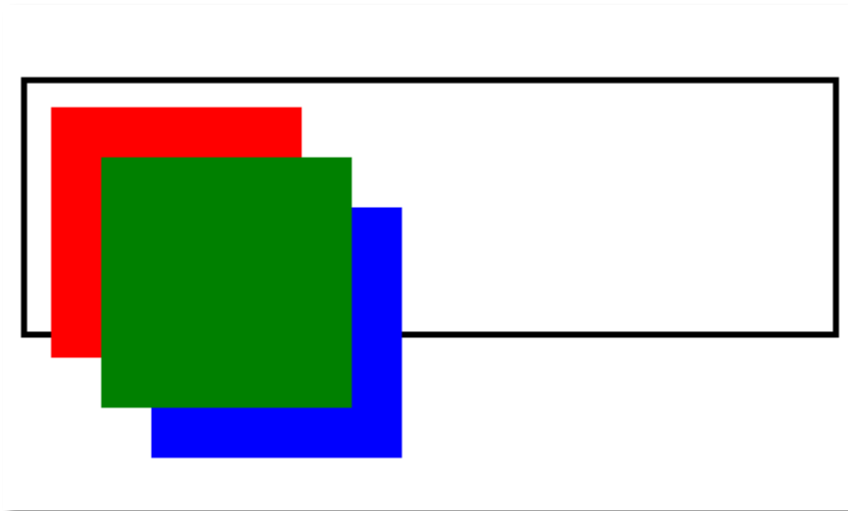
```
<style>
  .box {
    width: 100px; height: 100px;
    position: absolute;
  }
  .box:nth-child(1) {
    background-color: red;
    left: 10px; top: 10px;
  }
  .box:nth-child(2) {
    background-color: green;
    left: 50px; top: 50px;
  }
  .box:nth-child(3) {
    background-color: blue;
    left: 100px; top: 100px;
  }
</style>
...
<div>
  <div class="box"></div>
  <div class="box"></div>
  <div class="box"></div>
</div>
```

# CSS3 주요 속성

## : 위치(position) 속성

### ▶ absolute 좌표계 설정 팁

- ▶ 자손의 position 속성에 absolute 키워드를 적용하려면 조상 요소에 height 속성을 부여
- ▶ 자손의 position 속성에 absolute 키워드를 적용하려면 조상의 position 속성에 relative 키워드를 적용



```
<style>
    .container {
        position: relative;
        border: 2px solid;
        margin-top: 30px;
        height: 100px;
    }
    .box {
        width: 100px; height: 100px;
        position: absolute;
    }
</style>
...
<div class="container">
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
</div>
```

# CSS3 주요 속성

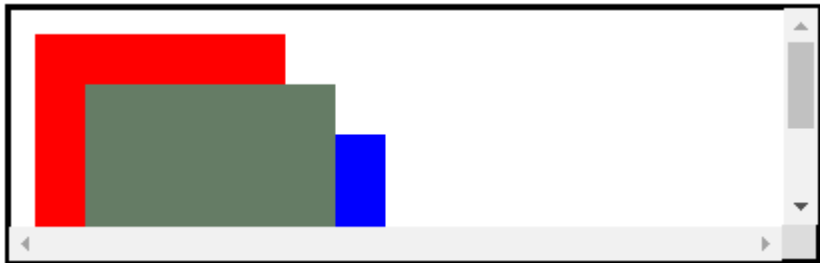
## : 위치(position) 속성

### ▶ absolute 좌표계

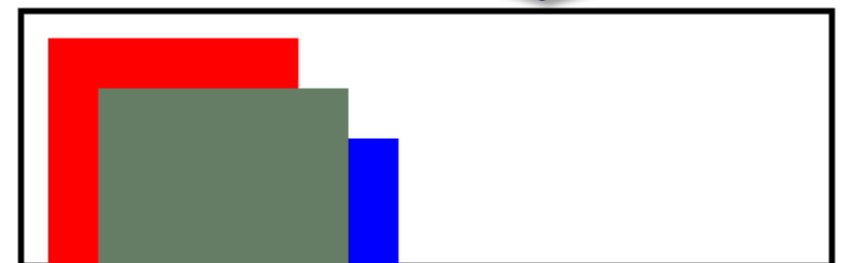
- ▶ **overflow** 속성 : 내부의 요소가 조상의 범위를 벗어날 때 처리하는 방법을 지정
  - ▶ **hidden** : 조상 영역을 벗어나는 부분 감춤
  - ▶ **scroll** : 조상 영역을 벗어나는 부분 스크롤 처리

```
<style>
    .box {
        width: 100px; height: 100px;
        position: absolute;
    }
</style>
...
<div class="container">
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
</div>
```

```
.container {
    position: relative;
    border: 2px solid;
    height: 100px;
    overflow: scroll;
}
```



```
.container {
    position: relative;
    border: 2px solid;
    height: 100px;
    overflow: hidden;
}
```



# CSS3 주요 속성

## : float 속성

### ▶ float

- ▶ '부유'라는 의미로 박스 모델을 물에 띄워 보낸다는 의미
- ▶ 현대 웹 프론트엔드 작업에서는 레이아웃을 잡을 때 유용하게 사용

키워드	설명
left	요소를 띄워 왼쪽으로 보내기
right	요소를 띄워 오른쪽으로 보내기

```
<style>
  .menu > li {
    float: right;
    ...
  }
</style>
```

메뉴 3

메뉴 2

메뉴 1

```
...
<ul class="menu">
  <li>메뉴 1</li>
  <li>메뉴 2</li>
  <li>메뉴 3</li>
</ul>
...
```

```
<style>
  .menu > li {
    float: left;
    ...
  }
</style>
```

메뉴 1

메뉴 2

메뉴 3

# CSS3 주요 속성

## : float 속성

### ▶ float

- ▶ 이미지와 본문이 자연스럽게 녹아 드는 레이아웃 구성 예제

```
<style>
  img {
    float: right;
    ...
  }
</style>
```

```
...

<p>Lorem ipsum dolor sit amet,
consectetur ...</p>
...
```

```
<style>
  img {
    float: left;
    ...
  }
</style>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Morbi tincidunt augue interdum velit euismod. Ut eu sem integer vitae justo eget magna. Cum sociis natoque penatibus et magnis. Volutpat commodo sed egestas egestas fringilla phasellus faucibus. Eleifend donec pretium vulputate sapien nec sagittis aliquam. Porttitor massa id neque aliquam vestibulum morbi blandit. Nisi porta lorem mollis aliquam. Sed tempus urna et pharetra pharetra. Vulputate ut pharetra sit amet aliquam id diam maecenas ultricies. Porttitor eget dolor morbi non arcu risus quis varius. Aenean vel elit scelerisque mauris.



pretium vulputate sapien nec sagittis aliquam. Porttitor massa id neque aliquam vestibulum morbi blandit. Nisi porta lorem mollis aliquam. Sed tempus urna et pharetra pharetra. Vulputate ut pharetra sit amet aliquam id diam maecenas ultricies. Porttitor eget dolor morbi non arcu risus quis varius. Aenean vel elit scelerisque mauris.