

01 인터넷과 웹 시작

■ 인터넷 vs. 웹

- 인터넷은 전 세계를 연결하는 국제 정보 통신망으로, 컴퓨터나 스마트폰 같은 디지털기기로 연결되어 사람들이 정보를 공유할 수 있는 공간
- 웹은 인터넷 공간에서 제공하는 서비스

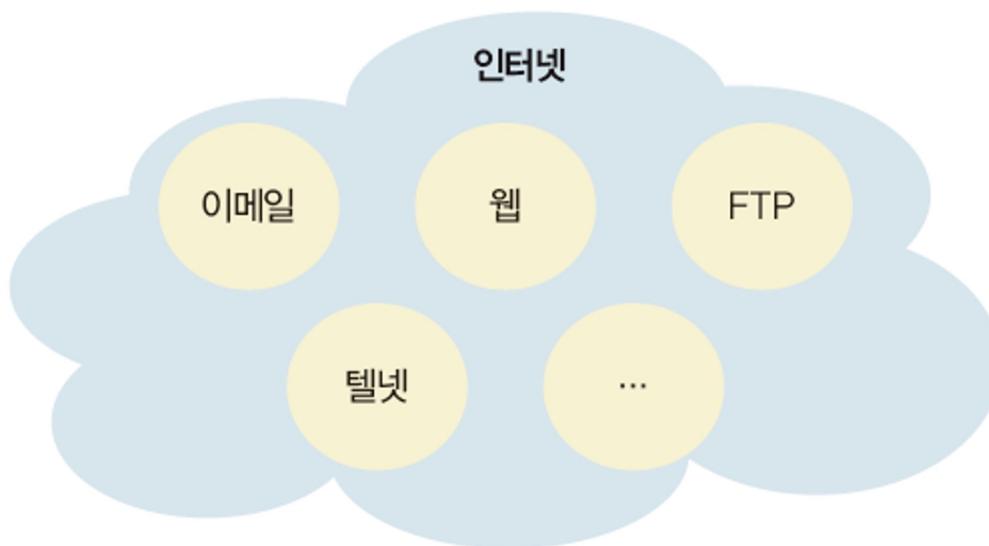


그림 1-1 인터넷 ≠ 웹

01 인터넷과 웹 시작

■ 인터넷의 시작

- 미국의 신기술을 연구하는 ARPA에서 현재 웹의 모태가 되는 아파넷 ARPANET을 1969년에 개발

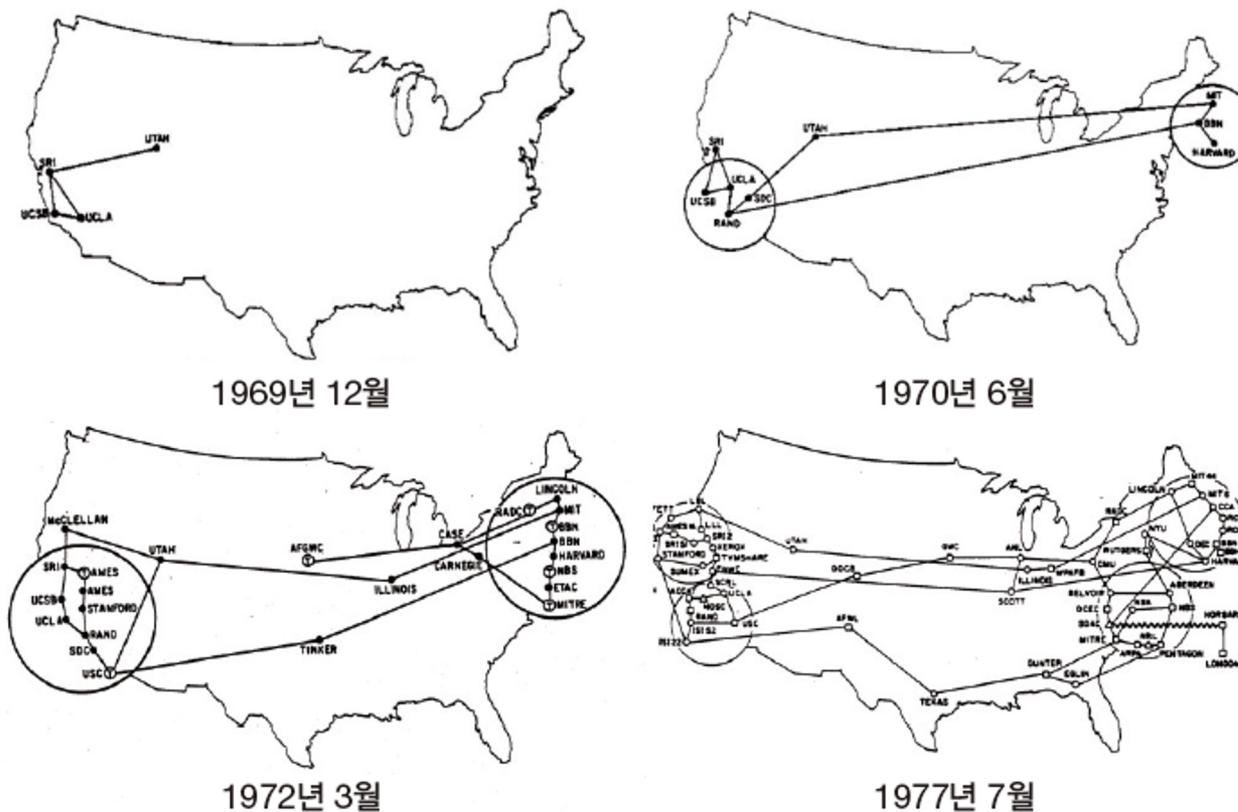


그림 1-2 인터넷 시초인 아파넷의 연결 변화

01 인터넷과 웹 시작

- 팀 버너스 리
 - 최초 웹 개발자
 - W3CWorld Wide Web Consortium 창설
 - HTML 표준을 비롯한 웹 표준안을 제작, 제안하는 일을 하는 국제적인 웹 표준화 단체

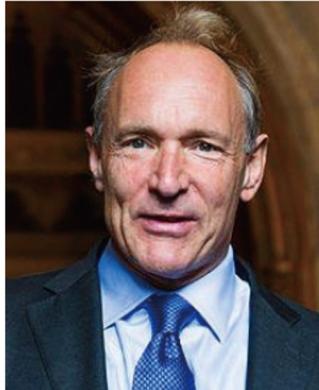


그림 1-3 팀 버너스 리와 월드 와이드 웹 컨소시엄

표 1-1 HTML 버전

버전	발표 연도
HTML1	1991년 10월
HTML2	1995년 11월
HTML3	1997년 1월
HTML4	1997년 12월
HTML5	2014년 10월

02 웹 브라우저 전쟁과 웹 표준

■ 웹 브라우저 전쟁의 시작

- NCSA는 1993년 모자이크라는 GUI 웹 브라우저 최초로 발표
- 넷스케이프 커뮤니케이션(마크 안데르센)은 넷스케이프 웹 브라우저 발표
- 마이크로소프트는 인터넷 익스플로러 발표 후 점유율을 높이기 위해 윈도우 운영체제에 강제로 설치
- 이에 넷스케이프는 넷스케이프의 소스 코드를 공개하고 모질라 재단을 설립
- 아메리카 온라인 AOL : 넷스케이프 커뮤니케이션을 인수, 점차 하락 후 붕괴

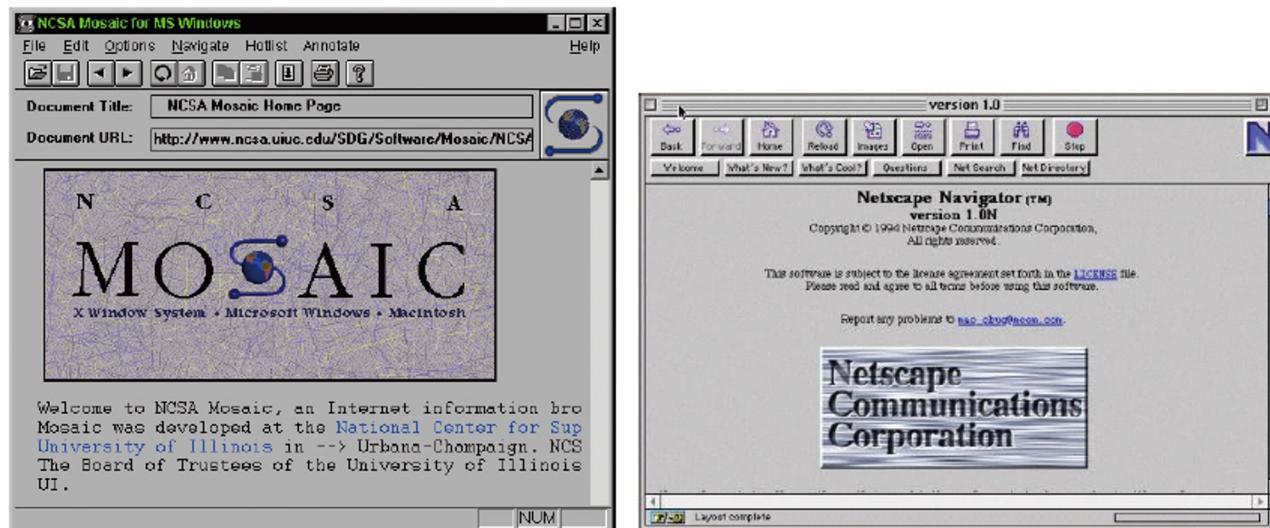


그림 1-4 모자이크와 넷스케이프 웹 브라우저

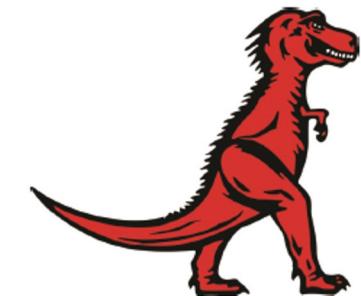


그림 1-5 모질라 재단의 캐릭터

02 웹 브라우저 전쟁과 웹 표준

■ 플러그인

- 웹 브라우저와 연동되는 프로그램을 사용자의 PC에 추가로 설치해 웹 브라우저의 기능을 확장하는 방법
- 어도비 플래시 Adobe Flash나 마이크로소프트 액티브X ActiveX 등
- 2000년 초반에는 액티브X 기반으로 웹 응용 프로그램 제작, 플래시 기반으로 애니메이션 제작



그림 1-6 플래시로 제작한 졸라맨

02 웹 브라우저 전쟁과 웹 표준

■ 웹 2.0 시대

- 서로 다른 사용자가 함께 새로운 콘텐츠를 창조할 수 있는 시대
- 유튜브, 위키피디아, 깃허브 등

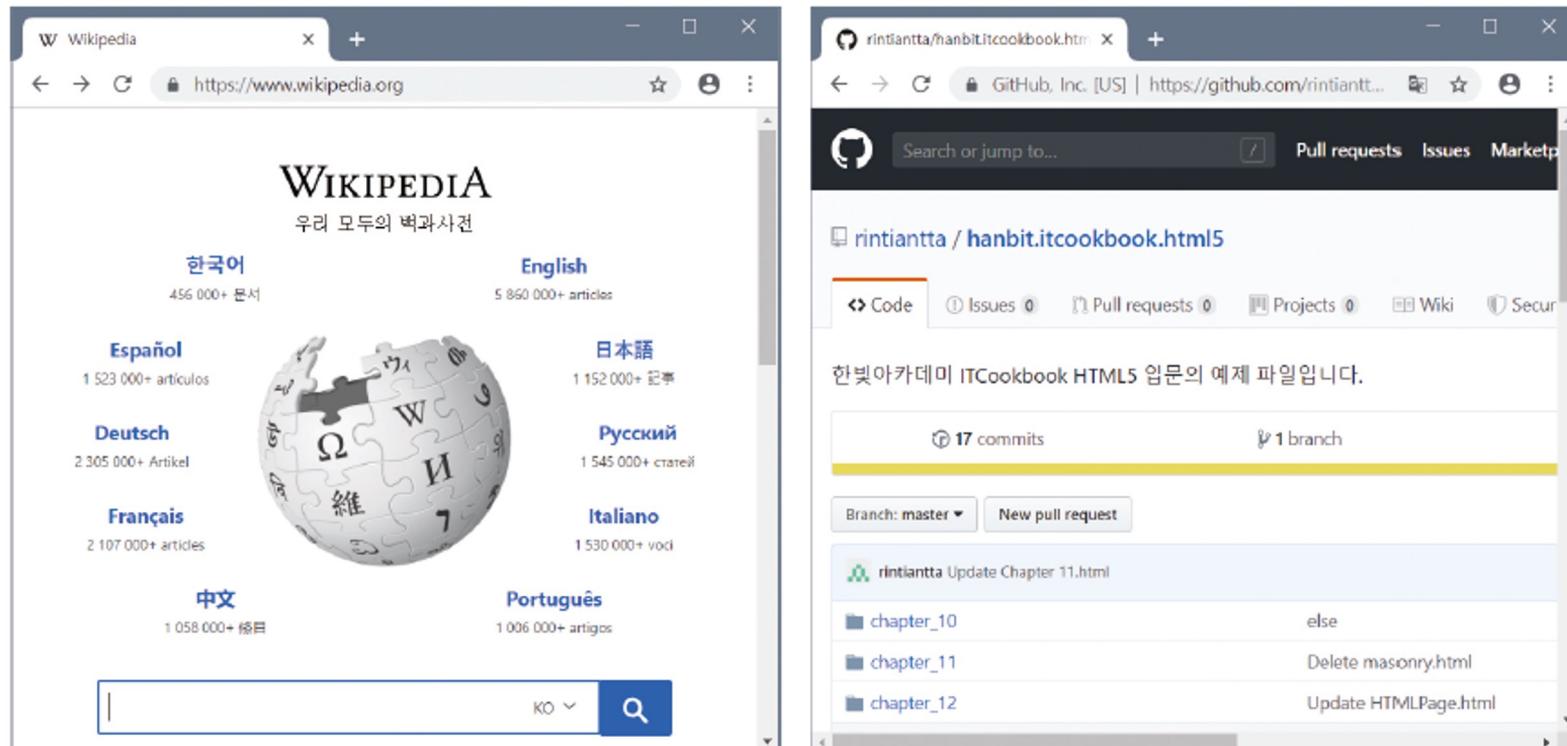


그림 1-7 위키피디아와 깃허브

02 웹 브라우저 전쟁과 웹 표준

■ 웹 브라우저와 웹 표준

- 인터넷 익스플로러가 전 세계적으로 점유율을 높이면서 액티브X 같은 플러그인을 삽입해 웹 사이트가 점점 무거워지고 있음
- 2004년 W3C 회의에서 모질라 재단과 오페라 소프트웨어가 새로운 HTML 표준을 제안하지만 거절 당함
- 웹 브라우저 제공 기업(애플, 모질라, 오페라 소프트웨어)에서 2004년 6월 새로운 웹 표준 기관 WHATWG 설립



WHATWG

Web Application 1.0 표준 작성

W3C는 웹 표준으로 Web Application 1.0 표준 채택, HTML5 표준으로 변경

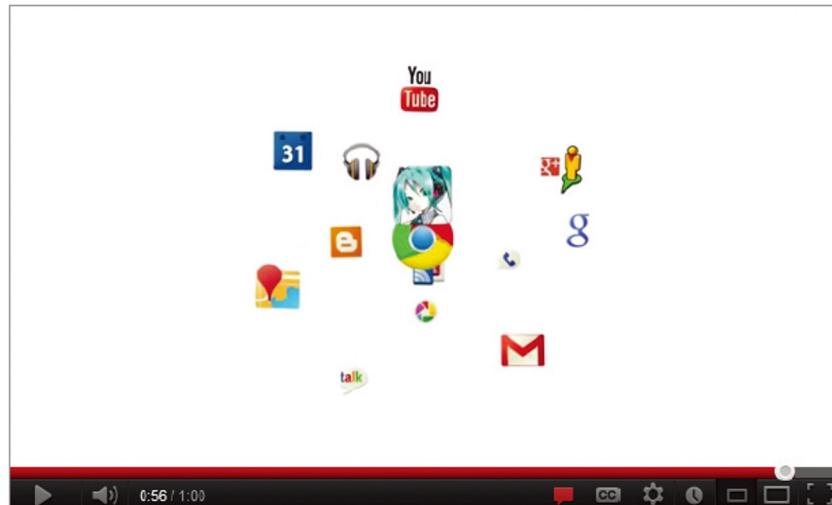
인터넷 익스플로러만 최신 표준을 지원하지 못하는 현상 초래
모든 웹 브라우저가 빠른 속도로 업데이트되고 있음

그림 1-8 WHATWG

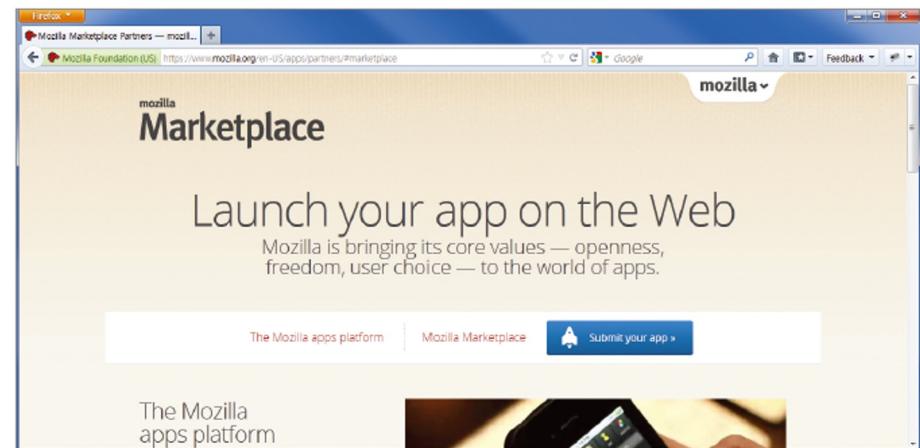
02 웹 브라우저 전쟁과 웹 표준

■ 웹 브라우저의 발달

- WHATWG는 Web Application 1.0 표준 작성
- W3C는 웹 표준으로 Web Application 1.0 표준 채택, HTML5 표준으로 변경
- 인터넷 익스플로러만 최신 표준을 지원하지 못하는 현상 초래
- 모든 웹 브라우저가 빠른 속도로 업데이트되고 다양한 웹 브라우저 마케팅 등장



(a) 구글 크롬 일본 광고



(b) 모질라 마켓플레이스

그림 1-9 다양한 웹 브라우저 마케팅

02 웹 브라우저 전쟁과 웹 표준

■ 웹 표준과 웹 브라우저 타임 라인

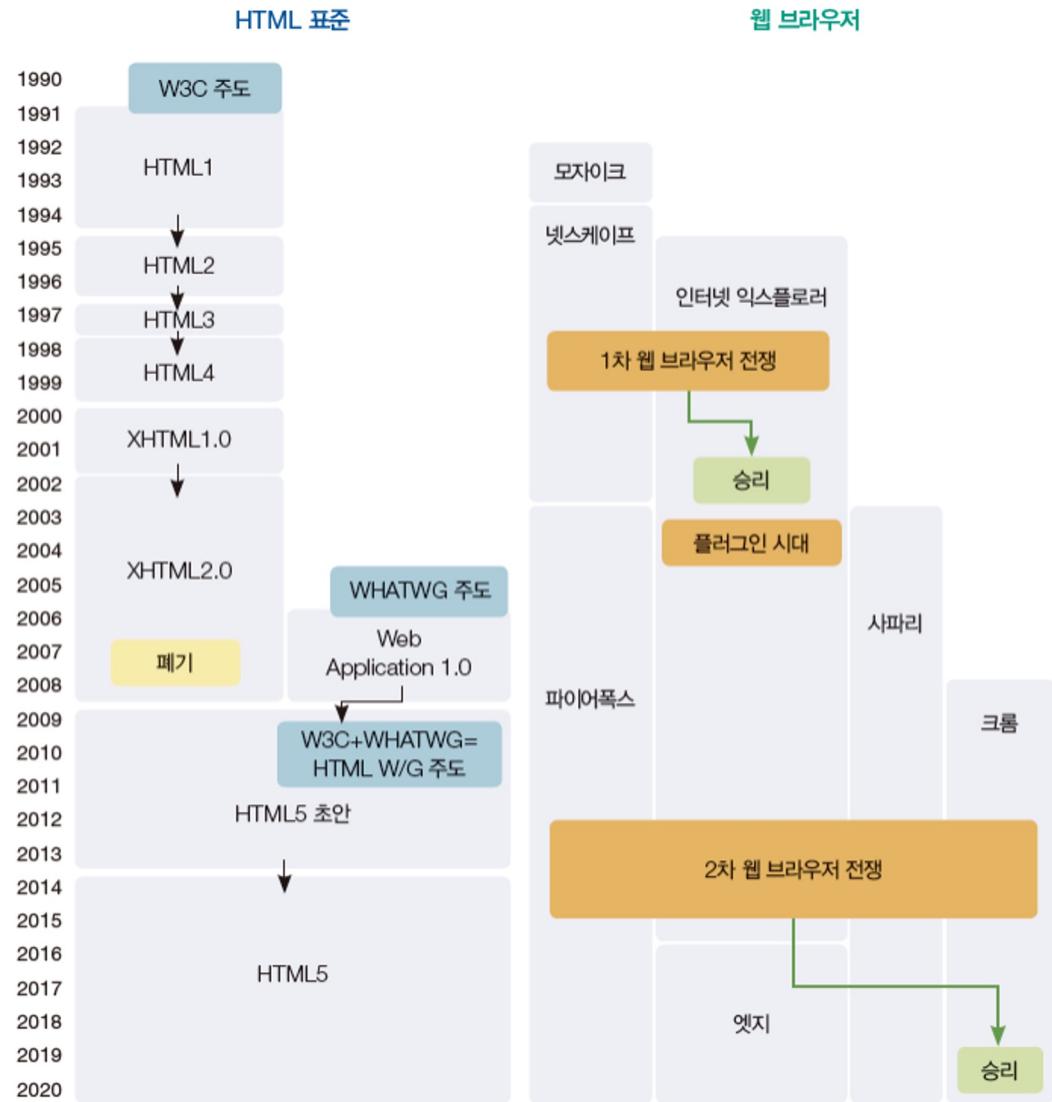


그림 1-10 웹 표준과 웹 브라우저 타임 라인 ※ HTML3와 HTML4는 W3C 권고안에 오른 날짜를 기준으로 표기

03 웹 동작

- 웹(Web): 요청과 응답 과정 (예 : 치킨 주문)



그림 1-11 요청과 응답 과정

- URL: 웹에서 어떤 대상을 구분하는 주소

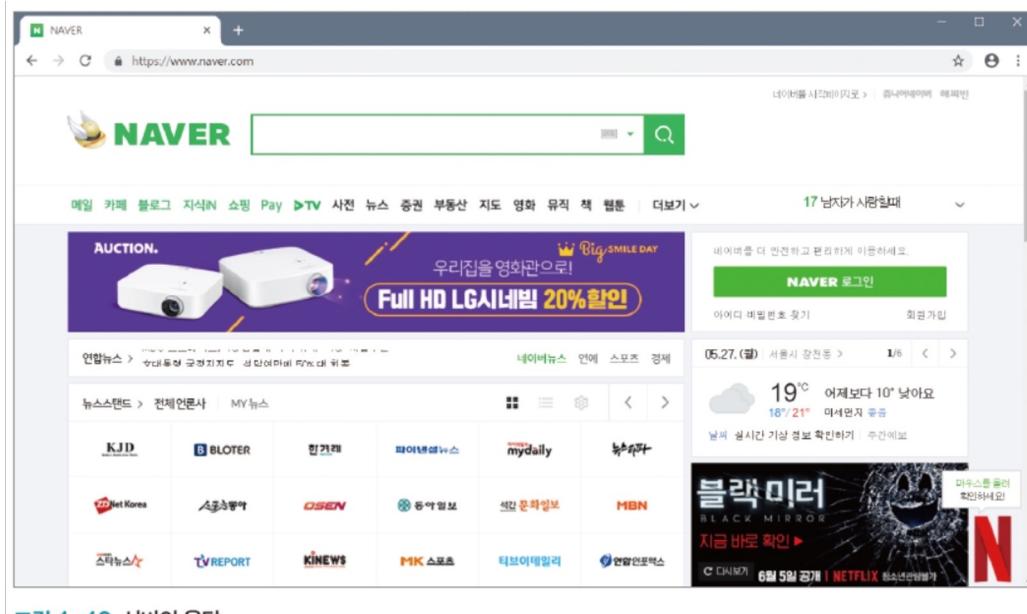


그림 1-12 서버의 응답

03 웹 동작

■ 서버와 클라이언트



■ 서버 프로그램(백엔드 프로그램)

- 자바, C#, 루비, 파이썬, 자바스크립트
 - 웹 프레임워크(ASP.NET, JSP, PHP 등)
 - MVC 프레임워크(ASP.NET MVC, Spring MVC, Ruby on Rails 등)
 - 비동기 프레임워크(Node.js Express, Jetty)

■ 클라이언트 프로그램(프론트엔드 프로그램)

- HTML, CSS, 자바스크립트로 개발
 - 구글 문서



그림 1-14 구글 문서 도구

04 웹 표준 기술과 HTML5 주요 기능

■ 웹 표준 기술



그림 1-15 웹 표준 기술

■ 큰 의미

- 웹 표준 기술을 총칭
- CSS3와 자바스크립트를 모두 포함

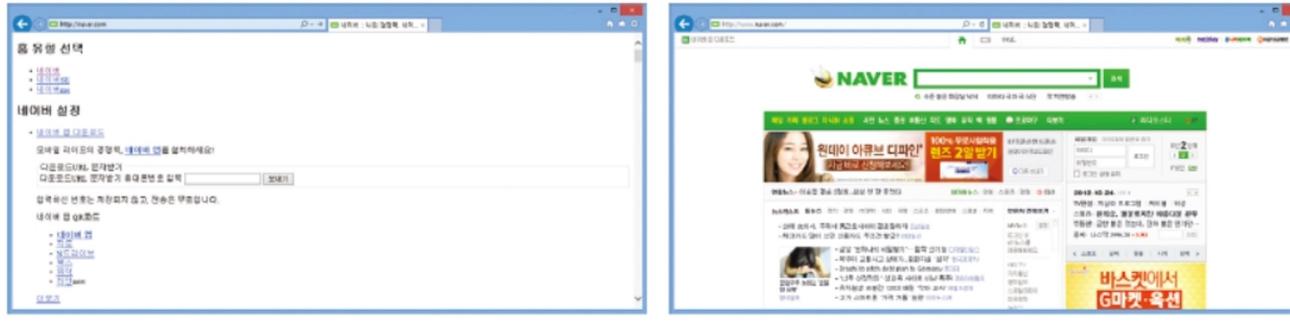
■ 작은 의미

- 웹 페이지를 구성하는 HTML 마크업 언어 그 자체

04 웹 표준 기술과 HTML5 주요 기능

■ CSS Cascading Style Sheets

- HTML 페이지에 스타일을 지정하는 스타일시트를 작성할 때 사용하는 언어
- 현대 웹 페이지에서 매우 중요한 역할 수행



(a) 스타일시트 적용 전
(b) 스타일시트 적용 후

그림 1-16 네이버 메인 페이지의 스타일시트 적용 전후 모습

■ 자바스크립트

- 표준 명칭: ECMAScript
- HTML 페이지에서 사용자 반응 등을 처리하는 스크립트를 작성하는 언어
- 클라이언트 웹 개발을 하기 위해 만들었지만, 현대에는 서버는 물론 로봇 개발에도 사용



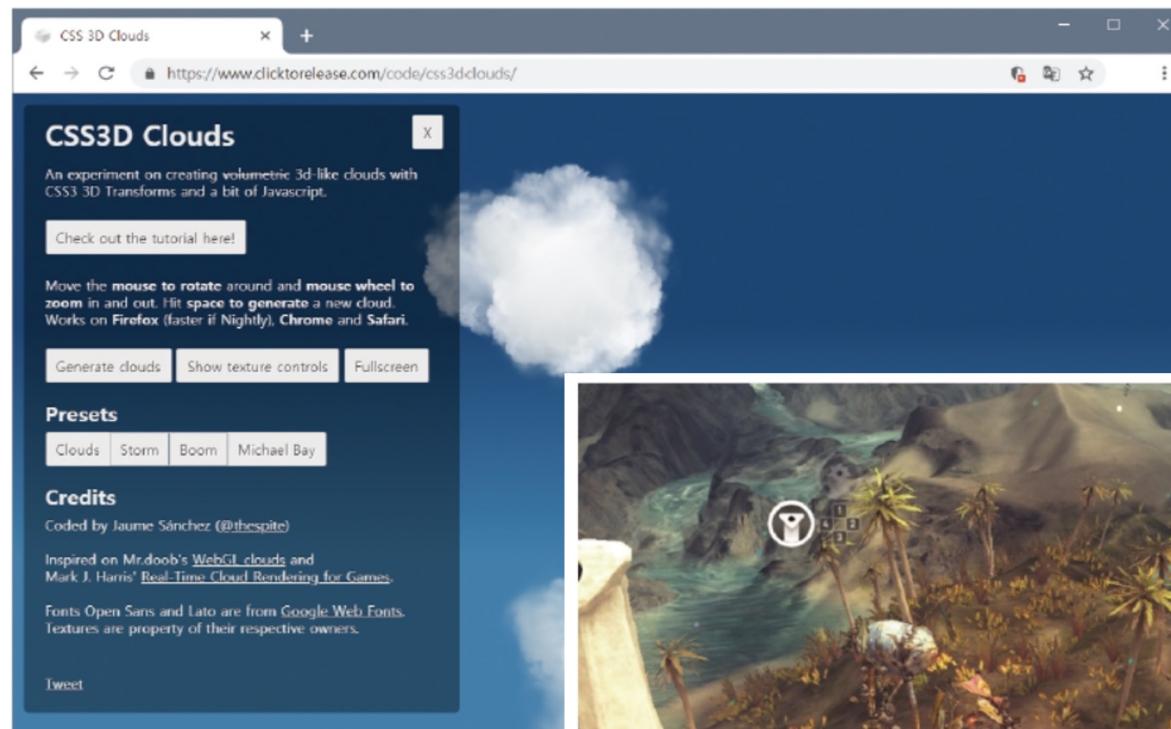
그림 1-17 유럽 컴퓨터제조협회(ECMA)

04 웹 표준 기술과 HTML5 주요 기능

- HTML5 주요 기능
 - 멀티미디어 기능
 - 플러그인 도움 없이도 스스로 음악과 동영상을 재생
 - 그래픽 기능
 - 2차원 그래픽 구현
 - 1 HTML 태그를 사용해 2차원 벡터 그래픽 구현
 - 2 자바스크립트 캔버스를 사용해 2차원 래스터 그래픽 구현
 - 3차원 그래픽 구현
 - 1 CSS3를 사용해 3차원 그래픽 구현
 - 2 자바스크립트 WebGL을 사용해 3차원 그래픽 구현

04 웹 표준 기술과 HTML5 주요 기능

- CSS3를 사용해 3차원 그래픽을 구현한 예



(a) CSS3를 사용한 구현 예: CSS3D Clouds(<https://www.clicktorelease.com/code/css3dclouds/>)



(b) WebGL을 사용한 구현 예: 프롬 더스트(From Dust)

그림 1-18 HTML5로 3차원 그래픽을 구현한 예

04 웹 표준 기술과 HTML5 주요 기능

- HTML5 주요 기능

- 통신 기능

- 서버와 실시간으로 쌍방향 통신을 수행
 - 웹에서 실시간 채팅, 다수의 사람과 온라인 게임, 고객 센터 문의-답변 가능

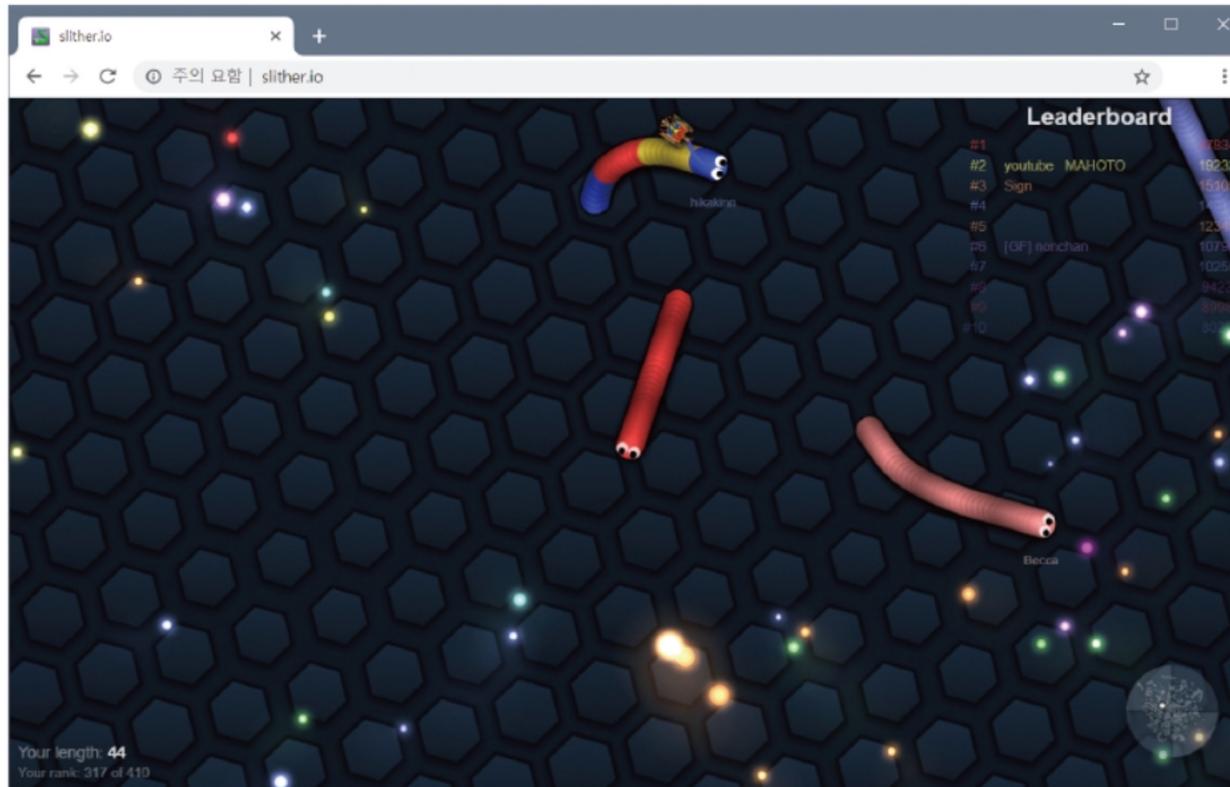


그림 1-19 지렁이 키우기 게임(slither.io)

04 웹 표준 기술과 HTML5 주요 기능

- HTML5 주요 기능
 - 장치 접근
 - 장치에 접근해 정보와 기능을 사용
 - 예) 배터리 잔량, 현재 위치, 알람, 스마트폰 GPS로 현재 위치 확인
 - 오프라인 및 저장소
 - 인터넷이 연결되지 않은 상태에서도 응용 프로그램을 동작 가능



그림 1-20 구글 크롬 OS

04 웹 표준 기술과 HTML5 주요 기능

- HTML5 주요 기능
 - 시맨틱
 - 검색 엔진 같은 프로그램이 정보를 분석, 자료를 검색 및 처리해서 제공하는 지능형 웹
 - CSS3 스타일시트
 - CSS3 스타일시트를 사용하면 3차원 변환과 애니메이션 효과 적용 가능
 - 웹의 성능 극대화 및 통합
 - HTML5의 추가 기능으로 기존에 이용하던 웹의 성능을 극대화할 수 있음
 - 웹 워커를 이용하면 사용자 화면을 멈추지 않으면서 연산 도 처리 가능
 - HTML5 표준을 적용한 웹 브라우저를 사용해 간단한 워드 작업이나 게임 가능

05 HTML5를 공부하면 좋은 이유

- 애플리케이션 수준의 웹 페이지를 만들 수 있다.

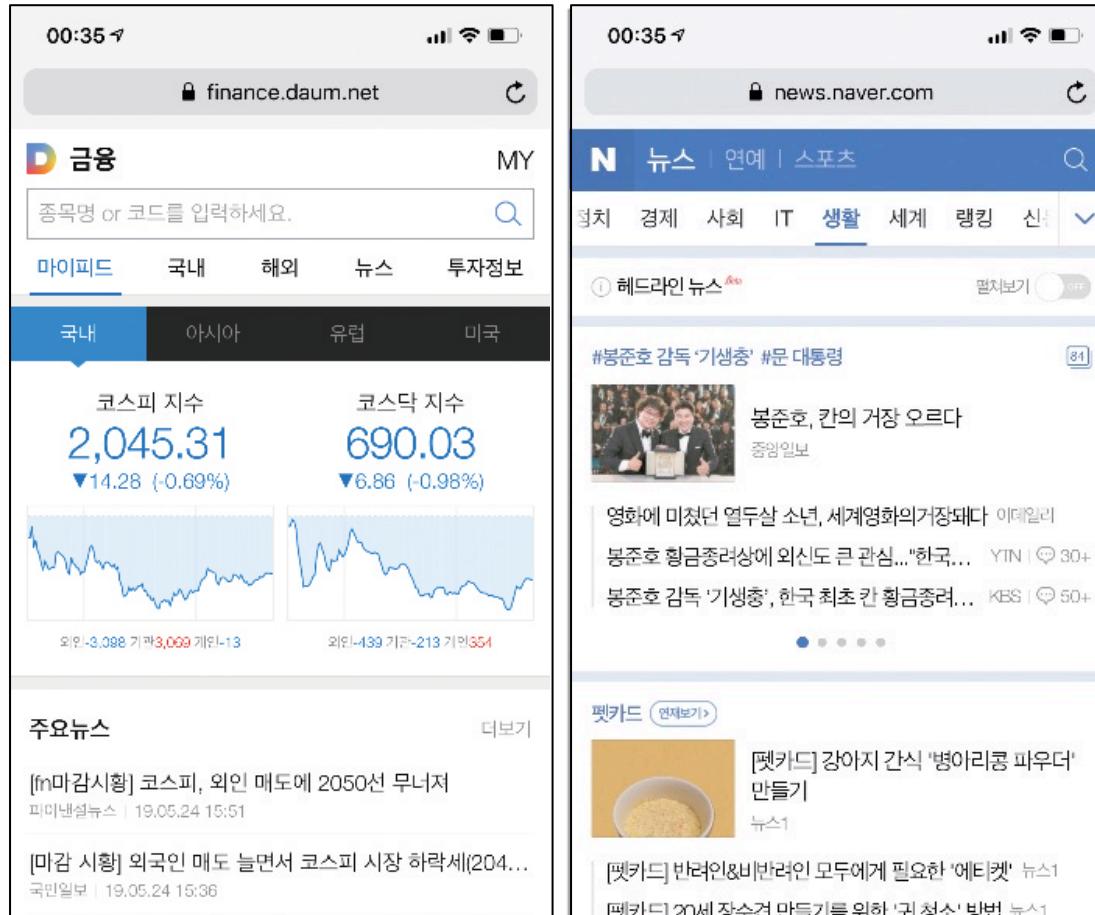


그림 1-21 HTML5로 만든 애플리케이션 수준의 웹 페이지: 다음 금융과 네이버 뉴스

05 HTML5를 공부하면 좋은 이유

- 데스크톱 애플리케이션을 만들 수 있다.
- 일렉트론 - HTML5 기반의 데스크톱 애플리케이션 개발 엔진

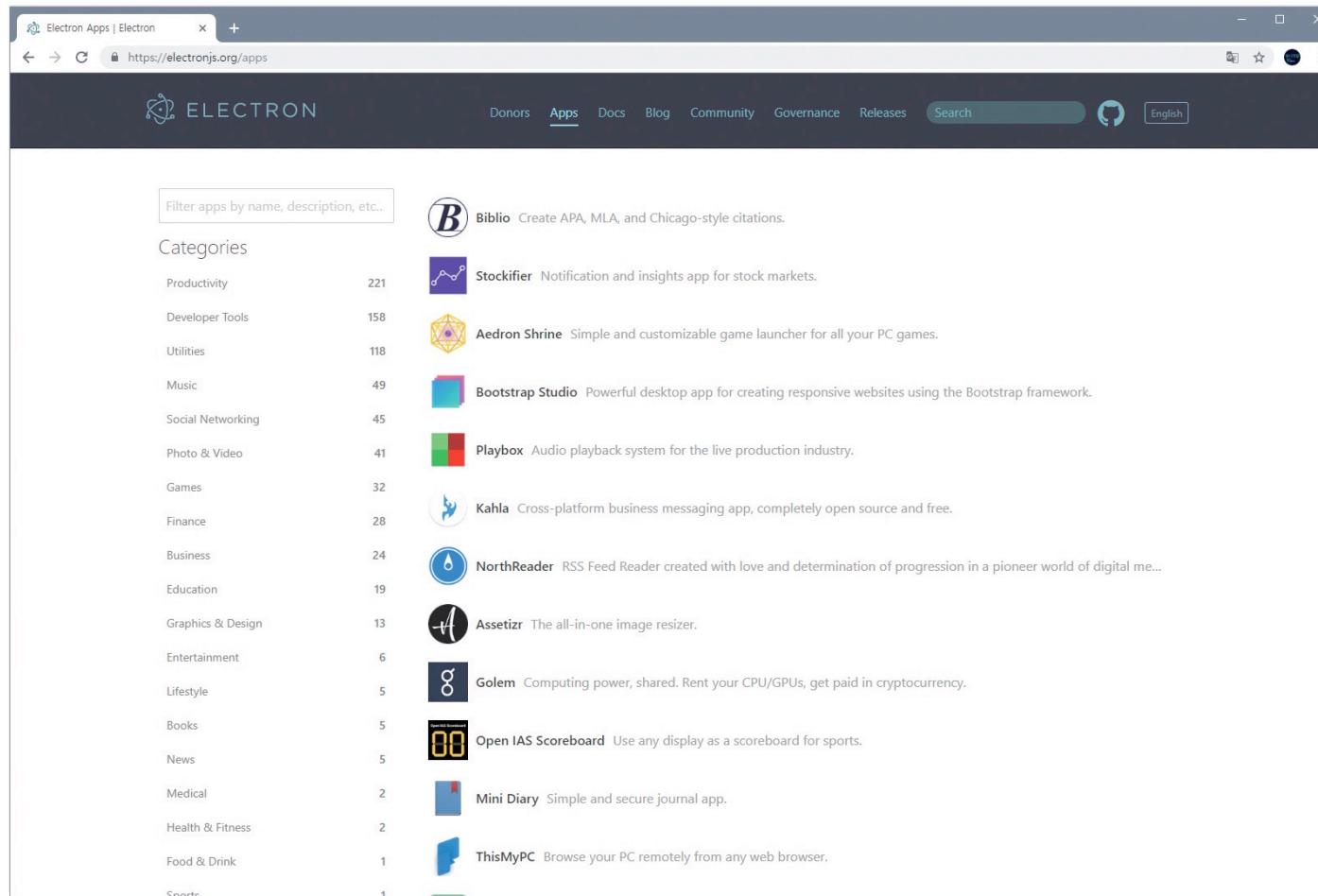


그림 1-22 일렉트론으로 개발된 애플리케이션

05 HTML5를 공부하면 좋은 이유

- 리액트 네이티브를 사용해 모바일 애플리케이션을 만들 수 있다.
 - HTML5로 개발 내부적으로 안드로이드와 아이폰에 맞는 네이티브 코드로 변환
 - 크로스 플랫폼 개발 - 한 번 개발해서 여러 장치(플랫폼)에 대응

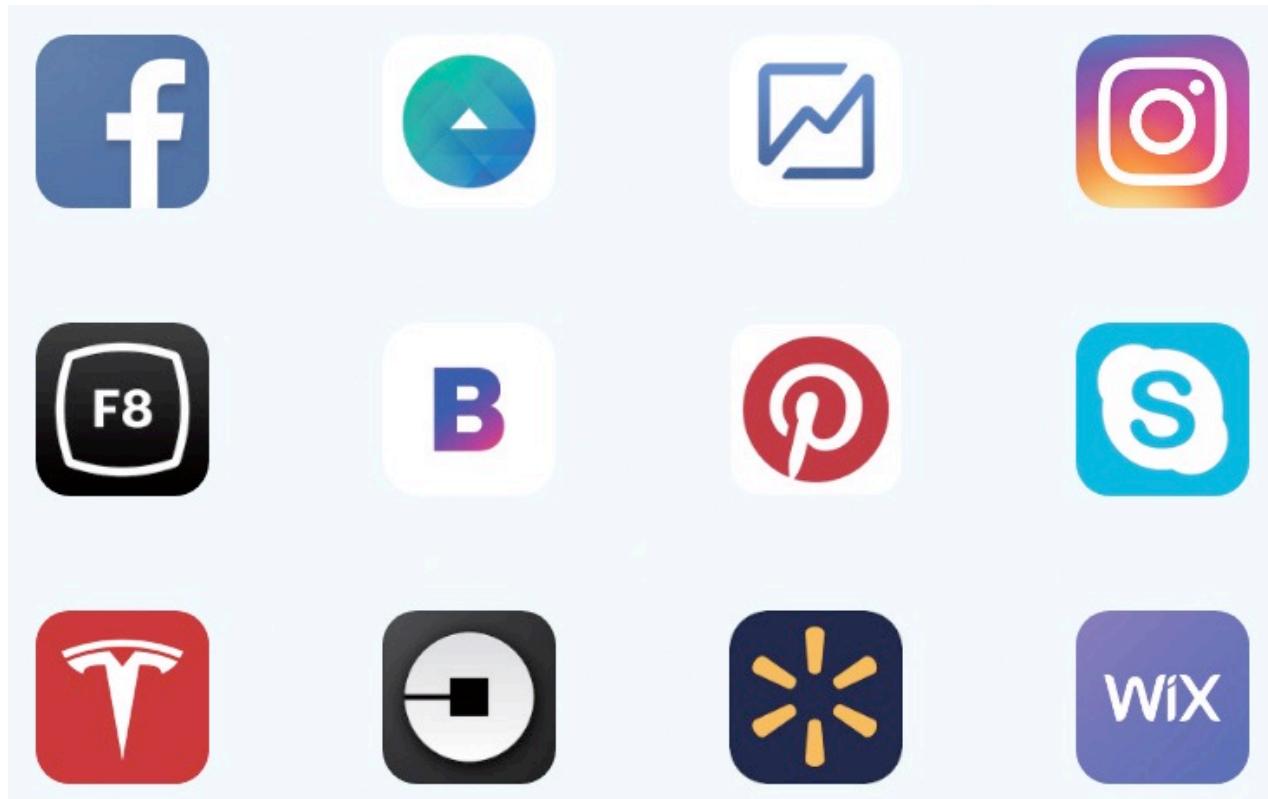


그림 1-23 리액트 네이티브로 개발된 모바일 애플리케이션

06 실습 환경 구축

■ 에디터

- 비주얼 스튜디오 코드 Visual Studio Code
- 비주얼 스튜디오 코드는 무료로 현재 웹 개발에서 가장 많이 사용

■ 웹 브라우저

- 구글 크롬
- 별도 표기가 없는 한 모든 웹 브라우저에서 동작(단, 인터넷 익스플로러 8 이하는 미지원)

06 실습 환경 구축

1. 웹 브라우저(크롬) 설치

- <http://www.google.com/chrome>

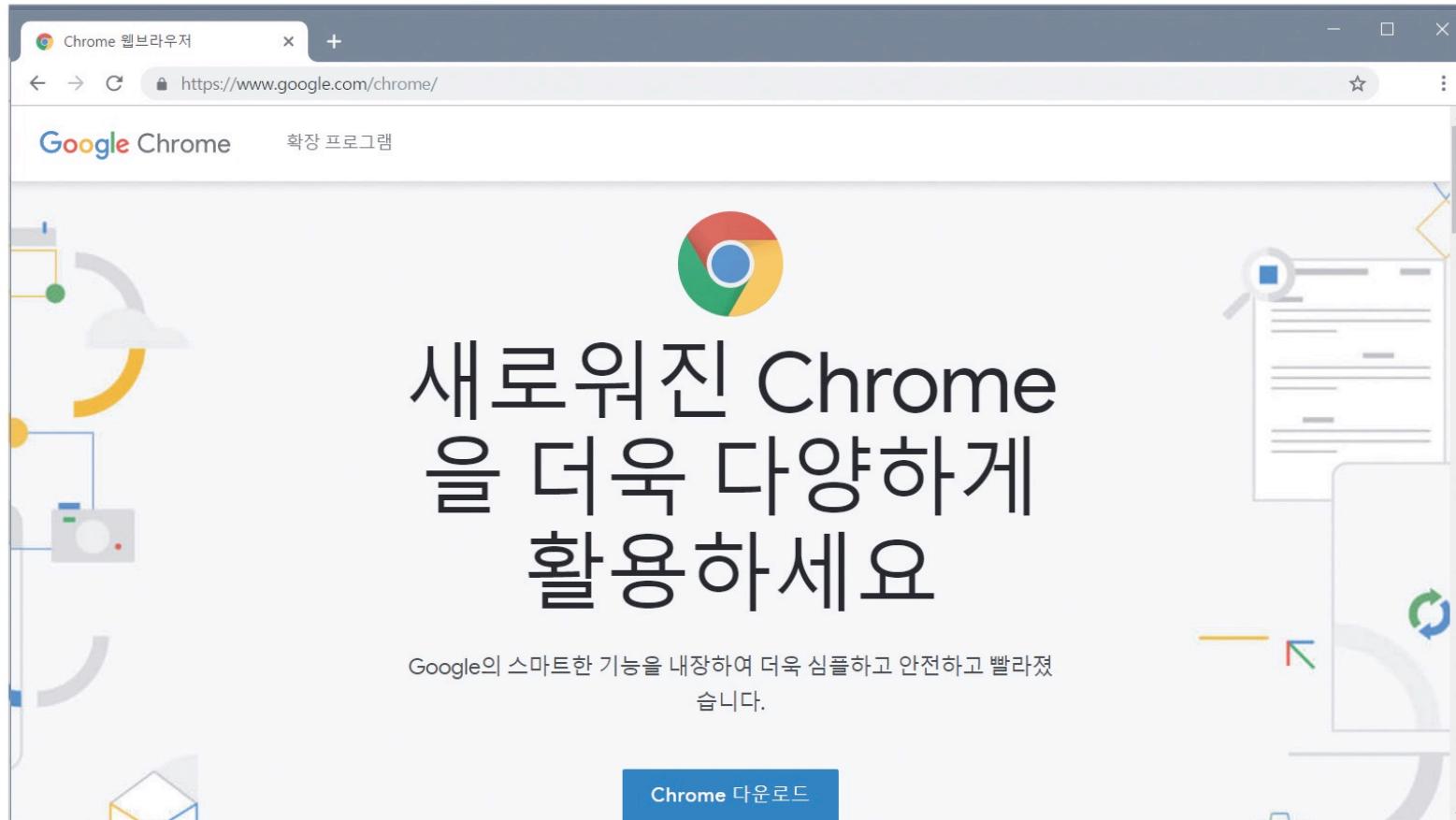


그림 1-24 구글 크롬 공식 사이트

06 실습 환경 구축

2. 에디터 설치와 설정(1)

- 비주얼 스튜디오 코드(<https://code.visualstudio.com/>)

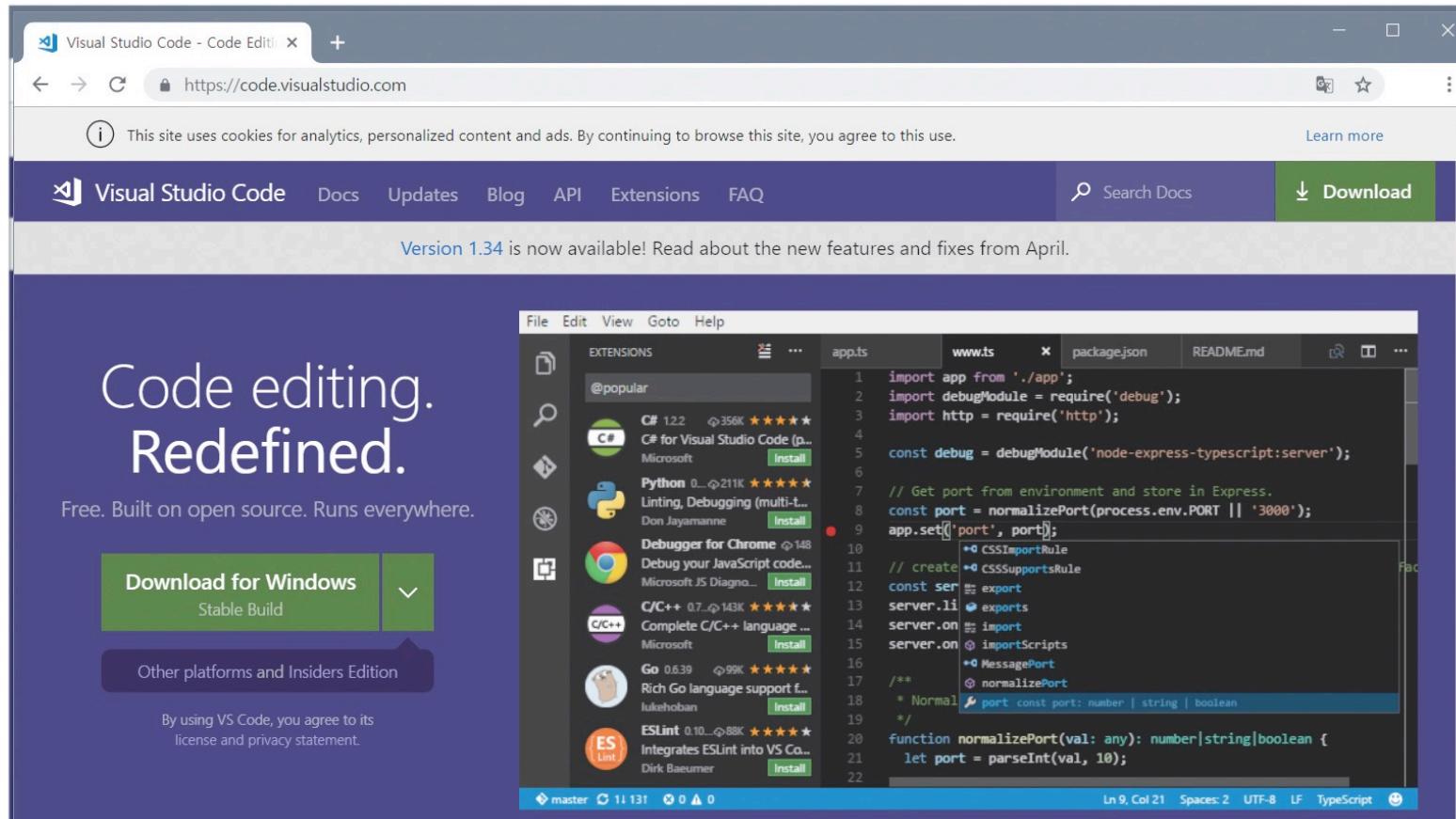


그림 1-25 비주얼 스튜디오 코드 공식 사이트

06 실습 환경 구축

3. 에디터 설치와 설정(2)

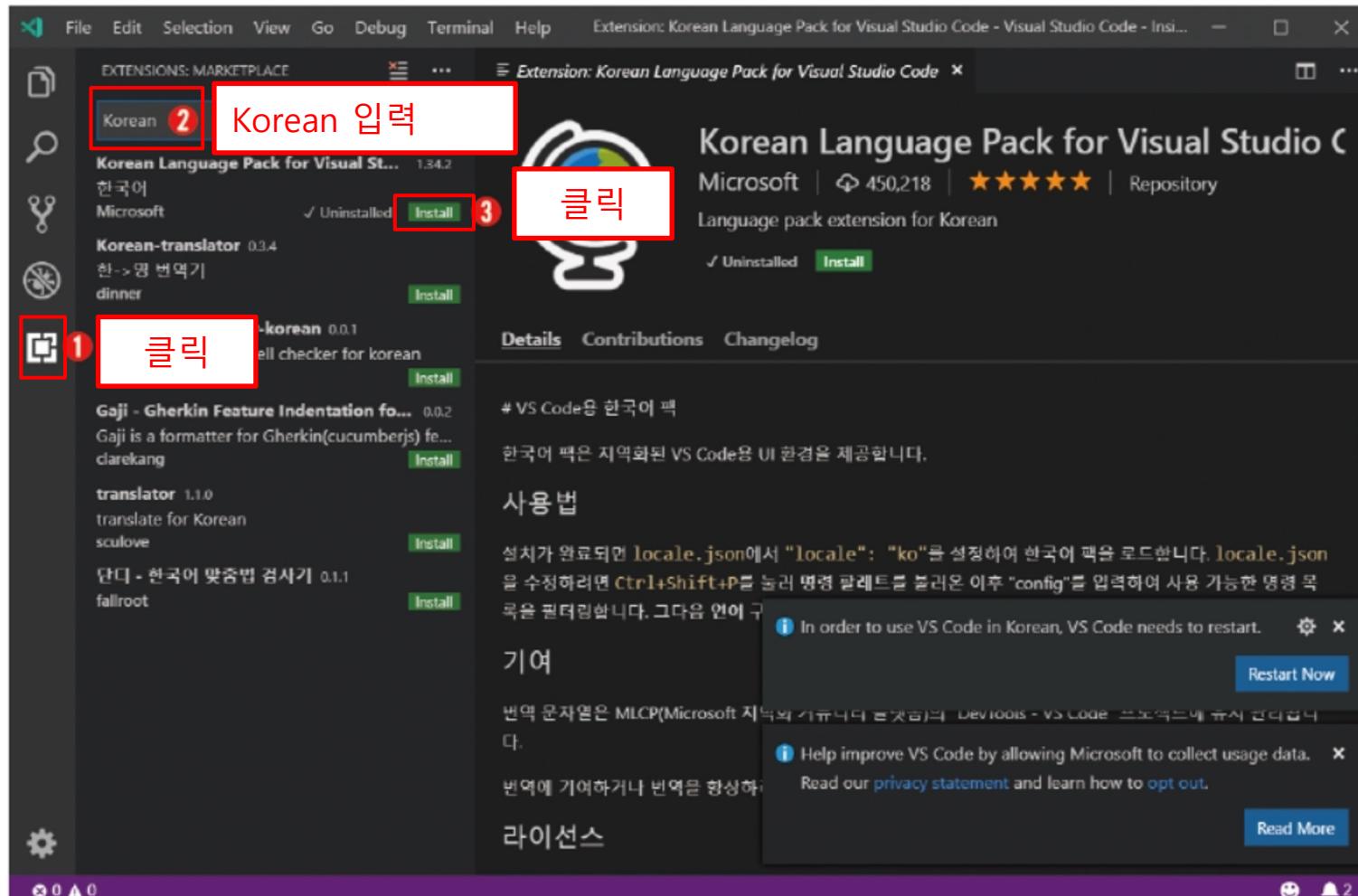


그림 1-26 비주얼 스튜디오 한국어 팩 설치

06 실습 환경 구축

4. 에디터 설치와 설정(3)

- D2Coding 글꼴
- (<https://github.com/naver/d2codingfont>)

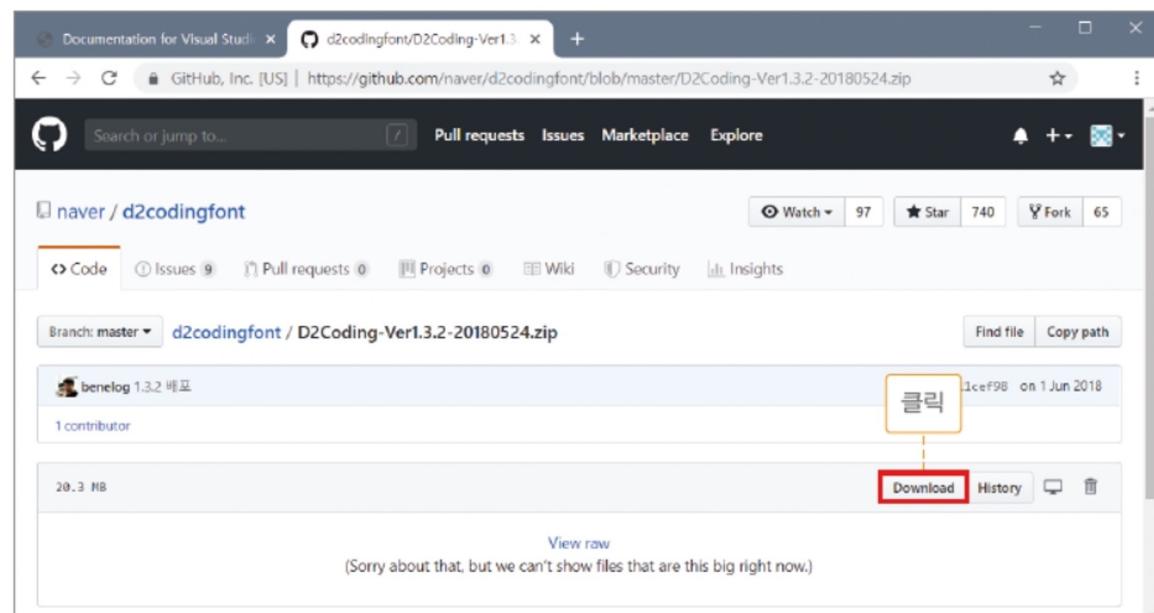
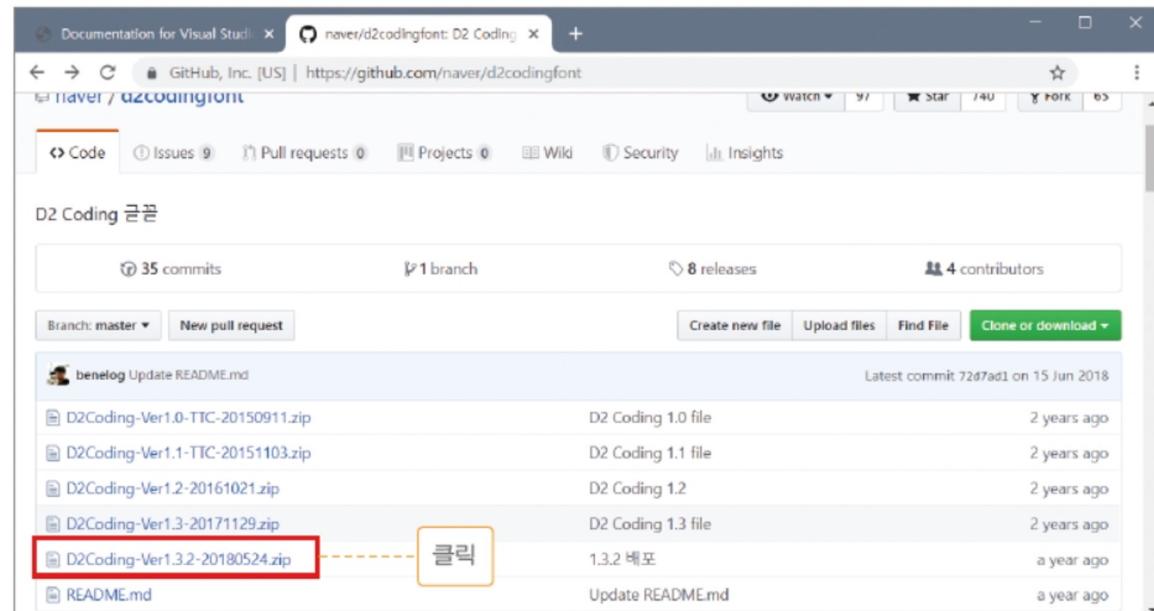


그림 1-27 코딩 글꼴 설치

06 실습 환경 구축

4. 에디터 설치와 설정(4)

- D2Coding 글꼴(<https://github.com/naver/d2codingfont>)

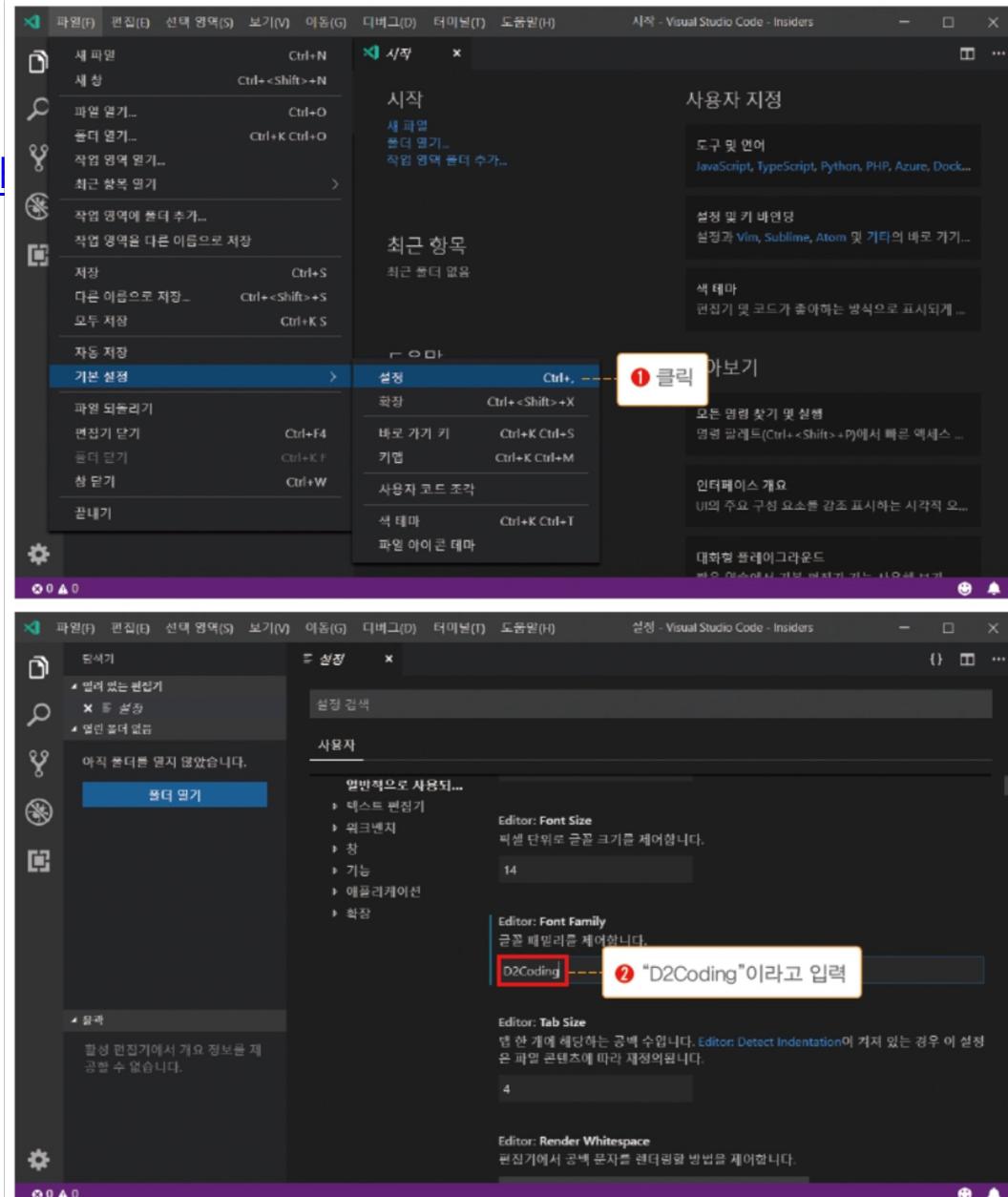


그림 1-28 비주얼 스튜디오 코드에 코딩 글꼴 등록

01 HTML5 기본 용어

■ 태그와 요소

- 요소 - HTML 페이지를 구성하는 각 부품(제목, 본문, 이미지 등)
- 태그 - 요소를 만들 때 사용하는 작성 방법

표 2-1 생성 방법에 따른 요소 구분

요소 구분	형태	예
내용을 가질 수 있는 요소	<요소 이름>내용</요소 이름>	<code><h1>Hello HTML5</h1></code> <code><p>즐거운 웹 프로그래밍 입문</p></code> <code><audio></audio></code>
내용을 가질 수 없는 요소	<요소 이름>	<code></code> <code>
</code> <code><hr></code>

01 HTML5 기본 용어

■ 태그와 요소

<h1>Hello HTML5</h1>

그림 2-1 내용을 가질 수 있는 요소 생성

표 2-2 내용 예

내용 구분	예
텍스트인 경우	<pre><h1>Hello HTML5</h1> <p>즐거운 웹 프로그래밍 입문</p></pre>
다른 태그인 경우	<pre><div> <h1>Hello HTML5</h1> <p>즐거운 웹 프로그래밍 입문</p> </div></pre>
내용을 입력하지 않은 경우	<pre><div></div> <audio></audio> <video></video></pre>

01 HTML5 기본 용어

■ 속성

- 속성 - 태그에 추가 정보를 부여할 때 사용하는 것

그림 2-2 속성의 사용 예

01 HTML5 기본 용어

■ 주석

- 주석 - 코드 설명 기록(프로그램의 실행에 영향을 미치지 않음)

```
<!-- 주석 -->
<!DOCTYPE html>
<html>
<head>
    <!-- title 태그 -->
    <title>TITLE</title>
</head>
<body>
    <!-- h1 태그 -->
    <h1>Hello HTML5</h1>
</body>
</html>
```

Hello HTML5

그림 2-3 주석 형식과 사용 예

02 HTML5 페이지 구조와 작성법

■ HTML5 페이지의 구조



그림 2-4 HTML5 구조 예

02 HTML5 페이지 구조와 작성법

■ HTML5 페이지의 구조

- <html> 태그
 - 웹 페이지의 사용 언어를 구글 검색 엔진에 제공

표 2-3 html 태그의 lang 속성

<html lang="ko">

lang 속성 값	언어
ko	한국어
en	영어
ja	일본어
ru	러시아어
zh	중국어
de	독일어

표 2-4 head 태그 내부에 넣을 수 있는 태그

태그	설명
meta	웹 페이지에 추가 정보 전달
title	페이지 제목 지정
script	웹 페이지에 <u>스크립트</u> 추가
link	웹 페이지에 다른 파일 추가
style	웹 페이지에 <u>스타일시트</u> 추가
base	웹 페이지의 기본 경로 지정

02 HTML5 페이지 구조와 작성법

- **HTML5 페이지의 작성과 실행 1 : 새 파일 만들기**
 - 비주얼 스튜디오 코드의 [파일]-[새 파일] (또는 단축키 Ctrl + N 사용)

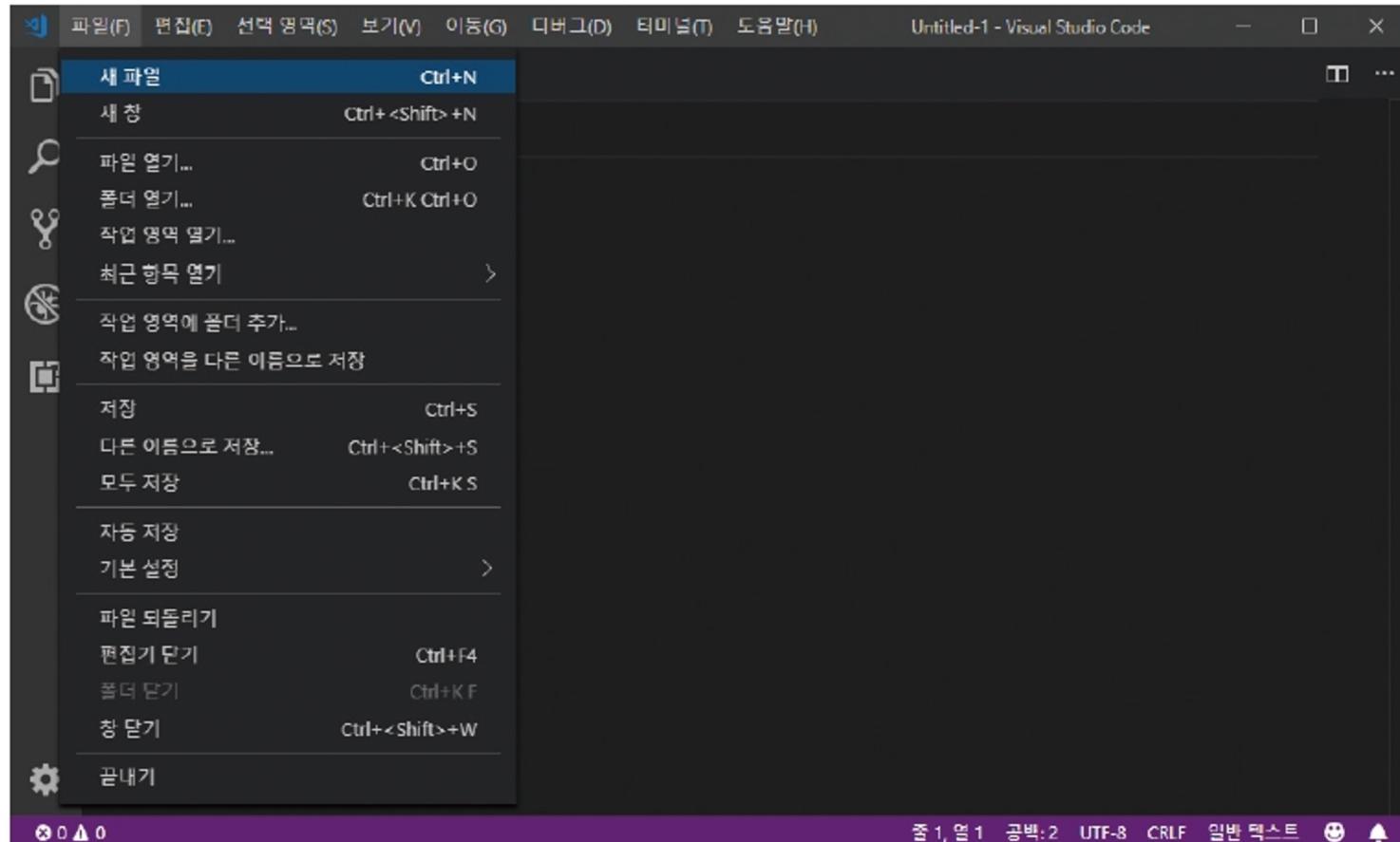


그림 2-6 새 파일 만들기

02 HTML5 페이지 구조와 작성법

■ HTML5 페이지의 작성과 실행 2 : 코드 작성 후 파일로 저장

- [파일]-[다른 이름으로 저장]
- OO.html 또는 OO.htm 형식으로 저장

코드 2-1 HTMLPage.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML5 Basic</title>
</head>
<body>
  <h1>Hello World..!</h1>
</body>
</html>
```

02 HTML5 페이지 구조와 작성법

■ HTML5 페이지의 작성과 실행 3 : 실행

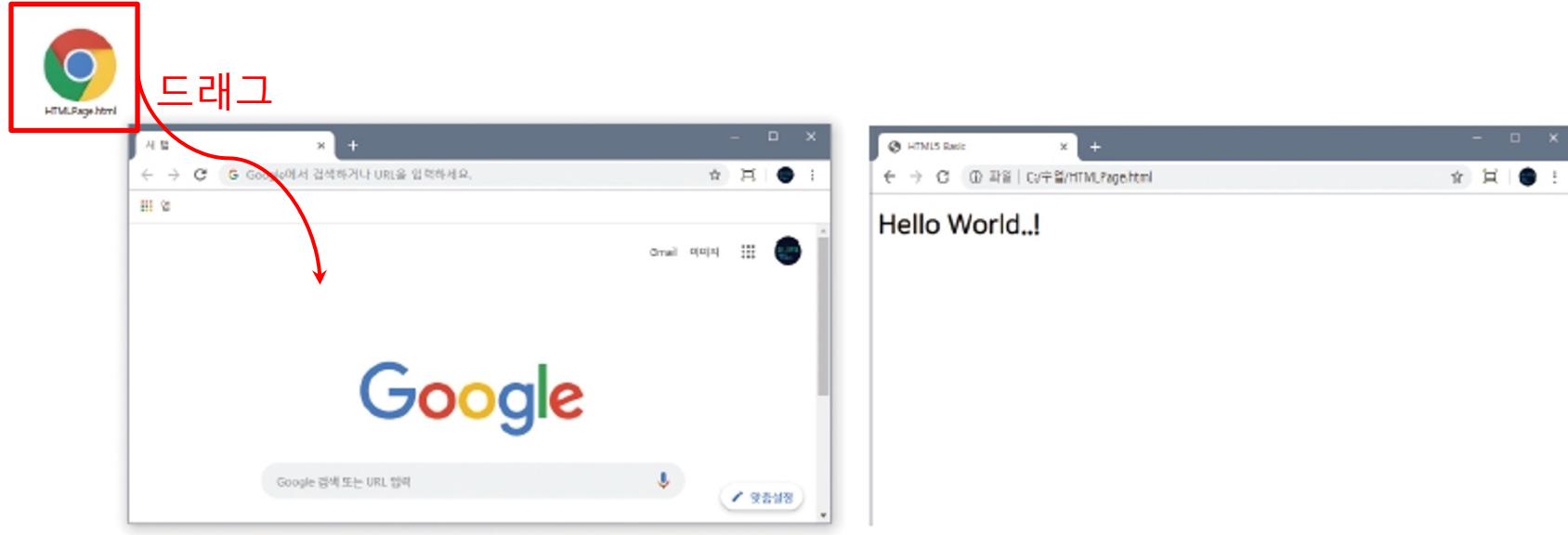


그림 2-7 실행

02 HTML5 페이지 구조와 작성법

■ 스타일시트 작성과 실행

▪ 내부 스타일

- HTML 페이지 내부에서 style 태그를 사용해 스타일시트를 직접 입력
- 스타일시트가 짧은 경우

▪ 외부 스타일

- 스타일시트를 별도로 생성하고 link 태그의 href 속성을 사용해 불러옴
- 협업 업무나 프로젝트의 규모가 클 경우 사용

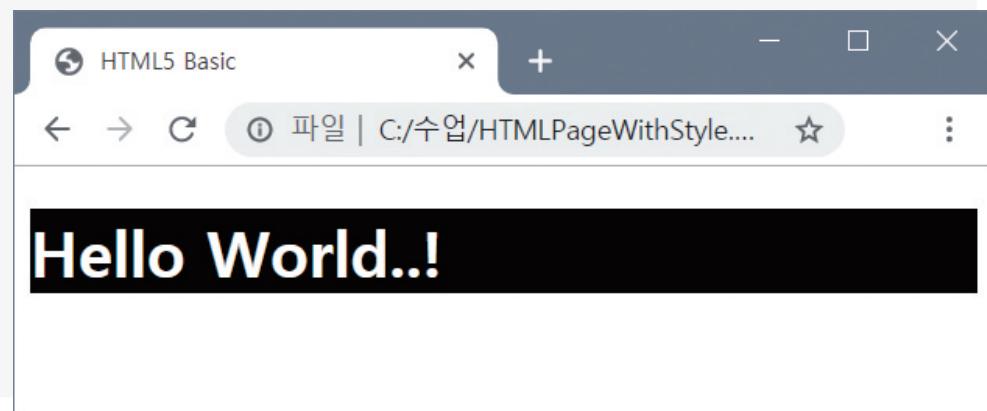
02 HTML5 페이지 구조와 작성법

■ [기본 예제 2-1] 내부 스타일시트 작성과 실행

[코드 데모](#)[온라인 에디터](#)

코드 2-2 HTMLPageWithStyle.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5 Basic</title>
    <style>
        h1 {
            color: white;
            background: black;
        }
    </style>
</head>
<body>
    <h1>Hello World..!</h1>
</body>
</html>
```



02 HTML5 페이지 구조와 작성법

- [기본 예제 2-2] 외부 스타일시트 작성과 실행

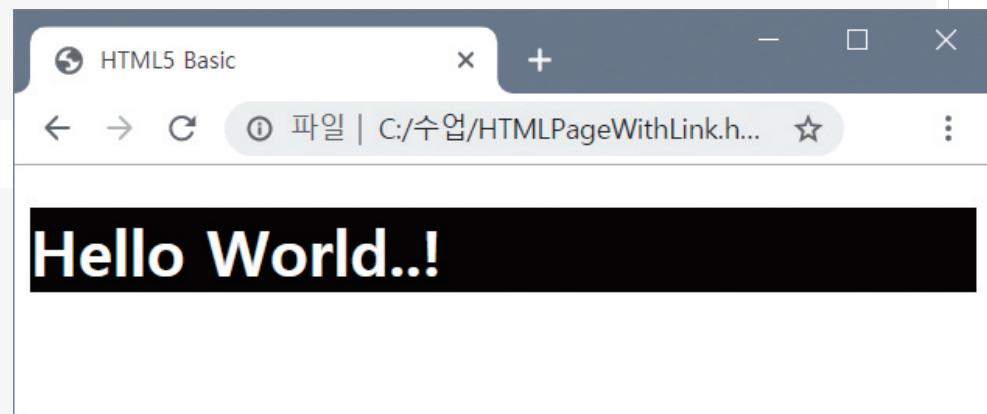
[코드 데모](#)

코드 2-3 Style.css

```
h1 {  
    color: white;  
    background: black;  
}
```

코드 2-4 HTMLPageWithLink.html

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>HTML5 Basic</title>  
    <link rel="stylesheet" href="Style.css">  
</head>  
<body>  
    <h1>Hello World..!</h1>  
</body>  
</html>
```



02 HTML5 페이지 구조와 작성법

- **자바스크립트 작성과 실행**

- **내부 자바스크립트**

- <script> 태그를 사용해 HTML 페이지 내부에 코드 작성

- **외부 자바스크립트**

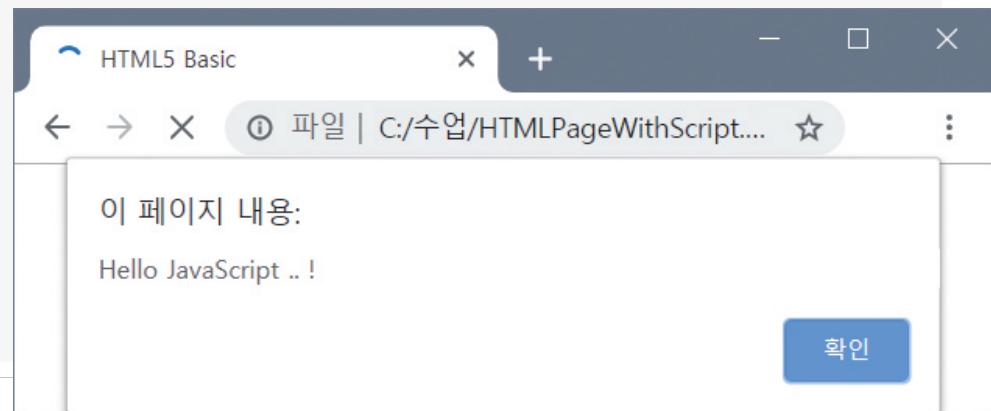
- <script> 태그의 src 속성에 파일 경로를 입력해 HTML 페이지로 불러옴

02 HTML5 페이지 구조와 작성법

- [기본 예제 2-3] 내부 자바스크립트 작성과 실행 [코드 데모](#)

코드 2-5 HTMLPageWithScript.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5 Basic</title>
    <script>
        // 경고 창을 출력합니다.
        alert('Hello JavaScript .. !');
    </script>
</head>
<body>
</body>
</html>
```



02 HTML5 페이지 구조와 작성법

- [기본 예제 2-4] 외부 자바스크립트 작성과 실행 [코드 데모](#)

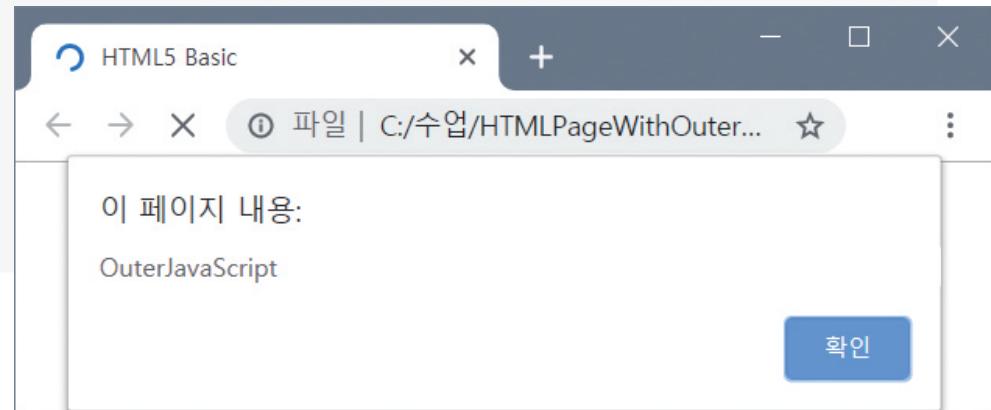
코드 2-6 OuterJavaScript.js

```
alert('OuterScript');
```

코드 2-7 HTMLPageWithOuterScript.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5 Basic</title>
    <script src="OuterJavaScript.js"></script>
</head>
<body>
</body>
</html>
```

HTML5 이전에서는 type 속성을 사용해야 했으나 이제는 type 속성 없이도 기본으로 인식합니다.
<script type="text/javascript" src="OuterJavaScript.js">



03 오류와 검증

■ 검사를 이용한 오류 확인

- 버그 ^{Bug} : 프로그램이 원하지 않는 방향으로 동작하는 것
- 디버그 ^{Debug} : 버그를 잡는(수정하는) 행위
- 웹 브라우저 검사 기능으로 디버그 수행
 - 크롬을 열고 [F12] 또는 [Ctrl] + [Shift] + [I]
 - 웹 페이지에서 마우스 오른쪽 버튼 클릭해 [검사] 메뉴

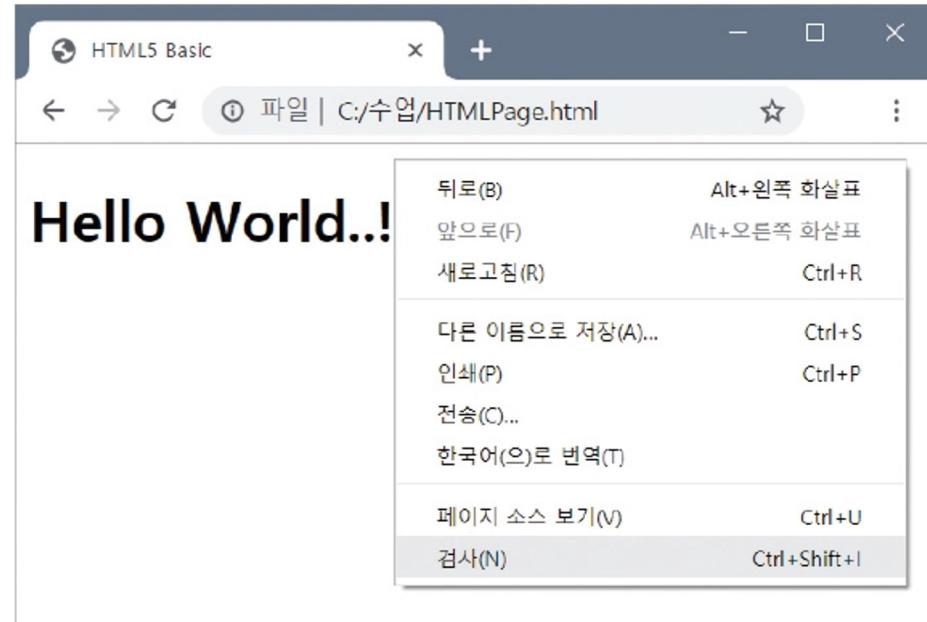


그림 2-8 검사 실행

03 오류와 검증

■ 검사를 이용한 오류 확인

- [Elements] 탭 : 현재 HTML 페이지의 계층 구조를 보여줌. 태그 스타일 파악
- [Console] 탭 : 오류를 확인, 자바스크립트 코드 추가 입력

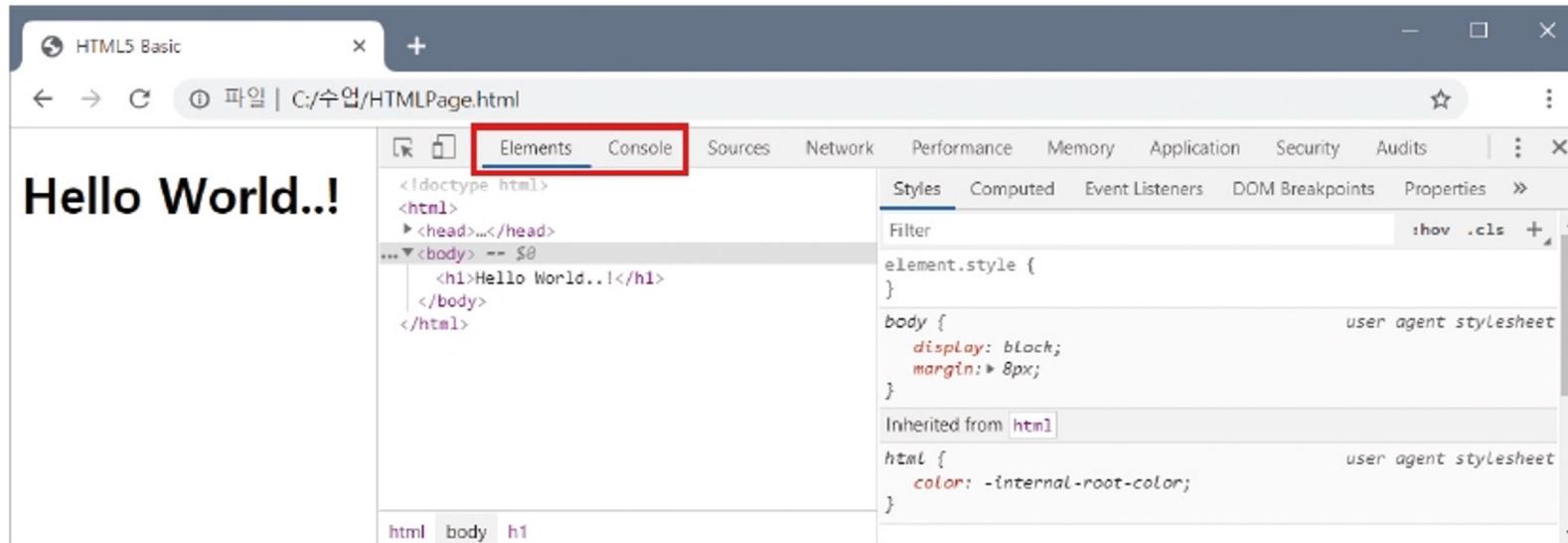


그림 2-9 검사를 사용한 계층 구조 확인

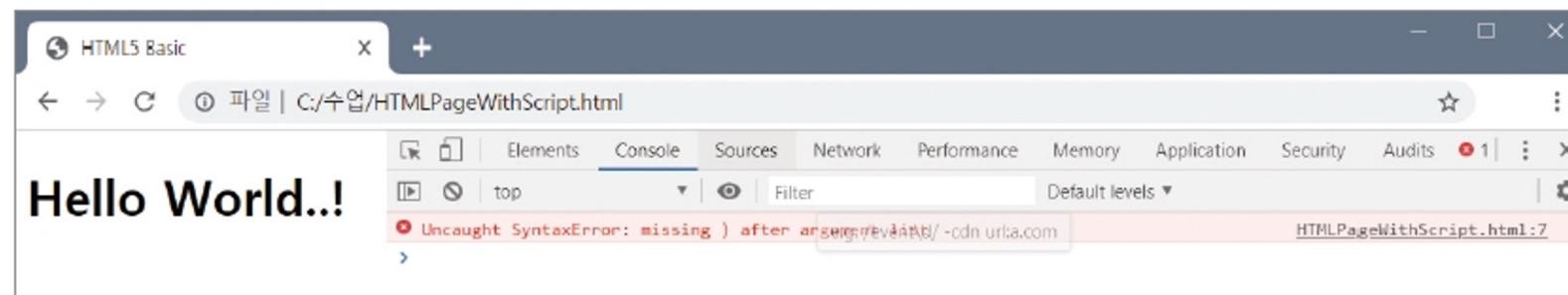


그림 2-10 검사를 사용한 오류 확인

Contents

■ 학습목표

- HTML5에서 지원하는 기본 태그를 사용할 수 있습니다.

| 구분 | 태그 | 설명 |
|--------|---------------------------------|---------------------------|
| 글자 태그 | h1~h6 | 제목 글자 생성 |
| | p, br, hr | 본문 생성 |
| | a | 하이퍼링크 생성 |
| | b, i, small, sub, sup, ins, del | 글자 모양 지정 |
| 목록 태그 | ul/ol, li | 순서가 없는/있는 목록 생성, 목록 요소 생성 |
| 테이블 태그 | table, tr, th, td | 표 생성 |
| 미디어 태그 | img, audio, video | 이미지, 오디오, 비디오 삽입 |

■ 내용

- 글자 태그
- 목록 태그
- 테이블 태그
- 미디어 태그

01 글자 태그

■ 페이지에서는 글자 태그가 비중이 큼

The screenshot shows a Wikipedia article page for "World Wide Web". The page title is "World Wide Web" and it is an article. The URL is https://en.wikipedia.org/wiki/World_Wide_Web. The page content discusses the World Wide Web (WWW) as an information system where documents and other web resources are identified by Uniform Resource Locators (URLs), which may be interlinked by hypertext, and are accessible over the Internet. It mentions Tim Berners-Lee's invention of the web browser in 1989 and its development over time. The page also covers web resources, web pages, and websites, mentioning the Hypertext Markup Language (HTML). Two callout boxes are present: one showing a screenshot of a web browser displaying a complex multi-page website, and another showing a world map with colored regions representing web index data for different countries in 2014.

그림 3-1 글자 위주의 웹 페이지

01 글자 태그

■ 제목과 본문 글자 태그

- 문서의 제목을 표현할 때 사용
- h는 Heading을 의미

표 3-1 제목 글자 태그

| 태그 | 설명 | |
|-------|----|-------------------|
| 제목 글자 | h1 | 첫 번째로 큰 제목 글자 생성 |
| | h2 | 두 번째로 큰 제목 글자 생성 |
| | h3 | 세 번째로 큰 제목 글자 생성 |
| | h4 | 네 번째로 큰 제목 글자 생성 |
| | h5 | 다섯 번째로 큰 제목 글자 생성 |
| | h6 | 여섯 번째로 큰 제목 글자 생성 |
| 본문 글자 | p | 본문 문단 생성 |
| | br | 줄 바꿈 |
| | hr | 수평 줄 삽입 |

01 글자 태그

■ [기본 예제 3-1] 제목 표현

[코드 데모](#) [온라인 에디터](#)

코드 3-1 text_header.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5+CSS3 Text</title>
</head>
<body>
    <h1>제목 글자 태그 1</h1>
    <h2>제목 글자 태그 2</h2>
    <h3>제목 글자 태그 3</h3>
    <h4>제목 글자 태그 4</h4>
    <h5>제목 글자 태그 5</h5>
    <h6>제목 글자 태그 6</h6>
</body>
</html>
```

제목 글자 태그 1

제목 글자 태그 2

제목 글자 태그 3

제목 글자 태그 4

제목 글자 태그 5

제목 글자 태그 6

■ [기본 예제 3-2] 본문 단락 구분

[코드 데모](#)

코드 3-2 text_paragraph.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML TEXT Basic Page</title>
</head>
<body>
    <h1>제목 글자</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p>Phasellus sem tortor, volutpat vitae euismod eget</p>
</body>
</html>
```

제목 글자

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer sagittis enim et felis fringilla sit amet vehicula enim tincidunt. Duis turpis velit, auctor eget porta a, consectetur euismod metus. Mauris auctor, augue sed elementum condimentum, sem felis vulputate ligula, ac commodo massa turpis et massa. Nam mollis orci sed augue condimentum rhoncus. Aenean libero lacus.

Phasellus sem tortor, volutpat vitae euismod eget, sagittis sed ipsum. Aliquam convallis accumsan massa, egestas consequat eros euismod at. In in porta nisi. Nam nec eros velit. Proin vulputate ornare varius. Phasellus dapibus lacus at nibh porttitor malesuada. Cras quam mauris, placerat ac euismod ac, dapibus non dui. Quisque vitae erat sit amet massa porta tempor ac ac risus.

01 글자 태그

■ [기본 예제 3-3] 제목과 본문 태그의 활용

[코드 데모](#)

코드 3-3 text_content.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML5+CSS3 Text</title>
</head>
<body>
    <h1>홍차</h1>
    <hr>
    <h2>정의</h2>
    <p>홍차는 백차, 녹차, 우롱차보다 더 많이 발효된 차의 일종이다. 동양에서는 찻물의 빛이 붉기 때문에 홍차라고 부르지만, 서양에서는 찻잎의 색깔 때문에 'black tea'라고 부른다.</p>
    <br>
    <h2>등급</h2>
    <p>홍차는 여러 가지로 등급이 매겨진다. 일반적으로 찻잎의 모양에 따른 등급과 가공 상태에 따른 등급을 조합하여 표시한다.</p>
    <p>- 브로큰 페코</p>
    <p>- 브로큰 페코 수송</p>
    <p>- 브로큰 오렌지 페코 패닝</p>
</body>
</html>
```

홍차

정의

홍차는 백차, 녹차, 우롱차보다 더 많이 발효된 차의 일종이다. 동양에서는 찻물의 빛이 붉기 때문에 홍차라고 부르지만, 서양에서는 찻잎의 색깔 때문에 'black tea'라고 부른다.

등급

홍차는 여러 가지로 등급이 매겨진다. 일반적으로 찻잎의 모양에 따른 등급과 가공 상태에 따른 등급을 조합하여 표시한다.

- 브로큰 페코
- 브로큰 페코 수송
- 브로큰 오렌지 페코 패닝

01 글자 태그

■ 앵커태그

- 하이퍼텍스트 : 사용자의 선택에 따라 특정 정보로 이동하는 조직된 문서
 - a 태그(Anchor) : 다른 웹 페이지나 웹 페이지 내부의 특정 위치로 이동
 - Href : Hyper Reference를 의미

표 3-2 앵커 태그

| 태그 | 설명 |
|----|----------|
| a | 하이퍼링크 생성 |

[한빛미디어](http://hanbit.co.kr)

그림 3-2 a 태그의 href 속성

01 글자 태그

■ 앵커 태그

- a 태그의 href 속성

① 절대 경로

http://naver.com - 네이버의 메인 페이지

/animal.jpg - 현재 웹 사이트 최상위 위치의 animal.jpg 파일

② 상대 경로

animal.jpg - 웹 페이지가 있는 폴더의 animal.jpg 파일

image/animal.jpg - 웹 페이지가 있는 폴더에 포함된 image 폴더의 animal.jpg 파일

../animal.jpg - 웹 페이지가 있는 폴더의 상위 폴더에 있는 animal.jpg 파일

③ 아이디 경로

#name - id 속성이 name인 태그의 위치로 이동

④ 메일 경로

mailto : hanb@hanbit.co.kr – 해당 주소로 메일 전송

■ [기본 예제 3-4] 하이퍼링크 설정

[코드 데모](#)

1

1. 특정 웹 페이지에 연결하기

- 하이퍼링크를 설정한 글자를 클릭하면 해당 웹 페이지로 이동

코드 3-5 text_anchor.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML TEXT Basic Page</title>
</head>
<body>
    <a href="http://www.hanbit.co.kr">한빛미디어</a><br>
    <a href="http://www.naver.com/">네이버</a><br>
    <a href="http://www.daum.com/">다음</a><br>
</body>
</html>
```

■ [기본 예제 3-4] 하이퍼링크 설정

[코드 데모](#)[온라인 에디터](#)

2. 웹 페이지 내부에 연결하기

- 하이퍼링크를 설정한 글자를 클릭하면 해당 웹 페이지로 이동

코드 3-6 text_anchorInner.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML TEXT Basic Page</title>
</head>
<body>
    <a href="#alpha">Alpha 부분</a>
    <a href="#beta">Beta 부분</a>
    <a href="#gamma">Gamma 부분</a>
    <hr>
    <h1 id="alpha">Alpha</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <h1 id="beta">Beta</h1>
    <p>Vivamus elementum pretium tortor, in tincidunt mauris gravida et.</p>
    <h1 id="gamma">Gamma</h1>
    <p>Phasellus sem tortor, volutpat vitae euismod eget, sagittis sed</p>
</body>
</html>
```

이동하려는 태그의 id 속성을 href 속성에 '#id 속성' 형태로 입력합니다.

01 글자 태그

■ 글자 모양 태그

표 3-3 글자 모양 태그

| 태그 | 설명 | 태그 | 설명 |
|----|----------------|-------|--------------------|
| b | 굵은 글자 bold | small | 작은 글자 |
| i | 기울어진 글자 italic | sub | 아래 첨자 subscript |
| | | sup | 위 첨자 superscript |
| | | ins | 밀줄 글자 insert |
| | | del | 취소선이 그어진 글자 delete |

- 글자 모양 태그 내부에 제목 글자 태그와 본문 글자 태그는 넣을 수 없음

<i>

<h1>웹 표준 위반</h1>

<p>웹 표준 위반</p>

</i>

그림 3-3 웹 표준을 위반한 글자 모양 태그 사용 예

01 글자 태그

■ [기본 예제 3-5] 다양한 글자 모양

[코드 데모](#)

코드 3-7 text_font.html

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML TEXT Basic Page</title>
</head>
<body>
    <h1><b>Lorem ipsum dolor sit amet</b></h1>
    <h1><i>Lorem ipsum dolor sit amet</i></h1>
    <h1><small>Lorem ipsum dolor sit amet</small></h1>
    <h1>Lorem ipsum dolor<sub>sit amet</sub></h1>
    <h1>Lorem ipsum dolor<sup>sit amet</sup></h1>
    <h1><ins>Lorem ipsum dolor sit amet</ins></h1>
    <h1><del>Lorem ipsum dolor sit amet</del></h1>
    <hr>
    <b>Lorem ipsum dolor sit amet</b><br>
    <i>Lorem ipsum dolor sit amet</i><br>
    <small>Lorem ipsum dolor sit amet</small><br>
    Lorem ipsum dolor<sub>sit amet</sub><br>
    Lorem ipsum dolor<sup>sit amet</sup><br>
    <ins>Lorem ipsum dolor sit amet</ins><br>
    <del>Lorem ipsum dolor sit amet</del><br>
</body>
</html>
```

Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet
 Lorem ipsum dolor sit amet

02 목록 태그

- 내비게이션 메뉴
 - 웹 사이트의 다른 웹 페이지로 이동할 수 있는 버튼



그림 3-4 다음 메인 페이지의 내비게이션 메뉴

표 3-4 목록 태그

- 내비게이션 메뉴를 만들기 위해 주로 사용되는 목록 태그

| 태그 | 설명 |
|----|--|
| ul | 순서가 없는 목록 <small>unordered list</small> 생성 |
| ol | 순서가 있는 목록 <small>ordered list</small> 생성 |
| li | 목록 요소 <small>list item</small> 생성 |

■ [기본 예제 3-6] 목록 태그 활용

[코드 데모](#)

1

1. 순서가 없는 기본(글머리 기호) 목록 만들기

코드 3-8 list_unordered.html

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

- 사과
- 바나나
- 오렌지

2. 순서가 있는 목록 만들기

코드 3-9 list_ordered.html

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

1. 사과
2. 바나나
3. 오렌지

■ [기본 예제 3-6] 목록 태그 활용

[코드 데모](#)

1

3. 중첩 목록 만들기

코드 3-10 nested_list.html

```
<body>
  <ul>
    <!-- 첫 번째 목록 -->
    <li>
      <b>과일</b>
      <ol>
        <li>사과</li>
        <li>바나나</li>
        <li>오렌지</li>
      </ol>
    </li>
    <!-- 두 번째 목록 -->
    <li>
      <b>채소</b>
      <ol>
        <li>상추</li>
        <li>치커리</li>
        <li>양배추</li>
      </ol>
    </li>
  </ul>
</body>
```

- 과일
 - 1. 사과
 - 2. 바나나
 - 3. 오렌지
- 채소
 - 1. 상추
 - 2. 치커리
 - 3. 양배추

03 테이블 태그

■ 테이블 태그

표 3-5 테이블 태그

| 태그 | 설명 |
|-------|--------------------------|
| table | 표 삽입 |
| tr | 표에 행 table row 삽입 |
| th | 표의 제목 셀 table heading 생성 |
| td | 표의 일반 셀 table data 생성 |

■ [기본 예제 3-7] 시간표 만들기

[코드 데모](#)

1. 표 만들기

코드 3-11 table_basic.html

```
<body>
  <table>

    </table>
</body>
```

03 테이블 태그

■ [기본 예제 3-7] 시간표 만들기

[코드 데모](#)

2. 표에 셀 추가하기

코드 3-12 table_basic.html

```
<body>
  <table border="1">
    <thead>
      <tr>
        <th></th>
        <th>월</th>
        <th>화</th>
        <th>수</th>
        <th>목</th>
        <th>금</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1교시</td>
        <td>영어</td>
        <td>국어</td>
        <td>과학</td>
        <td>미술</td>
        <td>기술</td>
      </tr>
      <tr>
        <td>2교시</td>
        <td>도덕</td>
        <td>체육</td>
        <td>영어</td>
        <td>수학</td>
        <td>사회</td>
      </tr>
    </tbody>
  </table>
</body>
```

표의 테두리를 지정하는 속성입니다.

```
<tr>
  <td>2교시</td>
  <td>도덕</td>
  <td>체육</td>
  <td>영어</td>
  <td>수학</td>
  <td>사회</td>
</tr>
</tbody>
</table>
</body>
```

| | 월 | 화 | 수 | 목 | 금 |
|-----|----|----|----|----|----|
| 1교시 | 영어 | 국어 | 과학 | 미술 | 기술 |
| 2교시 | 도덕 | 체육 | 영어 | 수학 | 사회 |

03 테이블 태그

- 테이블 태그의 속성

표 3-6 테이블 태그의 속성

| 태그 | 속성 | 설명 |
|--------|---------|--------------|
| table | border | 표의 테두리 두께 지정 |
| th, td | colspan | 셀의 너비 지정 |
| | rowspan | 셀의 높이 지정 |

03 테이블 태그

■ [기본 예제 3-8] 행·열 병합 표 생성

[코드 데모](#)

- colspan 속성과 rowspan 속성을 적용

코드 3-13 table_span.html

```
<body>
  <table border="1">
    <tr>
      <th colspan="2">지역별 흥차</th>
    </tr>
    <tr>
      <th rowspan="3">중국</th>
      <td>정산소종</td>
    </tr>
    <tr><td>기문</td></tr>
    <tr><td>운남</td></tr>
    <tr>
      <th rowspan="4">인도 및 스리랑카</th>
      <td>아삼</td>
    </tr>
    <tr><td>실론</td></tr>
    <tr><td>다질링</td></tr>
    <tr><td>닐기리</td></tr>
  </table>
</body>
```

| 지역별 흥차 | |
|-----------|------|
| 중국 | 정산소종 |
| | 기문 |
| | 운남 |
| 인도 및 스리랑카 | 아삼 |
| | 실론 |
| | 다질링 |
| | 닐기리 |

04 미디어 태그

- 미디어 태그
 - 이미지, 오디오, 비디오 등 멀티미디어를 넣을 때 사용

표 3-7 미디어 태그 구분

| 내용물을 가질 수 있는 태그 | 내용물을 가질 수 없는 태그 |
|--|--------------------------|
| <code><audio></audio></code> | |
| <code><video></video></code> | <code></code> |

04 미디어 태그

- 미디어 태그 속성
 - 이미지, 오디오, 비디오에 필요한 추가 정보는 속성을 사용

표 3-8 이미지, 오디오, 비디오 태그의 속성

| 태그 | 속성 | 설명 |
|-----------------|----------|-------------------------------------|
| img 태그 | src | 이미지의 경로 지정 |
| | alt | 이미지가 없을 때 나오는 글자 지정 |
| | width | 이미지의 너비 지정 |
| | height | 이미지의 높이 지정 |
| audio, video 태그 | src | 음악, 비디오 파일의 경로 지정 |
| | preload | 음악, 비디오를 준비 중일 때 데이터를 모두 불러올지 여부 지정 |
| | autoplay | 음악, 비디오의 자동 재생 여부 지정 |
| | loop | 음악, 비디오의 반복 여부 지정 |
| | controls | 음악, 비디오 재생 도구 출력 여부 지정 |
| video 태그 | width | 비디오의 너비 지정 |
| | height | 비디오의 높이 지정 |

04 미디어 태그

■ [기본 예제 3-9] 멀티미디어(이미지, 오디오, 비디오) 삽입

▪ 1. 이미지 삽입하기

[코드 데모](#)

- 이미지 파일 준비 : 준비 파일(이미지.jpg)을 HTML 페이지와 같은 폴더에 넣기

코드 3-14 image_basic.html

```
<body>
    
    
</body>
```

다음과 같이 웹에 있는 이미지 경로를 넣어도 됩니다.



그림이 존재하지 않습니다.

04 미디어 태그

■ [기본 예제 3-9] 멀티미디어(이미지, 오디오, 비디오) 삽입

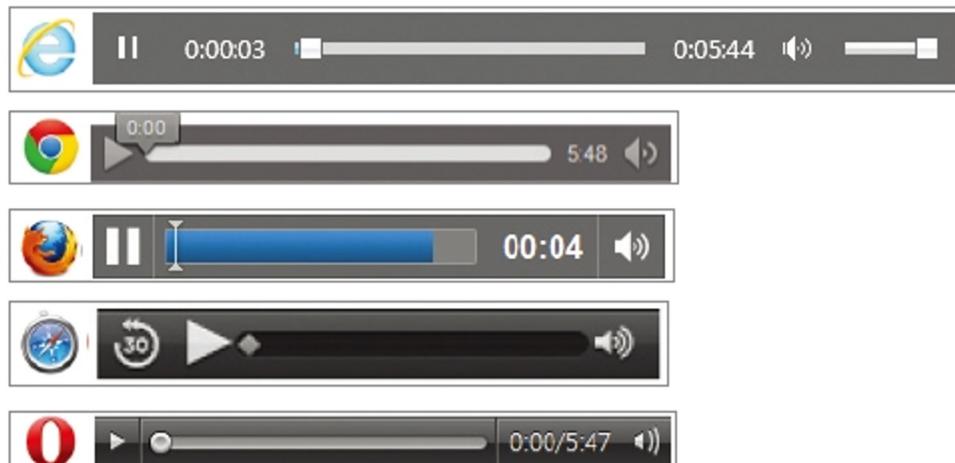
2. 음악 삽입하기

[코드 데모](#)

- 음악 파일 준비 : 준비 파일(오디오.mp3)을 HTML 페이지와 같은 폴더에 넣기

코드 3-15 audio_basic.html

```
<body>
    <audio src="Kalimba.mp3" controls="controls"></audio>
</body>
```



04 미디어 태그

■ [기본 예제 3-9] 멀티미디어(이미지, 오디오, 비디오) 삽입

3. 웹 브라우저 제약이 없도록 음악 삽입하기

[코드 데모](#)

- <source> 태그
 - 웹 브라우저마다 지원하는 음악 파일 확장자가 다른 문제 해결
 - <audio> 태그나 <video> 태그 내부에 입력
- ogg파일 준비: .ogg 확장자 파일을 HTML 페이지와 같은 폴더에 넣기

코드 3-16 audio_source.html

```
<body>
  <audio controls="controls">
    <source src="Kalimba.mp3" type="audio/mp3">
    <source src="Kalimba.ogg" type="audio/ogg">
  </audio>
</body>
```

type 속성을 입력하지 않으면 웹
브라우저가 음악 파일을 다운로드한 후
재생 가능한 파일인지 확인하는 작업이
필요하므로 꼭 지정합시다.

04 미디어 태그

■ [기본 예제 3-9] 멀티미디어(이미지, 오디오, 비디오) 삽입

4. 동영상 삽입하기

[코드 데모](#)

- 동영상 파일 준비 : 준비 파일(동영상.mp4, 동영상.webm)을 HTML 페이지와 같은 폴더에 넣기

코드 3-17 video_basic.html

```
<body>
  <video controls="controls">
    <source src="Wildlife.mp4" type="video/mp4">
    <source src="Wildlife.webm" type="video/webm">
  </video>
</body>
```



04 미디어 태그

■ [기본 예제 3-9] 멀티미디어(이미지, 오디오, 비디오) 삽입

5. 동영상을 불러오는 동안 다른 이미지 보여 주기

[코드 데모](#)

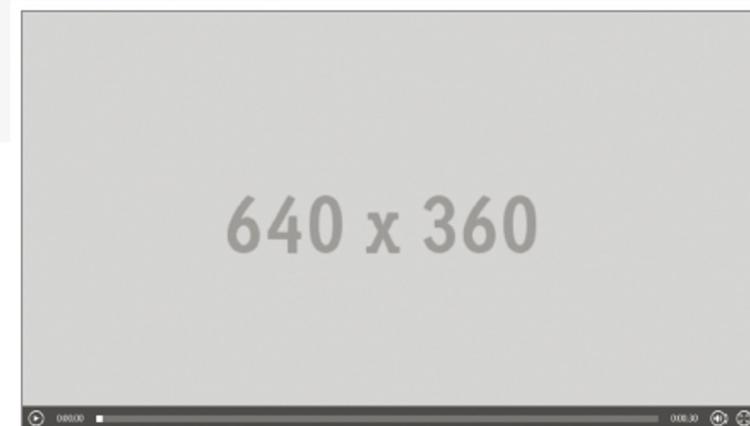
- poster 속성

- <video>태그의 속성
- 동영상을 불러오는 동안 사용자에게 보여 줄 이미지를 지정
- 이미지 경로 입력

코드 3-18 video_poster.html

여기서는 <http://placehold.it>에서 자동으로 생성한 이미지를 사용했습니다. 이 웹사이트는 주소 다음에 '너비×높이'를 입력하면 해당 크기의 이미지를 제공합니다.

```
<body>
  <video controls="controls" poster="http://placehold.it/640x360">
    <source src="Wildlife.mp4" type="video/mp4">
    <source src="Wildlife.webm" type="video/webm">
  </video>
</body>
```



01 입력 양식 태그

■ 입력 양식

- 사용자에게 정보를 입력받는 요소

회원가입

이름

생년월일 년도 월 일

성별 남 여

- 주민등록번호 입력 없이 전화번호로 회원가입이 가능합니다.
- 네이버카페/네이버드라이브 등의 대부분 서비스가 이용 가능합니다.
(단, 일부 공개 게시판 및 유료서비스는 실명확인 후 이용 가능합니다.)
- 이름, 생년월일과 성별은 추후 변경할 수 없습니다.

그림 4-1 간단한 회원가입 입력 양식

■ 입력 양식 개요

- <form>태그: 영역 생성

```
<body>
  <form>
    <input type="text" name="search">
    <input type="submit">
  </form>
</body>
```

그림 4-2 입력 양식 생성 코드와 실행 결과

01 입력 양식 태그

■ 데이터 전달 방식

- <form>태그는 method속성의 방식으로 action 속성 장소에 데이터 전달



그림 4-3 form 태그의 속성

- GET 방식 – 주소에 데이터를 입력해서 전달
- POST 방식 – 주소 변경 없이 비밀스럽게 데이터 전달

```
<!-- GET 전송 방식 -->
<body>
<form method="get">
<input type="text" name="search">
<input type="submit">
</form>
</body>
```

```
<!-- POST 전송 방식 -->
<body>
<form method="post">
<input type="text" name="search">
<input type="submit">
</form>
</body>
```

rint 제출

그림 4-4 입력 양식을 사용한 데이터 요청 예

01 입력 양식 태그

■ 입력 양식 종류

표 4-1 입력 양식 태그

| 태그 | 속성 | 모양 | 설명 |
|----------|-----------|--|--|
| form | 보이지 않음 | | 입력 양식의 시작과 끝 표시 |
| input | text | <input type="text" value="text"/> | 글자 입력 양식 생성 |
| | button | <input type="button" value="button"/> | 버튼 생성 |
| | checkbox | <input checked="" type="checkbox"/> | 체크 박스 생성 |
| | file | <input type="file" value="파일 선택 선택된 파일 없음"/> | 파일 입력 양식 생성 |
| | hidden | 보이지 않음 | 해당 내용 표시 안 함 |
| | image | | 이미지 형태 생성 |
| | password | <input type="password" value="....."/> | 비밀번호 입력 양식 생성 |
| | radio | <input checked="" type="radio"/> | 라디오 버튼 생성 |
| | reset | <input type="reset" value="reset"/> | 초기화 버튼 생성 |
| | submit | <input type="submit" value="submit"/> | 제출 버튼 생성 |
| textarea | cols/rows | <input type="text" value="textarea"/> | 여러 행의 글자 입력 양식 생성, cols는 너비를 지정하고 rows는 높이를 지정 |

01 입력 양식 태그

■ 입력 양식 종류

| 태그 | 속성 | 모양 | 설명 |
|----------|----|---|-----------------|
| select | |  | 선택 양식 생성 |
| optgroup | | | 옵션 그룹화 |
| option | | | 옵션 생성 |
| fieldset | | <input type="text" value="입력 양식"/> | 입력 양식의 그룹 지정 |
| legend | | | 입력 양식 그룹의 이름 지정 |

01 입력 양식 태그

■ [기본 예제 4-1] 기본 입력 양식 태그

[코드 데모](#)

- <input> 태그의 type 속성을 이용해 다양한 기본 입력 양식 생성

코드 4-1 form_basic.html

```
<body>
  <form>
    <!-- 사용자가 입력하는 입력 양식 -->
    <input type="text" name="text" value="text"><br>
    <input type="password" name="password" value="password"><br>
    <input type="file" name="file" value="file"><br>
    <input type="checkbox" name="checkbox" value="checkbox"><br>
    <input type="radio" name="radio" value="radio"><br>

    <!-- 보이지 않는 입력 양식 -->
    <input type="hidden" name="hidden" value="hidden"><br>

    <!-- 버튼 -->
    <input type="button" value="button"><br>
    <input type="reset" value="reset"><br>
    <input type="submit" value="submit"><br>
    <input type="image" src="http://placehold.it/100x100">
  </form>
</body>
```



01 입력 양식 태그

■ [기본 예제 4-2] 간단한 입력 양식 생성

[코드 데모](#)

- 라디오 버튼의 name 속성을 이용해 여러 대상 중 하나만 선택하는 형태

코드 4-3 form_radio.html

```
<body>
  <form>
    <table>
      <tr>
        <td><label for="username">이름</label></td>
        <td><input id="username" type="text" name="username"></td>
      </tr>
      <tr>
        <td>성별</td>
        <td>
          <input id="man" type="radio" name="gender" value="m">
          <label for="man">남자</label>
          <input id="woman" type="radio" name="gender" value="w">
          <label for="woman">여자</label>
        </td>
      </tr>
    </table>
    <input type="submit" value="가입">
  </form>
</body>
```

radio 버튼은 name 속성을 같게 입력해야
여러 항목 중 하나만 선택됩니다.

이름

성별 남자 여자

가입

01 입력 양식 태그

■ [기본 예제 4-3] 선택 가능한 입력 양식

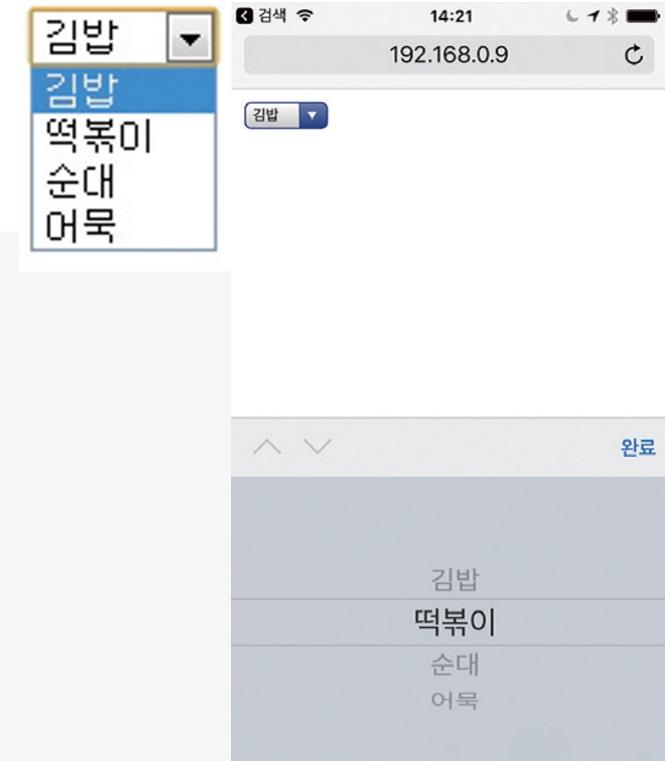
[코드 데모](#)

1. 한 항목만 선택하기 : <select> 태그 이용

- 목록으로 보여 주는 항목 중 하나 또는 여러 개를 선택할 때 사용
- 기본적으로 한번만 선택 가능
- <optgroup>, <option> 태그를 함께 사용

코드 4-4 form_selectBasic.html

```
<body>
  <select>
    <option>김밥</option>
    <option>떡볶이</option>
    <option>순대</option>
    <option>어묵</option>
  </select>
</body>
```



01 입력 양식 태그

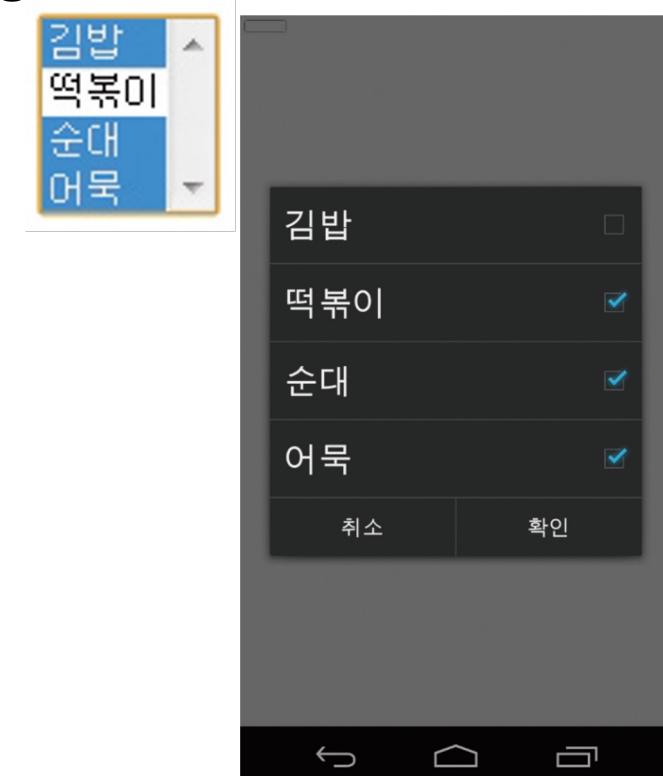
■ [기본 예제 4-3] 선택 가능한 입력 양식

[코드 데모](#)

2. 여러 항목 선택하기 : <select> 태그의 multiple 속성 사용

코드 4-5 form_selectMulti.html

```
<body>
  <select multiple="multiple">
    <option>김밥</option>
    <option>떡볶이</option>
    <option>순대</option>
    <option>어묵</option>
  </select>
</body>
```



01 입력 양식 태그

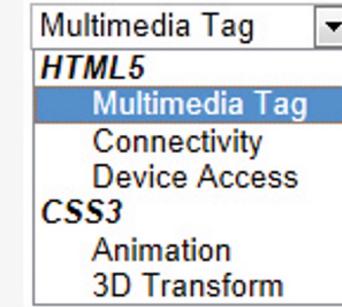
■ [기본 예제 4-3] 선택 가능한 입력 양식

[코드 데모](#)

3. 선택 옵션 묶기 : <optgroup> 태그 사용

코드 4-6 form_selectGroup.html

```
<body>
  <select>
    <optgroup label="HTML5">
      <option>Multimedia Tag</option>
      <option>Connectivity</option>
      <option>Device Access</option>
    </optgroup>
    <optgroup label="CSS3">
      <option>Animation</option>
      <option>3D Transform</option>
    </optgroup>
  </select>
</body>
```



01 입력 양식 태그

■ [기본 예제 4-4] 연관 있는 입력 양식 그룹으로 묶기

[코드 데모](#)

- <fieldset> 태그, <legend> 태그

코드 4-7 form_group.html

```
<body>
  <form>
    <fieldset>
      <legend>입력 양식</legend>
      <table>
        <tr>
          <td><label for="name">이름</label></td>
          <td><input id="name" type="text"></td>
        </tr>
        <tr>
          <td><label for="mail">이메일</label></td>
          <td><input id="mail" type="email"></td>
        </tr>
      </table>
      <input type="submit">
    </fieldset>
  </form>
</body>
```

입력 양식

| | |
|-----|-----------------------|
| 이름 | <input type="text"/> |
| 이메일 | <input type="email"/> |

제출

02 HTML5 문서 구조화

■ 공간 분할 태그

- CSS로 원하는 레이아웃을 구성하기 위해 공간 분할

```
171 <!-- 자동완성 레이어 -->
172 <div id="autoFrame" class="neatmp" style="background-color:rgb(255, 255, 255);display:none;">
173   <div class="api_atom_p_wrap_atom" style="display:none;">
174     <div class="words_words">
175       <div class="atom_p_result_wrap">
176         <ul class="resultBox"></ul>
177         <ul class="resultBox"></ul>
178         <ul class="resultBox"></ul>
179         <ul class="resultBox"></ul>
180       </div>
181     <!-- 주축 정답형 영역 -->
182     <div class="add_group _atom_p_answer_wrap"></div>
183   </div>
184   <!-- 컨텍스트 자동완성 플러스 -->
185   <!-- [AI] _plus -->
186   <div class="atom_p_plus_plus">
187     <span class="desc">
188       <span class="plus_txt _plusTxt">시간대와 관심사에 맞춘 <em class='txt'>컨텍스트 자동완성</em</span>
189       <a onclick="__atompCR(event, this, 'plus.help', ' ', '')" href="https://help.naver.com/support/contents/contents.nhn?serviceNo=806&categoryNo=16859" target="_blank" class="spat ico_info _plusHelp"><span class="blind">도움말 보기</span></a>
190     </span>
191     <!-- [AI] _plus_btn -->
192     <span class="switch _plus_btn">
193       <a href="#" class="btn_turmon active" onclick="__atompCR(event, this, 'plus.use', ' ', '')">ON<span class="blind">선택됨</span></a>
194       <a href="#" class="btn_turnoff" onclick="__atompCR(event, this, 'plus.unuse', ' ', '')">OFF</a>
195     </span>
196     <div class="layer_plus_plusAlert">
197       <strong class="tit">컨텍스트 자동완성</strong>
198       <div class="logout" style="display:block;">
199         <span class="desc"><em class="txt" style="color:#0000ff;">종료한 시간대/언어/남녀별</em> 사용자 그룹의<br>관심사에 맞춰 자동완성을 제공합니다.</span>
200       <div class="btn_area">
201         <a onclick="__atompCR(event, this, 'plus.login', ' ', '')" href="https://nid.naver.com/nidlogin.login" class="btn btn_login">로그인</a>
202         <a target="_blank" onclick="__atompCR(event, this, 'plus.detail', ' ', '')" href="https://help.naver.com/support/alias/search/word/word_16_never" class="btn btn_view">자세히</a>
203       </div>
204       <div class=" _login" style="display:none;">
205         <p class="desc">ON/OFF 설정은<br>해당 기기(브라우저)에 저장됩니다.</p>
206         <div class="btn_area">
207           <a target="_blank" onclick="__atompCR(event, this, 'plus.detail', ' ', '')" href="https://help.naver.com/support/contents/contents.nhn?serviceNo=806&categoryNo=16859" class="btn btn_view">자세히</a>
208         </div>
209       </div>
210     </div>
```

그림 4-5 네이버 메인 페이지의 div 태그

02 HTML5 문서 구조화

■ HTML5의 대표적인 공간 분할 태그

- <div> 태그, 태그

표 4-2 기본 공간 분할 태그

| 태그 | 설명 |
|------|----------------|
| div | 블록 형식으로 공간 분할 |
| span | 인라인 형식으로 공간 분할 |

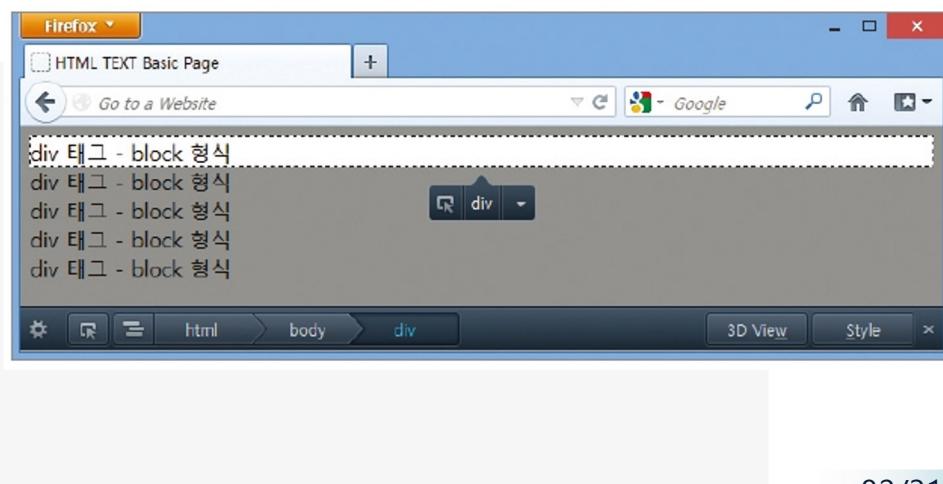
■ [기본 예제 4-5] 공간 분할 방법

[코드 데모](#)

1. 공간을 블록 형식으로 분할하기

코드 4-9 space_block.html

```
<body>
    <div>div 태그 - block 형식</div>
    <div>div 태그 - block 형식</div>
</body>
```



02 HTML5 문서 구조화

■ [기본 예제 4-5] 공간 분할 방법

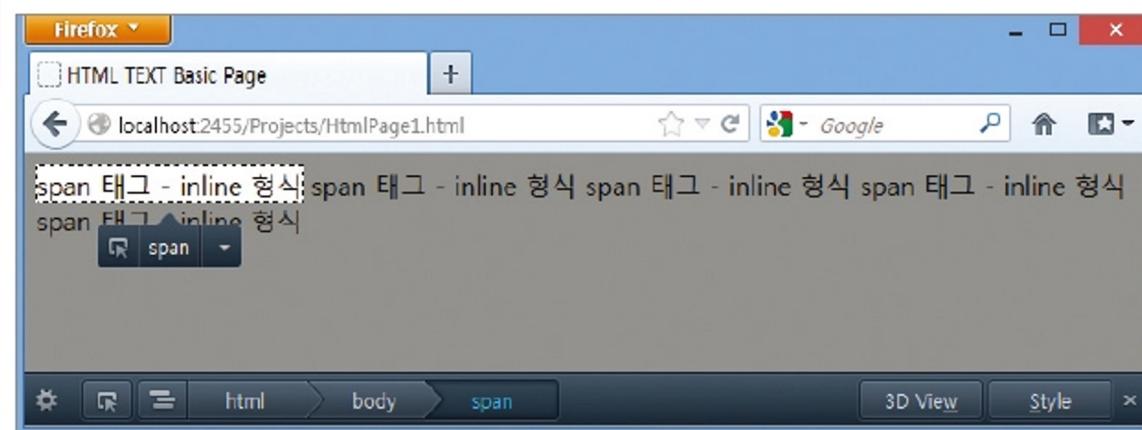
[코드 데모](#)

2. 공간을 인라인 형식으로 분할하기

- 자신의 글자 크기만큼 영역 차지
- 왼쪽에서 오른쪽으로 쌓임

코드 4-10 space_inline.html

```
<body>
  <span>span 태그 - inline 형식</span>
  <span>span 태그 - inline 형식</span>
</body>
```

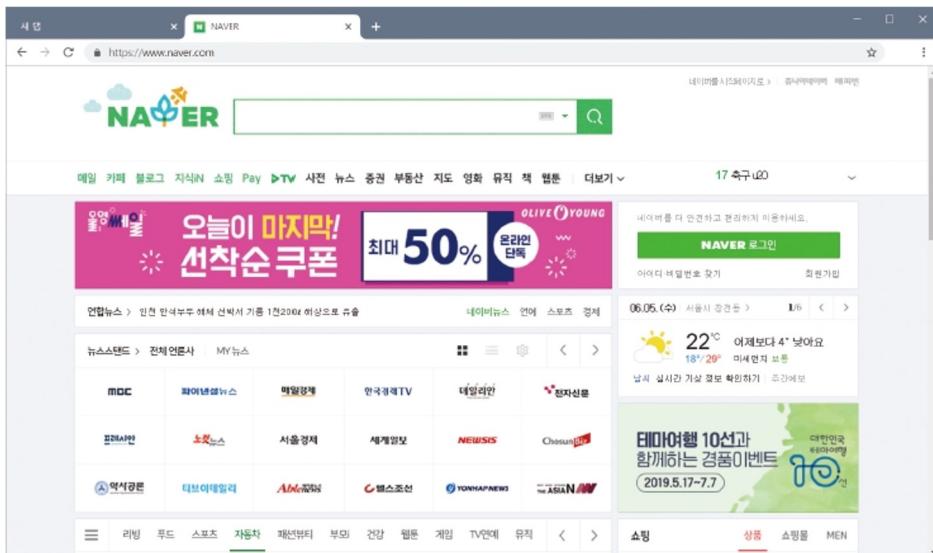


02 HTML5 문서 구조화

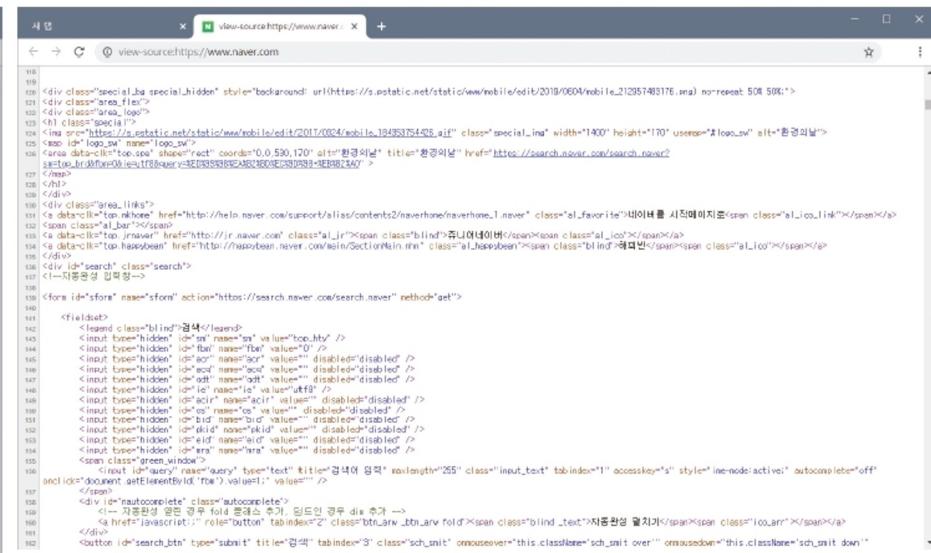
■ 시맨틱 태그

▪ 시맨틱 웹

- 특정 태그에 의미를 부여한 웹
 - 프로그램이 코드를 읽고 의미를 인식할 수 있는 지능형 웹



(a) 사람이 보는 네이버 메인 페이지



(b) 컴퓨터가 보는 네이버 메인 페이지

그림 4-6 시맨틱 웹의 필요성

02 HTML5 문서 구조화

■ 시맨틱 태그

- 시맨틱 태그를 사용한 시맨틱 웹 구현

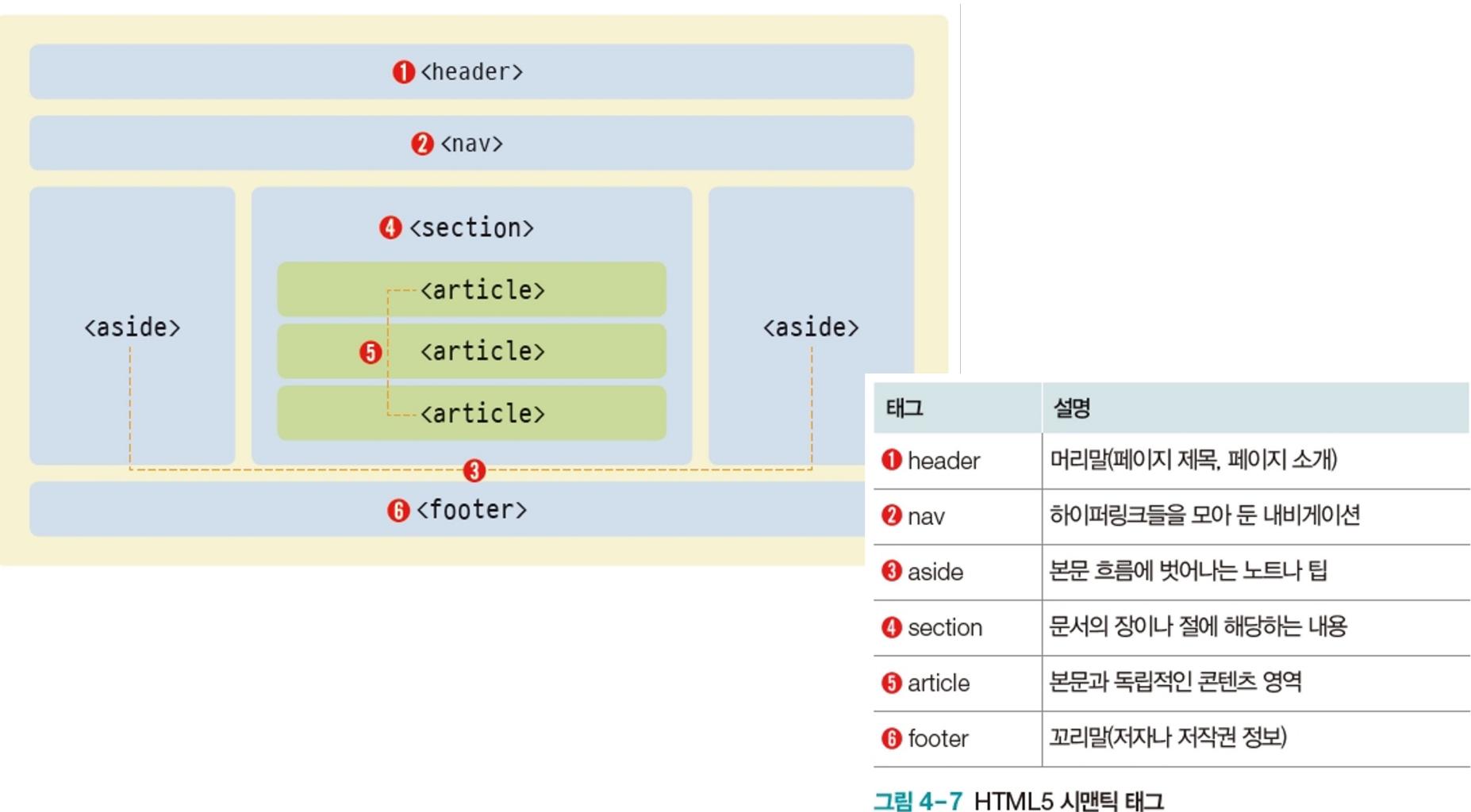


그림 4-7 HTML5 시맨틱 태그

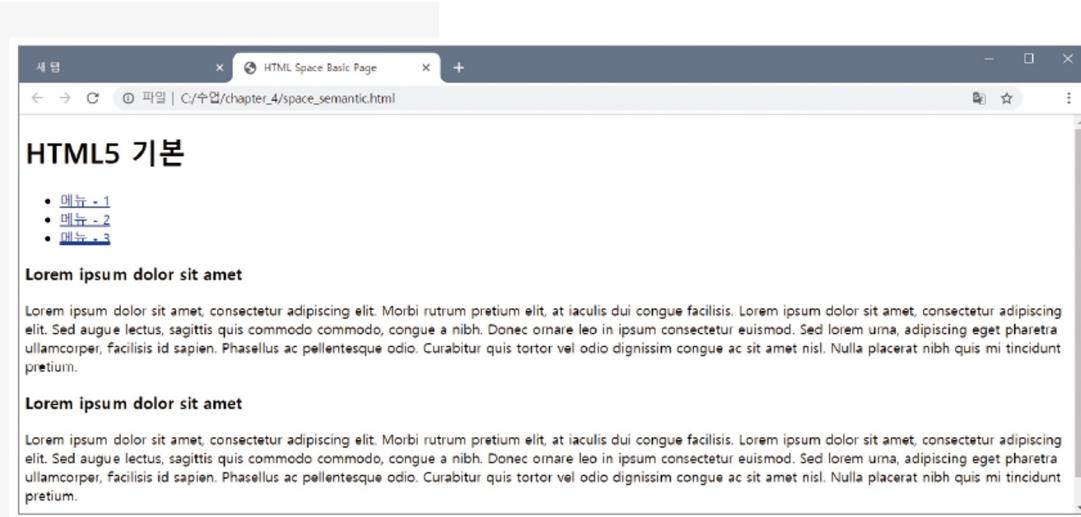
02 HTML5 문서 구조화

■ [기본 예제 4-6] 시맨틱 태그를 이용한 레이아웃 구성

[코드 데모](#)

코드 4-13 space_semantic.html

```
<body>
  <header>
    <h1>HTML5 기본</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">메뉴 - 1</a></li>
      <li><a href="#">메뉴 - 2</a></li>
      <li><a href="#">메뉴 - 3</a></li>
    </ul>
  </nav>
  <section>
    <article>
      <h1>Lorem ipsum dolor sit amet</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </article>
    <article>
      <h1>Lorem ipsum dolor sit amet</h1>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </article>
  </section>
  <footer>
    <address>서울특별시 강서구 마곡동</address>
  </footer>
</body>
```



01 선택자의 용도와 사용법

■ CSS 개요

- 선택자 - CSS3에서 특정 HTML 태그를 선택할 때 사용

h1 { color: red; }

선택자

스타일 속성

스타일 값

그림 5-1 CSS 블록

■ 스타일시트

- CSS로 작성된 코드
- <style>..</style>

코드 5-1 selector_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        h1 {
            color: red;
            background-color: orange;
        }
    </style>
</head>
<body>
    <h1>CSS3 선택자 기본</h1>
</body>
</html>
```

01 선택자의 용도와 사용법

■ CSS 선택자 종류

표 5-1 CSS3 선택자 종류

종류	형태	사용 예
전체 선택자	*	*
태그 선택자	태그	h1
아이디 선택자	#아이디	#id
클래스 선택자	.클래스	.header
속성 선택자	선택자[속성 = 값]	input[type = text]
	선택자[속성 ~= 값]	div[data-role ~= row]
	선택자[속성 != 값]	div[data-role != row]
	선택자[속성 ^= 값]	div[data-role ^= row]
	선택자[속성 \$= 값]	div[data-role \$= 9]
	선택자[속성 *= 값]	div[data-role *= row]
후손 선택자	선택자 선택자	header h1

01 선택자의 용도와 사용법

■ CSS 선택자 종류

종류	형태	사용 예
자손 선택자	선택자 > 선택자	header > h1
반응 선택자	선택자:active	div:active
	선택자:hover	div:hover
상태 선택자	선택자:checked	input:checked
	선택자:focus	input:focus
	선택자:enabled	input:enabled
	선택자:disabled	input:disabled
구조 선택자	선택자:first-child	li:first-child
	선택자:last-child	li:last-child
	선택자:nth-child(수열)	li:nth-child(2n+1)
	선택자:nth-last-child(수열)	li:nth-last-child(2n+1)
	선택자:first-of-type	h1:first-of-type
	선택자:last-of-type	h1:last-of-type
	선택자:nth-of-type(수열)	h1:nth-of-type(2n+1)
	선택자:nth-last-of-type(수열)	h1:nth-last-of-type(2n+1)

01 선택자의 용도와 사용법

■ CSS 선택자 종류

종류	형태	사용 예
동위 선택자	선택자 + 선택자	h1 + div
	선택자 ~ 선택자	h1 ~ div
링크 선택자	선택자:link	a:link
	선택자:visited	a:visited
문자 선택자	선택자::first-letter	p::first-letter
	선택자::first-line	p::first-line
	선택자::after	p::after
	선택자::before	p::before
부정 선택자	선택자::selection	p::selection
	선택자:not(선택자)	li:not(.item)

02 기본 선택자

■ 기본 선택자

표 5-2 기본 선택자

종류	형태	설명
전체 선택자	*	HTML 페이지 내부의 태그를 모두 선택
태그 선택자	태그	HTML 페이지 내부의 특정 태그를 모두 선택
아이디 선택자	#아이디	특정 id 속성이 있는 태그 선택. 웹 표준에 id 속성은 웹 페이지 내부에서 중복되면 안 된다는 규정이 있으므로 아이디 선택자는 특정 태그 하나를 선택할 때 사용
클래스 선택자	.클래스	특정 클래스가 있는 태그 선택

02 기본 선택자

■ [기본 예제 5-1] 전체 선택자와 태그 선택자

[코드 데모](#)[온라인 에디터](#)

1. 전체 선택자 적용하기

코드 5-2 default_wildcard.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        * { color: red; }
    </style>
</head>
<body>
    <h1>제목 글자 태그</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

제목 글자 태그

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer sagittis enim et felis fringilla sit amet vehicula enim tincidunt. Duis turpis velit, auctor eget porta a, consectetur euismod metus. Mauris auctor, augue sed elementum condimentum, sem felis vulputate ligula, ac commodo massa turpis et massa.

02 기본 선택자

■ [기본 예제 5-1] 전체 선택자와 태그 선택자

[코드 데모](#)

2. 태그 선택자 적용하기

코드 5-3 default_tag.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic Page</title>
    <style>
        h1 { color: red; }
        p { color: blue; }
    </style>
</head>
<body>
    <h1>제목 글자</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p>Nunc nisl turpis, aliquet et gravida non, facilisis a sem.</p>
</body>
</html>
```

제목 글자

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer sagittis enim et felis fringilla sit amet vehicula enim tincidunt. Duis turpis velit, auctor eget porta a, consectetur euismod metus. Mauris auctor, augue sed elementum condimentum, sem felis vulputate ligula, ac commodo massa turpis et massa.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer sagittis enim et felis fringilla sit amet vehicula enim tincidunt. Duis turpis velit, auctor eget porta a, consectetur euismod metus. Mauris auctor, augue sed elementum condimentum, sem felis vulputate ligula, ac commodo massa turpis et massa.

■ 선택자 여러 개에 속성 적용하기: 쉼표 사용

```
<style>
body, p, h1, h2, h3, h4, h5, h6 { margin: 0; padding: 0; }
</style>
```

그림 5-2 선택자를 여러 개 사용

02 기본 선택자

■ [기본 예제 5-2] 아이디 선택자

[코드 데모](#)

코드 5-4 default_id.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        #header {
            width: 800px; margin: 0 auto;
            background: red;
        }
        #wrap {
            width: 800px; margin: 0 auto;
            overflow: hidden;
        }
        #aside {
            width: 200px; float: left;
            background: blue;
        }
        #content {
            width: 600px; float: left;
            background: green;
        }
    </style>
</head>
```

02 기본 선택자

■ [기본 예제 5-2] 아이디 선택자

[코드 데모](#)

```
<body>
  <div id="header">
    <h1>#header 태그</h1>
  </div>
  <div id="wrap">
    <div id="aside">
      <h1>#aside 태그</h1>
    </div>
    <div id="content">
      <h1>#content 태그</h1>
    </div>
  </div>
</body>
</html>
```

#header 태그

#aside 태그 #content 태그

02 기본 선택자

■ [기본 예제 5-3] 클래스 선택자

[코드 데모](#)

1. 클래스 사용자를 1개 사용하기

코드 5-6 default_class.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        .select { color: red; }
    </style>
</head>
<body>
    <ul>
        <li class="select">사과</li>
        <li>바나나</li>
        <li class="select">오렌지</li>
        <li>감</li>
    </ul>
</body>
</html>
```

- 사과
- 바나나
- 오렌지
- 감

02 기본 선택자

- [기본 예제 5-3] 클래스 선택자
 - 2. 클래스 선택자를 여러 개 사용하기

[코드 데모](#)

코드 5-7 default_multiClass.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        .item { color: red; }
        .header { background-color: blue; }
    </style>
</head>
<body>
    <h1 class="item header">동해물과 백두산이</h1>
</body>
</html>
```

두 가지 속성이 적용되어 파란색 배경에
빨간색 글자로 나타납니다.

03 속성 선택자

■ 속성 선택자

표 5-3 속성 선택자

형태	설명
선택자[속성]	특정한 속성이 있는 태그 선택
선택자[속성 = 값]	특정한 속성 내부 값이 특정 값과 같은 태그 선택

■ [기본 예제 5-4] 속성 선택자

[코드 데모](#)

코드 5-9 attribute_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        input[type="text"] { background: red; }
        input[type="password"] { background: blue; }
    </style>
</head>
<body>
    <form>
        <input type="text">
        <input type="password">
    </form>
</body>
</html>
```



04 후손 선택자와 자손 선택자

- “자손은 무엇이고 또 후손은 무엇인가요?”

코드 5-10 desc_basic.html

```
<body>
  <div>
    <h1>CSS3 선택자 기본</h1>
    <h2>기본 선택자</h2>
    <ul>
      <li>전체 선택자</li>
      <li>태그 선택자</li>
      <li>아이디 선택자와 클래스 선택자</li>
    </ul>
  </div>
</body>
```

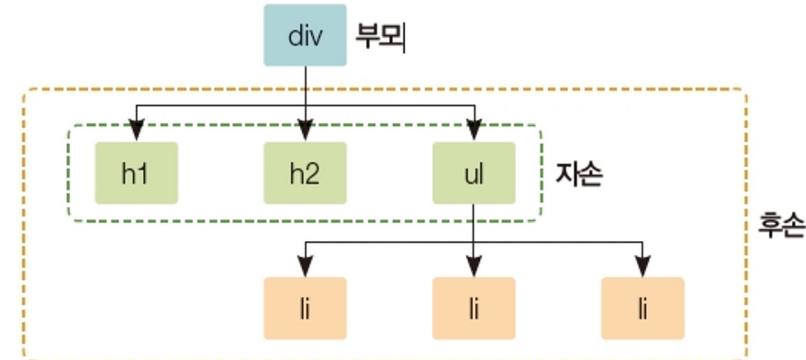


그림 5-3 자손 선택자와 후손 선택자

- 후손 선택자

표 5-4 후손 선택자

형태	설명
선택자A 선택자B	선택자A의 후손인 선택자B 선택

04 후손 선택자와 자손 선택자

■ [기본 예제 5-5] 후손 선택자

[코드 데모](#)

코드 5-11 desc_descendants.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        #header h1 { color: red; }
        #section h1 { color: orange; }
    </style>
</head>
<body>
    <div id="header">
        <h1 class="title">Lorem ipsum</h1>
        <div id="nav">
            <h1>Navigation</h1>
        </div>
    </div>
    <div id="section">
        <h1 class="title">Lorem ipsum</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
</body>
</html>
```

Lorem ipsum

Navigation

Lorem ipsum

04 후손 선택자와 자손 선택자

- **후손 선택자 여러 개를 함께 사용할 경우 주의 사항**

- ① <header> 태그의 후손인 <h1> 태그와 일반적인 <h2> 태그 선택

```
❶ <style>
    #header h1, h2 { color: red; }
</style>
```

- ② <header> 태그의 후손인 <h1> 태그와 header 태그의 후손인 <h2> 태그 선택

```
❷ <style>
    #header h1, #header h2 { color: red; }
</style>
```

그림 5-4 후손 선택자 주의 사항

04 후손 선택자와 자손 선택자

■ 자손 선택자 표 5-5 자손 선택자

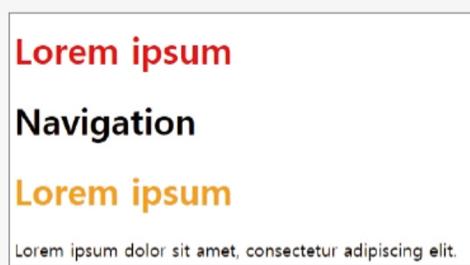
형태	설명
선택자A > 선택자B	선택자A의 자손인 선택자B 선택

■ [기본 예제 5-6] 자손 선택자

[코드 데모](#)

코드 5-12 desc_children.html

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    #header > h1 { color: red; }
    #section > h1 { color: orange; }
  </style>
</head>
<body>
  <div id="header">
    <h1 class="title">Lorem ipsum</h1>
    <div id="nav">
      <h1>Navigation</h1>
    </div>
  </div>
  <div id="section">
    <h1 class="title">Lorem ipsum</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
</html>
```



04 후손 선택자와 자손 선택자

- [기본 예제 5-6] <table> 태그 요소 선택할 때 자손 선택자 주의사항

[코드 데모](#)

코드 5-13 desc_tableWithChildren.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        table > tr > th {
            color: red;
        }
    </style>
</head>

<body>
    <table border="1">
        <tr>
            <th>이름</th>
            <th>지역</th>
        </tr>
        <tr>
            <td>윤인성</td>
            <td>서울특별시 강서구 내발산동</td>
        </tr>
    </table>
</body>
</html>
```

이름	지역
윤인성	서울특별시 강서구 내발산동

04 후손 선택자와 자손 선택자

■ 검사를 통한 HTML 페이지의 계층 구조

- 웹 브라우저가 <tbody> 태그를 자동으로 추가하므로 스타일 속성이 적용되지 않음
- table 태그에 스타일을 적용할 때는 자손 선택자를 사용하지 않음

```
▼<table border="1">
  ▼<tbody>
    ▼<tr>
      <th>이름</th>
      <th>지역</th>
    </tr>
    ▼<tr>
      <td>윤민성</td>
      <td>서울특별시 강서구 내발산동</td>
    </tr>
  </tbody>
</table>
```

그림 5-5 검사 결과

05 반응 · 상태 · 구조 선택자

■ 반응 선택자

표 5-6 반응 선택자

형태	설명
:active	사용자가 마우스로 클릭한 태그 선택
:hover	사용자가 마우스 커서를 올린 태그 선택

■ [기본 예제 5-7] 반응 선택자

[코드 데모](#)

코드 5-14 etc_action.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        h1:hover { color: red; }
        h1:active { color: blue; }
    </style>
</head>
<body>
    <h1>반응 선택자</h1>
</body>
</html>
```

반응 선택자
반응 선택자
반응 선택자

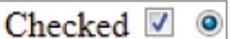
05 반응 · 상태 · 구조 선택자

■ 상태 선택자

표 5-7 상태 선택자

형태	설명
:checked	체크 상태의 input 태그 선택
:focus	포커스를 맞춘 input 태그 선택
:enabled	사용 가능한 input 태그 선택
:disabled	사용 불가능한 input 태그 선택

표 5-8 상태 선택자의 상태

상태	설명
 	type 속성이 checkbox 또는 radio인 input 태그가 선택된 상태
	사용자가 바로 입력할 수 있도록 입력 양식에 포커스를 둔 상태. 웹 페이지 하나당 input 태그 하나에만 포커스를 둘 수 있음
 	input 태그에 값을 입력할 수 있는 상태고 disabled는 input 태그에 값을 입력할 수 없는 상태. input 태그에 disabled 속성을 입력해 적용 <pre><body> <input value="enabled"> <input value="disabled" disabled="disabled"> </body></pre>

05 반응 · 상태 · 구조 선택자

■ [기본 예제 5-8] 상태 선택자

[코드 데모](#)

코드 5-15 etc_state.html

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS3 Selector Basic</title>
  <style>
    /* input 태그가 사용 가능할 때
       background-color 속성에 white 키워드를 적용합니다. */
    input:enabled { background-color: white; }
    /* input 태그가 사용 불가능할 때
       background-color 속성에 gray 키워드를 적용합니다. */
    input:disabled { background-color: gray; }
    /* input 태그에 포커스를 맞출 때
       background-color 속성에 orange 키워드를 적용합니다. */
    input:focus { background-color: orange; }
  </style>
</head>
<body>
  <h2>사용 가능</h2>
  <input>
  <h2>사용 불가능</h2>
  <input disabled="disabled">
</body>
</html>
```

사용 가능

사용 불가능

05 반응 · 상태 · 구조 선택자

■ 구조 선택자

표 5-9 구조 선택자

형태	설명
:first-child	형제 관계에서 첫 번째로 등장하는 태그 선택
:last-child	형제 관계에서 마지막으로 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

- nth-child와 nth-lastchild 선택자
 - $2n+1$ 수열에 0부터 숫자를 넣을 경우... "1, 3, 5, 7, 9.....?"
 - 구조 선택자는 수열의 결과 값에 해당하는 위치에 있는 태그에 스타일을 적용

05 반응 · 상태 · 구조 선택자

■ [기본 예제 5-9] 구조 선택자

[코드 데모](#)

코드 5-16 etc_structure.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        ul { overflow: hidden; }
        li {
            list-style: none;
            float: left; padding: 15px;
        }
        li:first-child { border-radius: 10px 0 0 10px; }
        li:last-child { border-radius: 0 10px 10px 0; }

        li:nth-child(2n) { background-color: #FF0003; }
        li:nth-child(2n+1) { background-color: #800000; }
    </style>
</head>
<body>
    <ul>
        <li>첫 번째</li>
        <li>두 번째</li>
        <li>세 번째</li>
        <li>네 번째</li>
        <li>다섯 번째</li>
        <li>여섯 번째</li>
        <li>일곱 번째</li>
    </ul>
</body>
</html>
```

첫 번째 두 번째 세 번째 네 번째 다섯 번째 여섯 번째 일곱 번째

05 반응 · 상태 · 구조 선택자

■ [기본 예제 5-9] 구조 선택자

[코드 데모](#)

- 구조 선택자 사용시 주의 사항

코드 5-17 etc_structureFalse.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Selector Basic</title>
    <style>
        li > a:first-child { color: red; }
    </style>
</head>
<body>
    <ul>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
        <li><a href="#">주의 사항</a></li>
    </ul>
</body>
</html>
```

- 주의 사항

06 CSS3 단위

■ 키워드 단위

- W3C에서 미리 정의한 단어
- 키워드를 입력하면 해당하는 스타일이 자동으로 적용

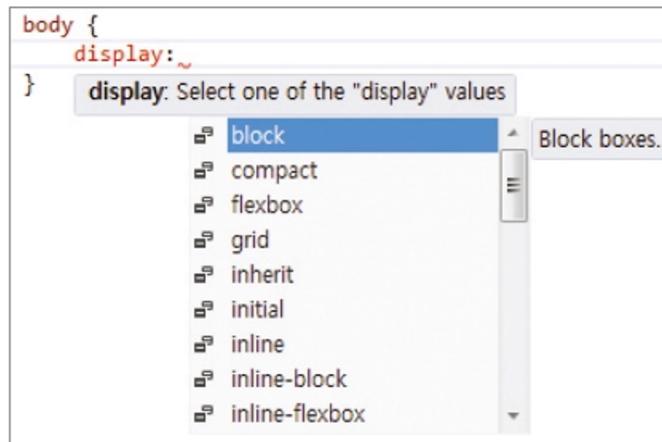


그림 5-6 개발 환경을 이용한 속성별 자동 완성 기능

■ 크기 단위

표 5-10 크기 단위

단위	설명
%	백분율 단위
em	배수 단위
px	픽셀 단위

■ [기본 예제 5-10] 다양한 크기의 단위 적용

1. HTML 페이지 만들기

코드 5-19 size_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Style Property Basic</title>
    <style>

        </style>
    </head>
    <body>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

■ [기본 예제 5-10] 다양한 크기의 단위 적용

2. 2% 단위 적용하기

- % 단위는 기본으로 설정된 크기를 기준으로 상대적인 크기를 지정.
- 초기 설정 크기: 100%

코드 5-20 size_percent.html

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

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

06 CSS3 단위

■ [기본 예제 5-10] 다양한 크기의 단위 적용

3. em 단위 적용하기

- 1배=1em=100%
- 1.5배=1.5em=150%
- 초기 설정 크기: 100%

코드 5-21 size_em.html

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

4. px 단위 적용하기

- 크기를 절대적으로 지정

코드 5-22 size_pixel.html

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

06 CSS3 단위

■ 색상 단위

표 5-11 색상 단위

RGB 색상	형태	rgb(red, green, blue)
	설명	R(빨간색)과 G(초록색)와 B(파란색)를 조합해 색상을 표현하며, 0부터 255 사이의 숫자를 입력합니다.
	예	<pre><style> h1 { background-color: rgb(255, 255, 255); } </style></pre>
RGBA 색상	형태	rgba(red, green, blue, alpha)
	설명	RGB 색상 단위에 알파 값을 추가한 형태입니다. 알파 값은 투명도를 나타내며, 0.0부터 1.0 사이의 숫자를 입력합니다. 0.0은 완전 투명 상태고, 1.0은 완전 불투명 상태입니다.
	예	<pre><style> h1 { background-color: rgba(255, 255, 255, 0.5); } </style></pre>
HEX 코드	형태	#000000
	설명	RGB 색상 단위를 짧게 입력하는 방법으로 RGB 색상 조합을 #000000 같은 16진수로 입력합니다.
	예	<pre><style> h1 { background-color: #0094FF; } </style></pre>

06 CSS3 단위

■ 색상 선택 도구

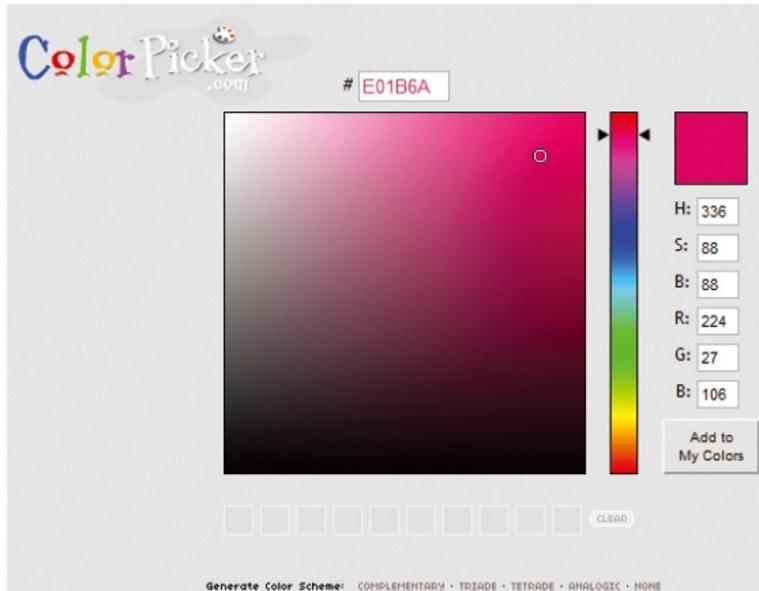


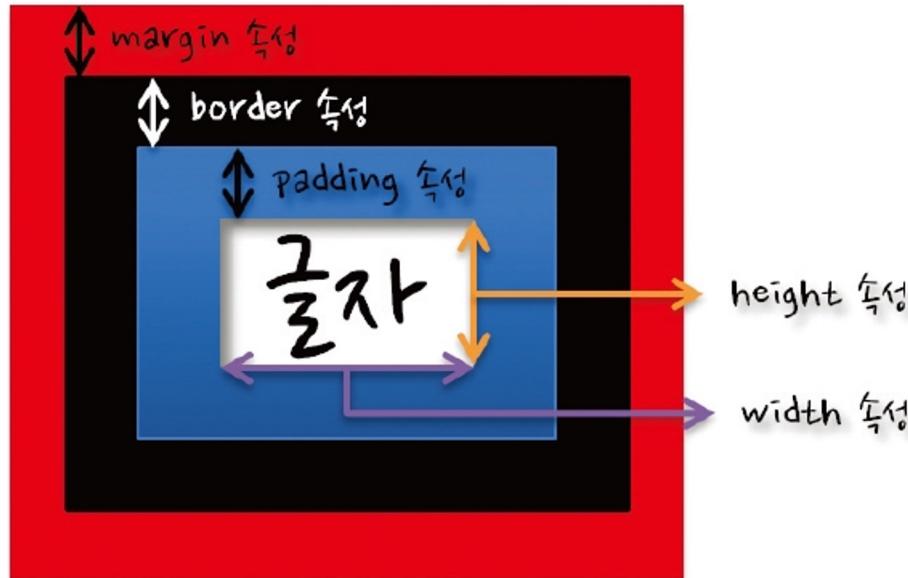
그림 5-7 색상 선택 프로그램 예(<http://www.colorpicker.com/>)

■ URL 단위

- 이미지나 글꼴 파일을 불러올 때 사용
- url('경로')

01 박스 속성

- 웹 페이지의 레이아웃을 구성할 때 가장 중요



margin 속성	테두리와 다른 태그 사이의 테두리 바깥쪽 여백
border 속성	테두리
padding 속성	테두리와 글자 사이의 테두리 안쪽 여백. 배경색은 padding 영역까지만 적용
width 속성	글자를 감싸는 영역의 가로 크기
height 속성	글자를 감싸는 영역의 세로 크기

그림 6-1 박스 속성

01 박스 속성

■ 박스 크기와 여백

- width와 height 속성 : 글자를 감싸는 영역의 크기를 지정
- border 속성 : 테두리 두께 지정
- margin : 테두리와 다른 태그의 간격을 지정
- padding : 테두리와 바깥쪽 여백을 지정

■ [기본 예제 6-1] 박스 크기와 패딩 조정

- 1. width와 height 속성 적용하기

코드 6-1 box_widthHeight.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Property Basic</title>
    <style>
        div {
            width: 100px; height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
    <div></div>
</body>
</html>
```

[코드 데모](#)

[온라인 에디터](#)



01 박스 속성

- [기본 예제 6-1] 박스 크기와 패딩 조정
 - 2. margin과 padding 속성 적용하기

[코드 데모](#)

코드 6-2 box_marginPadding.html

```
<style>
  div {
    width: 100px; height: 100px;
    background-color: red;
    border: 20px solid black;
    margin: 10px; padding: 30px;
  }
</style>
```



01 박스 속성

■ [기본 예제 6-1] 박스 크기와 패딩 조정

- 웹 브라우저에서 검사로 코드 분석

[코드 데모](#)

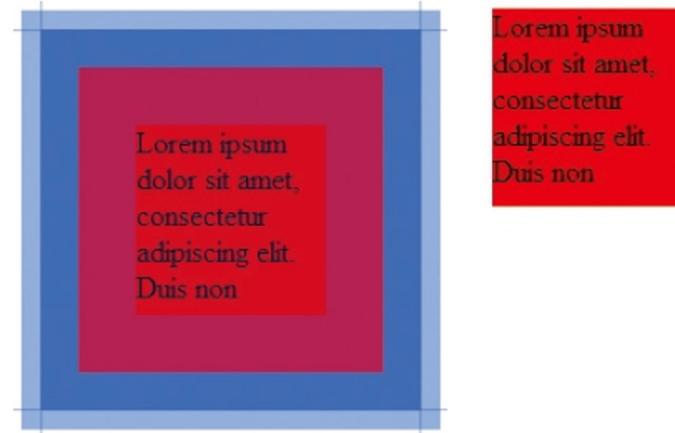


그림 6-2 width와 height 속성의 적용 범위

■ 태그 전체의 크기 구하기

- margin, border, padding 속성은 양쪽에 위치하므로 2를 곱함
- 전체 너비 = width + 2 ×(margin + border + padding)
- 전체 높이 = height + 2 ×(margin + border + padding)

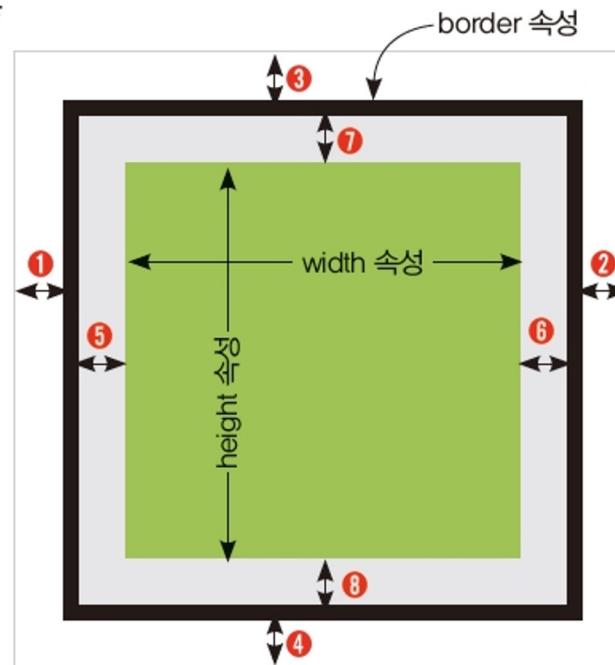
01 박스 속성

■ [기본 예제 6-2] 박스 여백 부분 조정

- margin 속성과 padding 속성은 두 가지 방법으로 상하좌우를 각각 지정할 수 있음

표 6-1 margin 속성과 padding 속성의 방향 적용

속성	설명
① margin-left	왼쪽 여백
② margin-right	오른쪽 여백
③ margin-top	위쪽 여백
④ margin-bottom	아래쪽 여백
⑤ padding-left	왼쪽 패딩
⑥ padding-right	오른쪽 패딩
⑦ padding-top	위쪽 패딩
⑧ padding-bottom	아래쪽 패딩



01 박스 속성

■ [기본 예제 6-2] 박스 여백 부분 조정

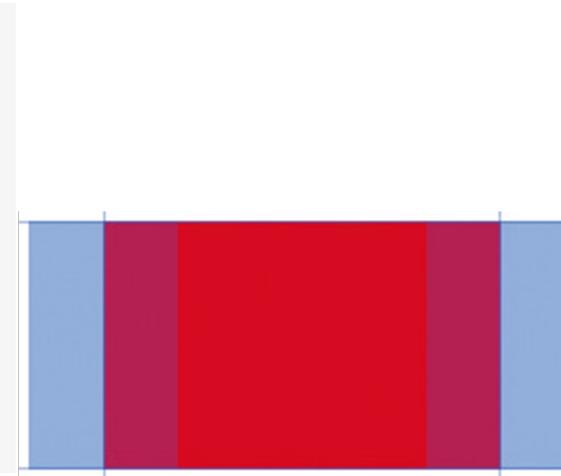
[코드 데모](#)

▪ 1. 네 방향 속성 지정하기 : 순서대로 크기 단위 띄어쓰기

코드 6-3 box_marginPaddingEachFour.html

```
<style>
  div {
    width: 100px; height: 100px;
    background-color: red;

    /* margin: 위쪽 오른쪽 아래쪽 왼쪽 */
    /* padding: 위쪽 오른쪽 아래쪽 왼쪽 */
    margin: 0 30px 0 30px;
    padding: 0 30px 0 30px;
  }
</style>
```

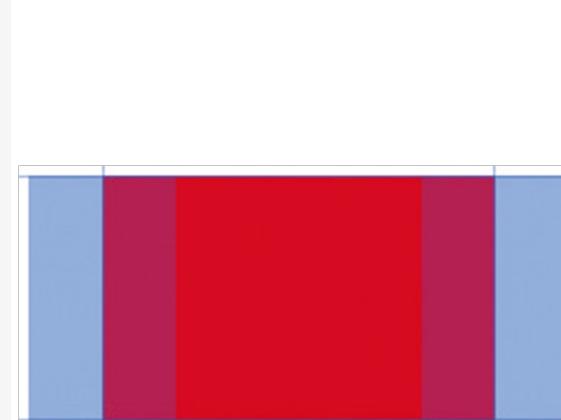


▪ 2. 두 방향씩 속성 지정하기 : 네 방향을 2개씩 묶어 지정

코드 6-4 box_marginPaddingEachTwo.html

```
<style>
  div {
    width: 100px; height: 100px;
    background-color: red;

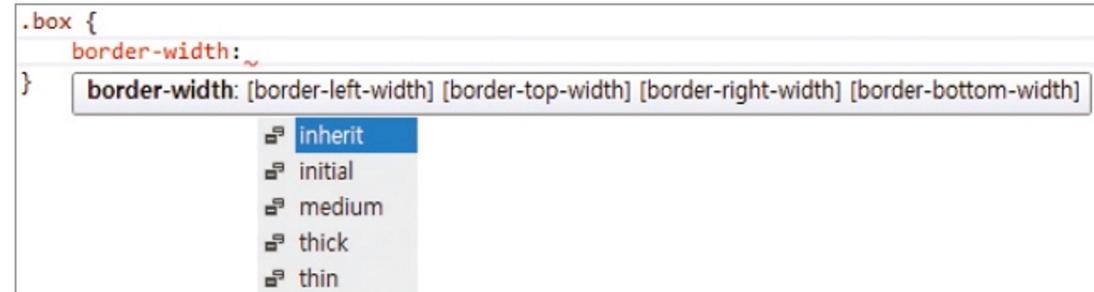
    /* margin: <위아래> <왼쪽오른쪽> */
    /* padding: <위아래> <왼쪽오른쪽> */
    margin: 0 30px;
    padding: 0 30px;
  }
</style>
```



01 박스 속성

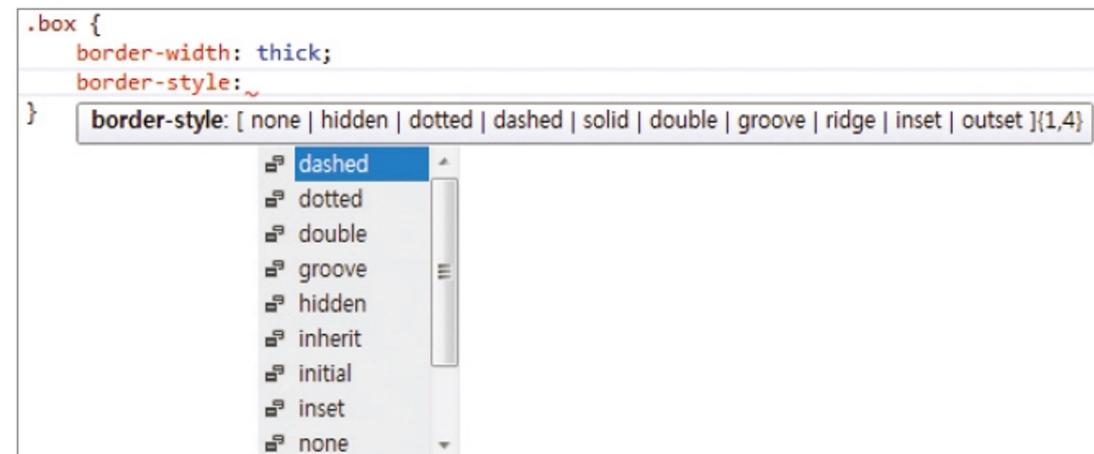
■ 박스 테두리

- border-width : 테두리 두께



(a) border-width 속성: 테두리 두께

- border-style : 테두리 형태
- border-color : 테두리 색상



(b) border-style 속성: 테두리 형태

그림 6-3 테두리의 두께와 형태 지정

01 박스 속성

■ 박스 테두리

- border
 - border-width, border-style, border-color 속성을 한 번에 입력할 수도 있음

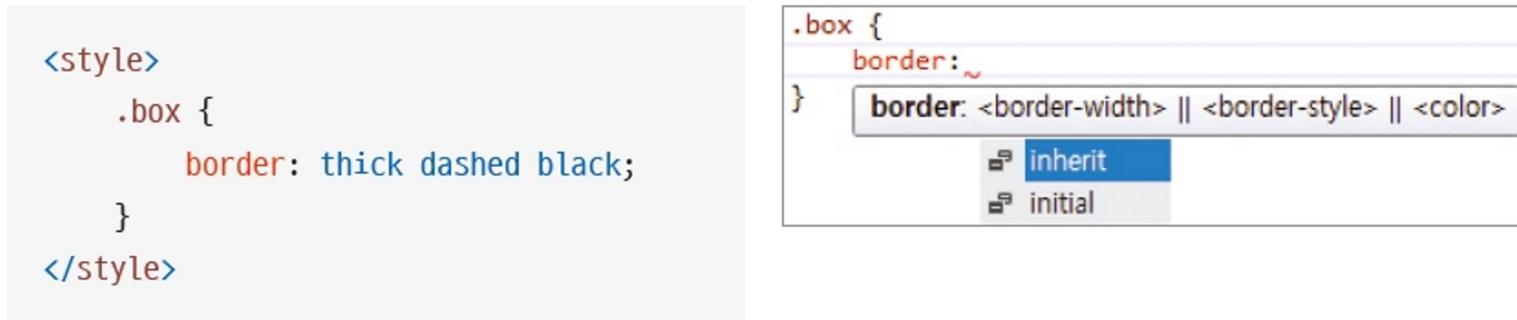


그림 6-4 border 속성: 테두리 두께, 형태, 색상 지정

- 상하좌우 속성을 각각 입력할 수 있음

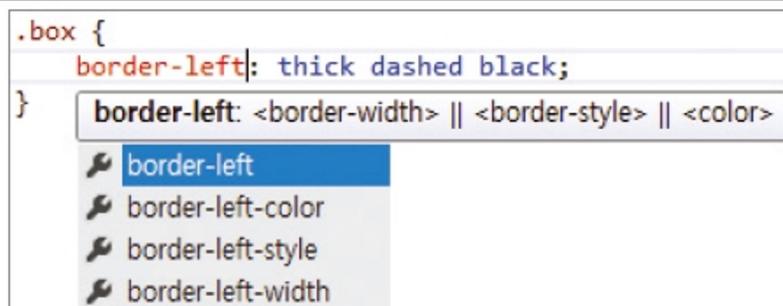


그림 6-5 테두리의 상하좌우 지정

01 박스 속성

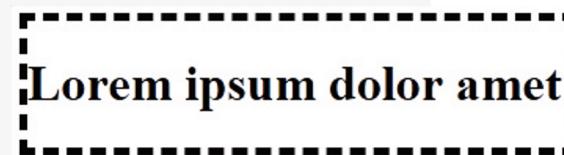
■ [기본 예제 6-3] 박스 여백 부분 조정

[코드 데모](#)

▪ 1. 기본 테두리 만들기

코드 6-5 border_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        .box {
            border-width: thick;
            border-style: dashed;
            border-color: black;
        }
    </style>
</head>
<body>
    <div class="box">
        <h1>Lorem ipsum dolor amet</h1>
    </div>
</body>
</html>
```



01 박스 속성

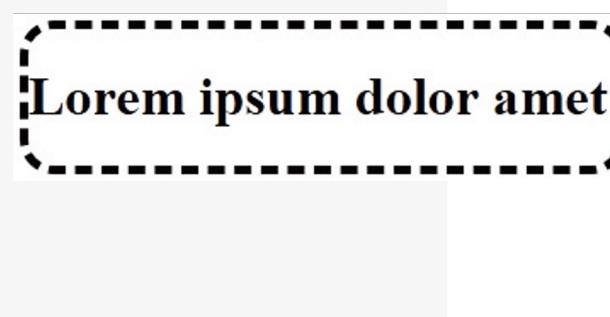
■ [기본 예제 6-3] 박스 여백 부분 조정

[코드 데모](#)

▪ 2. 둥근 테두리 만들기

코드 6-6 border_radius.html

```
<style>
  .box {
    border-width: thick;
    border-style: dashed;
    border-color: black;
    border-radius: 20px;
  }
</style>
```

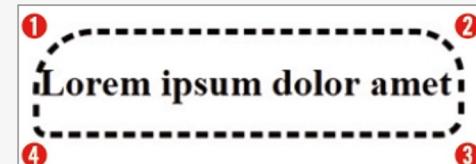


▪ 3. 모서리마다 둥글기를 다르게 적용하기

코드 6-7 border_radiusEach.html

```
<style>
  .box {
    border-width: thick;
    border-style: dashed;
    border-color: black;

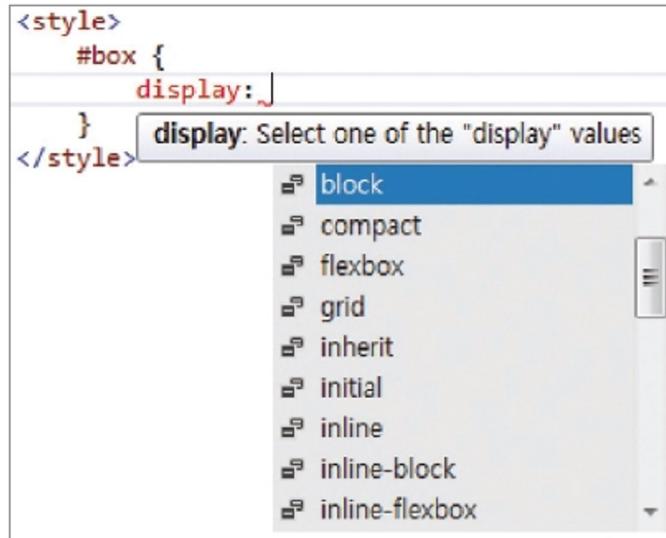
    /* border-radius: "왼쪽 위" "오른쪽 위" "오른쪽 아래" "왼쪽 아래" */
    border-radius: 50px 40px 20px 10px;
  }
</style>
```



02 가시 속성

■ 가시 속성

- 태그가 화면에 보이는 방식을 지정
- display 속성



키워드	설명
none	화면에 보이지 않음
block	블록 박스 형식으로 지정
inline	인라인 박스 형식으로 지정
inline-block	블록과 인라인의 중간 형태로 지정

그림 6-6 display 속성과 주로 사용하는 키워드

■ [기본 예제 6-4] display 속성

[코드 데모](#)

▪ 1. none 키워드 적용하기

- 태그가 화면에서 보이지 않음

코드 6-8 display_none.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Display</title>
    <style>
        #box {
            display: none;
        }
    </style>
</head>
<body>
    <span>더미 객체</span>
    <div id="box">대상 객체</div>
    <span>더미 객체</span>
</body>
</html>
```

더미 객체 더미 객체

02 가시 속성

■ [기본 예제 6-4] display 속성

[코드 데모](#)

▪ 2. block 키워드 적용하기

- #box 태그의 display 속성을 block 키워드로 바꿈

코드 6-9 display_block.html

```
<style>
  #box {
    display: block;
  }
</style>
```



▪ 3. inline과 inline-block 키워드 비교하기

코드 6-10 display_inline.html

```
<style>
  #box {
    display: inline;
  }
</style>
```

코드 6-11 display_inline-block.html

```
<style>
  #box {
    display: inline-block;
  }
</style>
```

A screenshot of a code editor showing three lines of Korean text: "더미 객체", "대상 객체", and "더미 객체". The lines are separated by spaces, demonstrating the effect of inline display.

02 가시 속성

▪ inline과 inline-block 키워드 적용

▪ block 형식, inline-block 형식

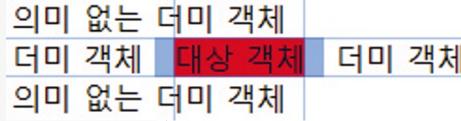
- margin 속성, padding 속성을 상하좌우 네 방향으로 지정할 수 있음

▪ inline 형식

- margin 속성, padding 속성을 왼쪽과 오른쪽으로만 지정할 수 있음

코드 6-12 display_inlineWithMargin.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Display</title>
    <style>
        #box {
            display: inline;
            background-color: red;
            width: 100px; height: 50px;
            margin: 10px;
        }
    </style>
</head>
<body>
    <p>의미 없는 더미 객체</p>
    <span>더미 객체</span>
    <div id="box">대상 객체</div>
    <span>더미 객체</span>
    <p>의미 없는 더미 객체</p>
</body>
</html>
```



코드 6-13 display_inline-blockWithMargin.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Display</title>
    <style>
        #box {
            display: inline-block;
            background-color: red;
            width: 100px; height: 50px;
            margin: 10px;
        }
    </style>
</head>
<body>
    <p>의미 없는 더미 객체</p>
    <span>더미 객체</span>
    <div id="box">대상 객체</div>
    <span>더미 객체</span>
    <p>의미 없는 더미 객체</p>
</body>
</html>
```



03 배경 속성

■ 배경 속성

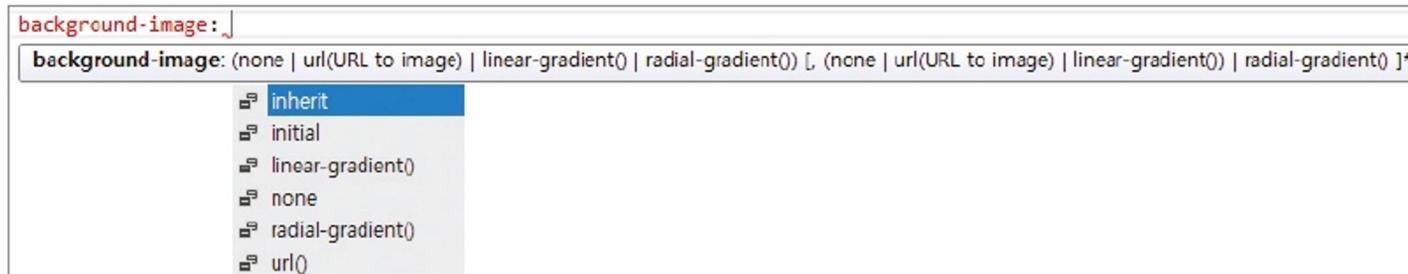
표 6-2 주로 사용하는 배경 속성

속성	설명
background-image	배경 이미지 삽입
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	한 번에 모든 배경 속성 입력

03 배경 속성

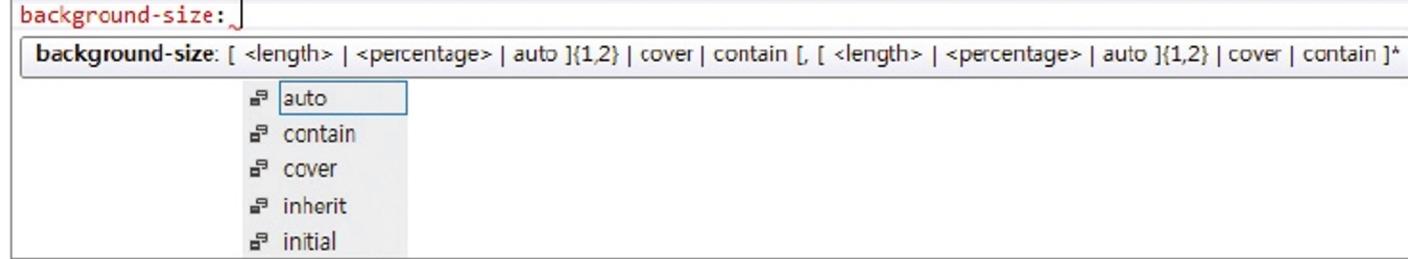
■ 배경 이미지 삽입과 크기 지정

- background-image 속성
 - 배경 이미지 지정
 - URL, 그레이디언트 입력



(a) `background-image` 속성: 배경 이미지 삽입

- `background-size` 속성
 - 배경 이미지 크기



(b) `background-size` 속성: 배경 이미지 크기

그림 6-7 배경 이미지 삽입과 이미지 크기 조정

03 배경 속성

■ [기본 예제 6-5] 배경 이미지 삽입과 크기 조정

[코드 데모](#)

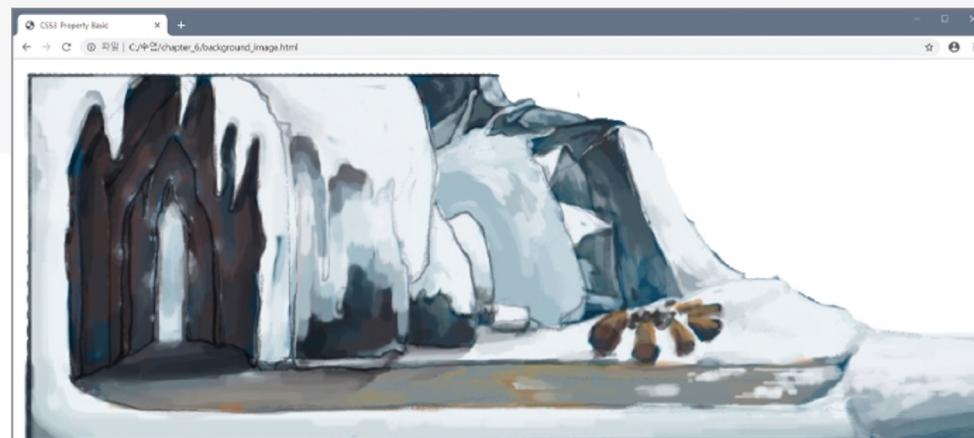
▪ 1. 배경으로 사용할 이미지 파일 준비하기

- 배경 이미지와 작성할 HTML 페이지를 같은 폴더에 넣기

▪ 2. 배경 이미지 1개 삽입하기

코드 6-14 background_image.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        body {
            background-image: url('BackgroundFront.png');
        }
    </style>
</head>
<body>
</body>
</html>
```



03 배경 속성

■ [기본 예제 6-5] 배경 이미지 삽입과 크기 조정

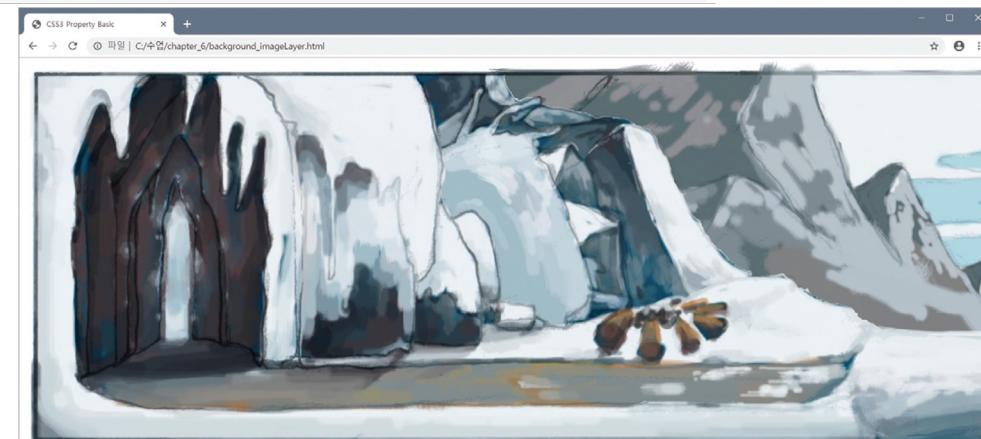
[코드 데모](#)

▪ 3. 배경 이미지 2개 삽입하기

- 먼저 입력한 이미지(왼쪽)가 앞쪽에 위치

코드 6-15 background_imageLayer.html

```
<style>
  body {
    /* 각각의 그림을 레이어라고 합니다. */
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
  }
</style>
```



03 배경 속성

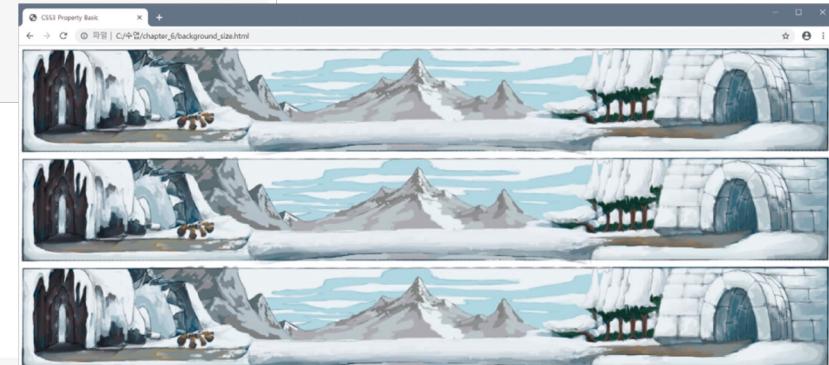
■ [기본 예제 6-5] 배경 이미지 삽입과 크기 조정

[코드 데모](#)

■ 4. 배경 이미지의 너비 조정하기

코드 6-16 background_size.html

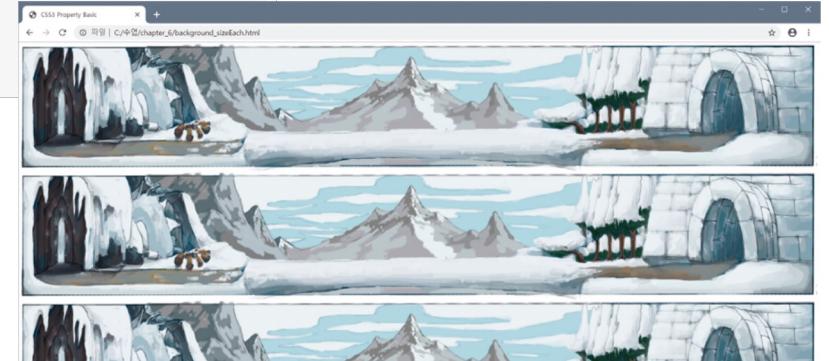
```
<style>
    body {
        background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
        background-size: 100%;
    }
</style>
```



■ 5. 배경 이미지 높이 조정하기

코드 6-17 background_sizeEach.html

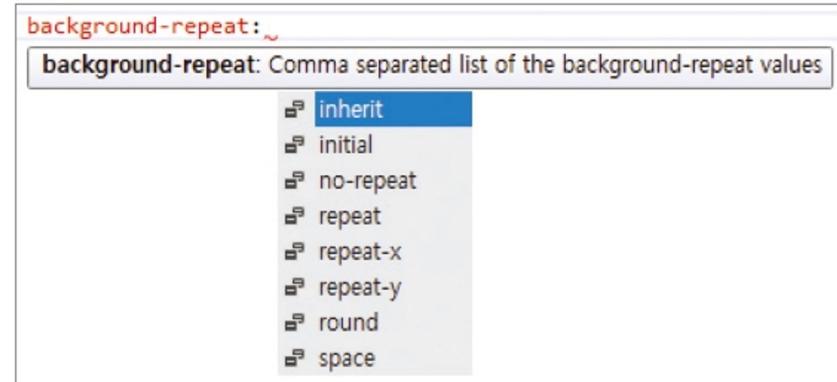
```
<style>
    body {
        background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
        background-size: 100% 250px;
    }
</style>
```



03 배경 속성

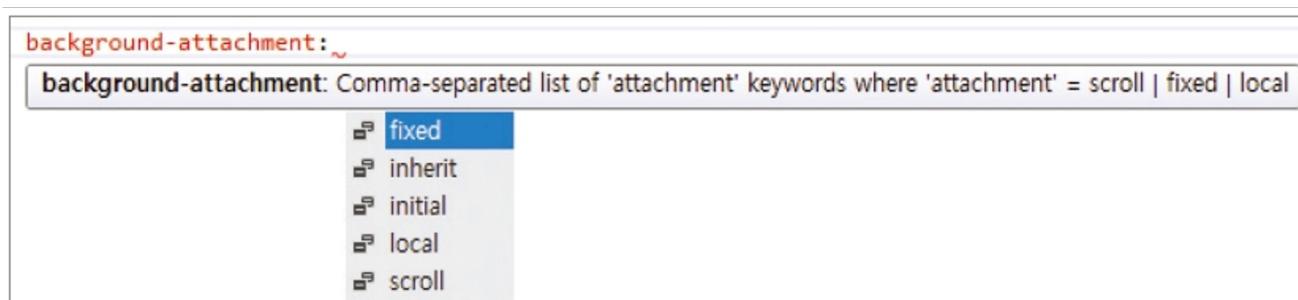
■ 배경 이미지 반복과 부착 형태, 위치

- background-repeat 속성
 - 배경 이미지의 반복 형태 지정
 - repeat-x : X축 방향으로만 이미지 반복
 - repeat-y : Y축 방향으로만 이미지 반복



(a) background-repeat 속성: 배경 이미지 반복 형태

- background-attachment 속성
 - 배경 이미지를 화면에 고정



(b) background-attachment 속성: 배경 이미지 부착 형태

그림 6-8 배경 이미지의 반복과 부착 형태 지정

03 배경 속성

■ 배경 이미지 반복과 부착 형태, 위치

▪ background-position 속성

- 배경 이미지의 배치 위치 지정
- background-position: 키워드;
- background-position: X축 위치;
- background-position: X축 위치 Y축 위치;

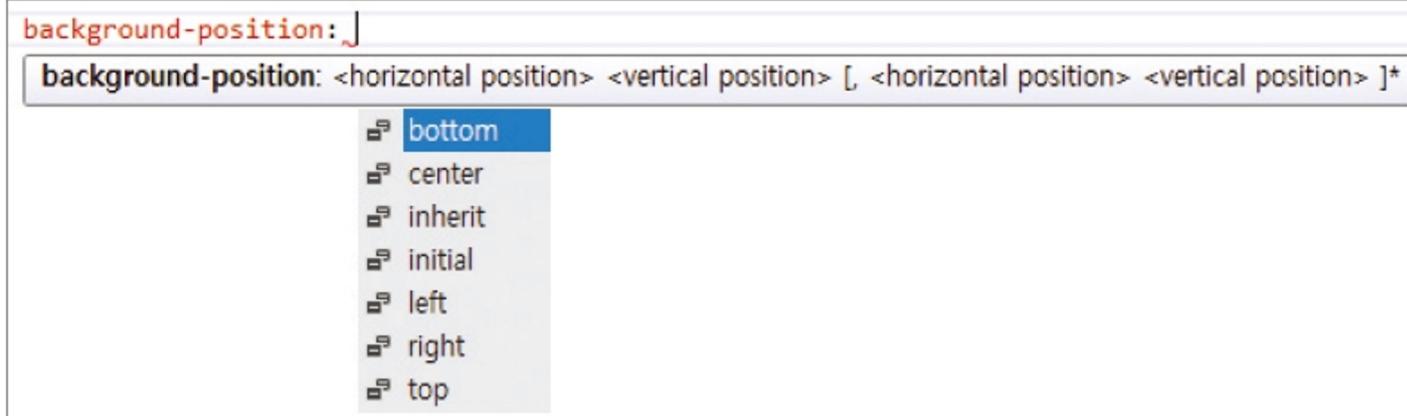


그림 6-9 background-position 속성: 배경 이미지 위치 지정

03 배경 속성

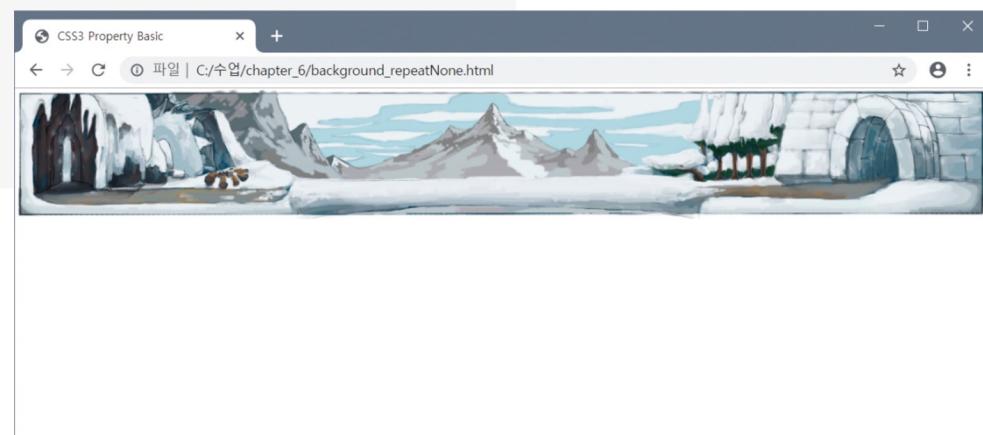
■ [기본 예제 6-6] 반복과 부착 형태, 위치

[코드 데모](#)

▪ 1. 배경 이미지 반복 없애기

코드 6-19 background_repeatNone.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        body {
            background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
            background-size: 100%;
            background-repeat: no-repeat;
        }
    </style>
</head>
<body>
</body>
</html>
```



03 배경 속성

- [기본 예제 6-6] 반복과 부착 형태, 위치
 - 2. 스크롤이 가능한 분량의 텍스트 추가하기(1)

[코드 데모](#)

코드 6-20 background_attachmentScroll.html

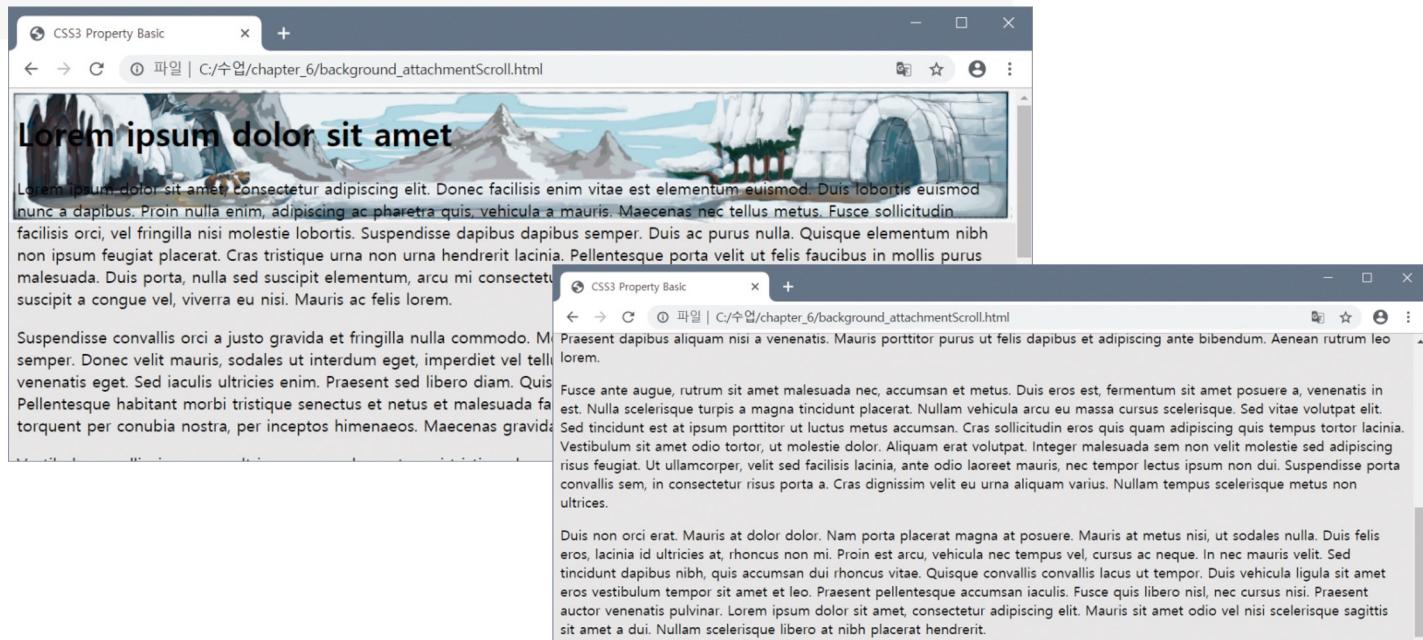
```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        body {
            background-color: #E7E7E8;
            background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
            background-size: 100%;
            background-repeat: no-repeat;
            background-attachment: scroll;
        }
    </style>
</head>
```

- [기본 예제 6-6] 반복과 부착 형태, 위치

코드 데모

- 2. 스크롤이 가능한 분량의 텍스트 추가하기(2)

```
<body>
  <h1>Lorem ipsum dolor sit amet</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p>Pellentesque est velit, laoreet vel rhoncus sit amet.</p>
  <p>Proin vitae elit est, ut accumsan arcu. Sed consectetur.</p>
  <p>Donec molestie massa id lorem hendrerit eu bibendum augue.</p>
  <p>Morbi ut lorem in purus facilisis vulputate. Sed purus nibh.</p>
</body>
</html>
```



03 배경 속성

■ [기본 예제 6-6] 반복과 부착 형태, 위치

[코드 데모](#)

▪ 3. 배경 이미지 화면에 고정시키기

코드 6-21 background_attachmentFixed.html

```
<style>
  body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
  }
</style>
```



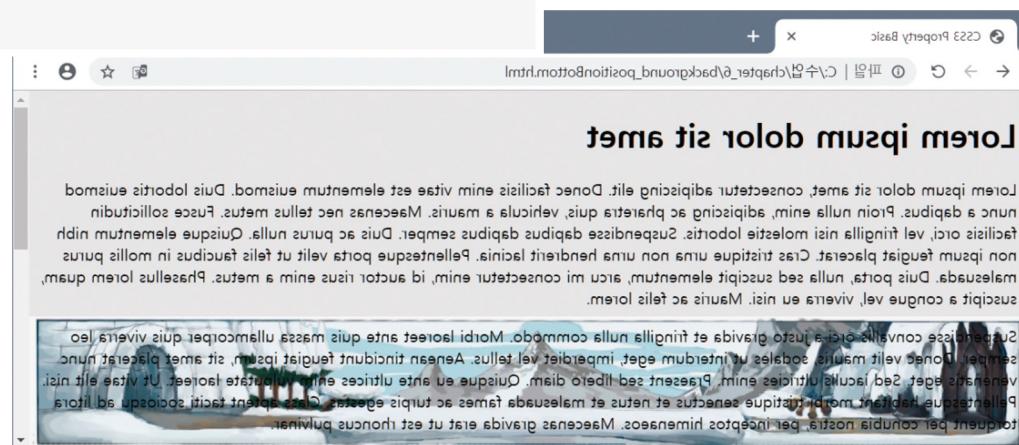
03 배경 속성

- [기본 예제 6-6] 반복과 부착 형태, 위치
- 4. 배경 이미지 아래쪽에 배치하기

[코드 데모](#)

코드 6-22 background_positionBottom.html

```
<style>
  body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: bottom;
  }
</style>
```



03 배경 속성

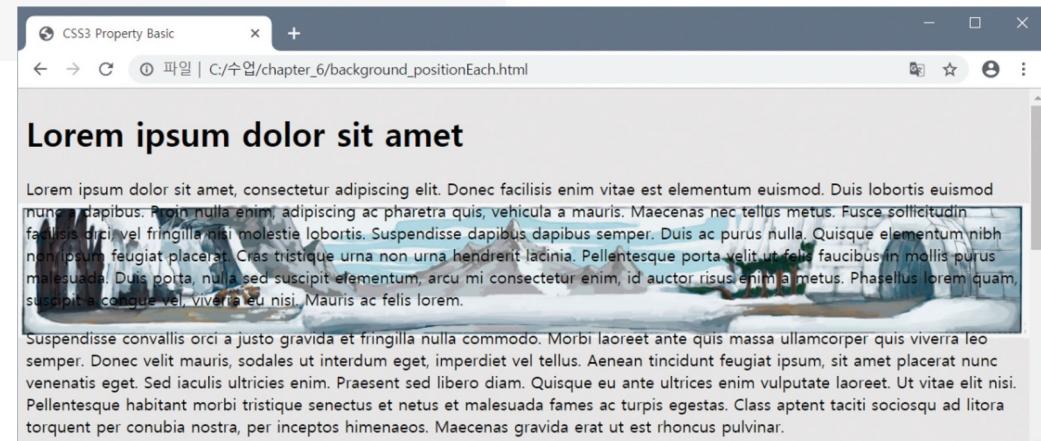
■ [기본 예제 6-6] 반복과 부착 형태, 위치

[코드 데모](#)

▪ 5. 배경 이미지 위치 조정하기

코드 6-23 background_positionEach.html

```
<style>
body {
    background-color: #E7E7E8;
    background-image: url('BackgroundFront.png'), url('BackgroundBack.png');
    background-size: 100%;
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: 0px 50%; background-position: 0px 50%;
}
</style>
```



03 배경 속성

■ 배경 속성 정리

표 6-3 이 절에서 다룬 배경 속성을 입력하는 방법

방법	코드
①	<pre>background-image: url('BackgroundBack.png'); background-position: bottom; background-size: 100%; background-repeat: no-repeat; background-attachment: fixed; background-color: #E7E7E8;</pre>
②	<pre>background: url('BackgroundFront.png') bottom / 100% no-repeat fixed, url('BackgroundBack.png') bottom / 100% no-repeat fixed #E7E7E8;</pre>

04 글자 속성

■ 글자 크기와 글꼴

font-size:

- inherit
- initial
- large
- larger
- medium
- small
- smaller
- unset
- x-large
- x-small
- xx-large
- xx-small

font-family:

- 'Courier New', Courier, monosp...
- 'Franklin Gothic Medium', 'Aria...
- 'Gill Sans', 'Gill Sans MT', Ca...
- 'Lucida Sans', 'Lucida Sans Reg...
- 'Segoe UI', Tahome, Geneva, Ver...
- 'Times New Roman', Times, serif
- 'Trebuchet MS', 'Lucida Sans Un...
- Arial, Helvetica, sans-serif
- Cambria, Cochin, Georgia, Times...
- Georgia, 'Times New Roman', Tim...
- Impact, Haettenschweiler, 'Aria...
- Verdana, Geneva, Tahome, sans-s...

(a) font-size 속성: 글자 크기

(b) font-family 속성: 글꼴

그림 6-10 글자 크기와 글꼴 지정

04 글자 속성

■ 글자 크기와 글꼴

- generic-family - 다국적 사용자 대상용 글꼴

AaBbCc AaBbCc

(a) Serif 글꼴(명조체)

(b) Sans-serif 글꼴(고딕체)

그림 6-11 generic-family 글꼴



그림 6-12 크롬의 글꼴 설정([설정]→[고급 설정 표시]→[글꼴 맞춤 설정] 메뉴)

04 글자 속성

■ [기본 예제 6-7] 글자 크기와 글꼴 지정

[코드 데모](#)

▪ 1. 글자 크기 조정하기

코드 6-24 font_size.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .a { font-size: 32px; }
        .b { font-size: 2em; }
        .c { font-size: large; }
        .d { font-size: small; }
    </style>
</head>
<body>
    <h1>Lorem ipsum</h1>
    <p class="a">Lorem ipsum</p>
    <p class="b">Lorem ipsum</p>
    <p class="c">Lorem ipsum</p>
    <p class="d">Lorem ipsum</p>
</body>
</html>
```

Lorem ipsum

Lorem ipsum

Lorem ipsum

Lorem ipsum

Lorem ipsum

04 글자 속성

- [기본 예제 6-7] 글자 크기와 글꼴 지정
 - 2. 글꼴 1개 지정하기

[코드 데모](#)

코드 6-25 font_family.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .font_arial { font-family: Arial; }
        .font_roman { font-family: 'Times New Roman'; }
    </style>
</head>
<body>
    <h1 class="font_arial">Lorem ipsum</h1>
    <p class="font_roman">Lorem ipsum</p>
</body>
</html>
```

Lorem ipsum

Lorem ipsum

04 글자 속성

■ [기본 예제 6-7] 글자 크기와 글꼴 지정

[코드 데모](#)

▪ 3. 글꼴 여러 개 지정하기

코드 6-26 font_familyEach.html

```
<style>
    .font_arial { font-family: '없는 글꼴', Arial; }
    .font_roman { font-family: 'Times New Roman', Arial; }
</style>
```

Lorem ipsum

Lorem ipsum

▪ 4. 웹 브라우저 고유 글꼴 적용하기

코드 6-27 font_familyGenericFont.html

```
<style>
    .font_arial { font-family: '없는 글꼴', sans-serif; }
    .font_roman { font-family: '없는 글꼴', serif; }
</style>
```

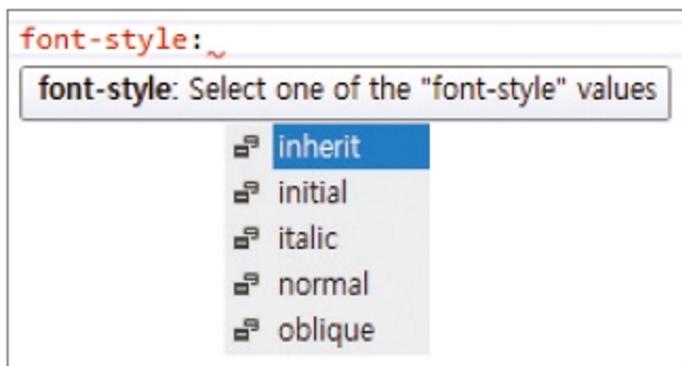
인터넷 익스플로러는 Sans-serif
글꼴과 Serif 글꼴에 따옴표를 적용
하면 안 됩니다.

Lorem ipsum

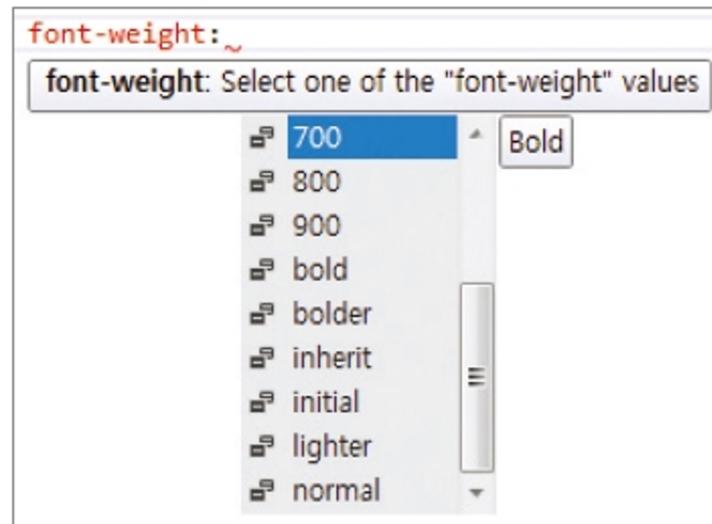
Lorem ipsum

04 글자 속성

■ 글자의 스타일과 두께



(a) font-style 속성: 글자 스타일



(b) font-weight 속성: 글자 두께

그림 6-13 글자 스타일과 두께 지정

■ [기본 예제 6-8] 글자의 스타일과 두께 조정

[코드 데모](#)

- 두께를 지원하지 않는 글꼴은 font-weight 속성을 사용해 두께 조정 불가능

코드 6-28 font_styleWeight.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .font_big { font-size: 2em; }
        .font_italic { font-style: italic; }
        .font_bold { font-weight: bold; }
    </style>
</head>
<body>
    <p class="font_big font_italic font_bold">Lorem ipsum dolor amet</p>
</body>
</html>
```

Lorem ipsum dolor amet

04 글자 속성

■ 글자 정렬

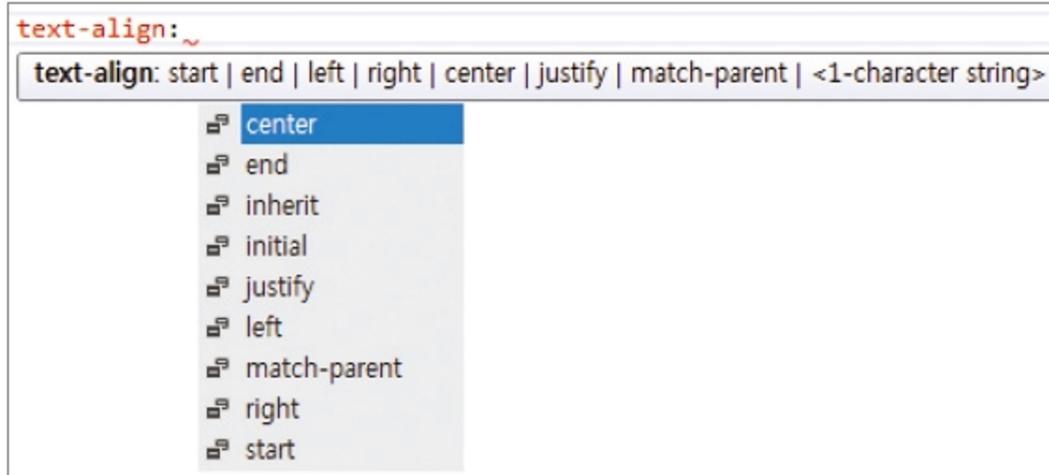


그림 6-14 text-align 속성: 글자 정렬

■ [기본 예제 6-9] 글자 정렬 지정

[코드 데모](#)

코드 6-29 font_textAlign.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .font_big { font-size: 2em; }
        .font_italic { font-style: italic; }
        .font_bold { font-weight: bold; }
        .font_center { text-align: center; }
        .font_right { text-align: right; }
    </style>
</head>
<body>
    <p class="font_big font_italic font_bold font_center">Lorem ipsum dolor amet</p>
    <p class="font_bold font_right">2019.07.10</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

Lorem ipsum dolor amet

2019.07.10

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam malesuada felis id massa adipiscing eget pellentesque ipsum lacinia. In orci massa, venenatis quis ultrices ac, accumsan vitae nunc. Sed eget pellentesque enim. Maecenas eros risus, hendrerit eget lacinia nec, mollis nec velit. Mauris sem mi, molestie sed bibendum eu, blandit et ante. Pellentesque vel mauris et mauris ornare pharetra. Etiam scelerisque pulvinar diam posuere ultrices. Duis et enim at velit ultricies cursus a at felis. Phasellus sit amet est hendrerit lorem convallis consequat ac ut massa. Aliquam luctus porttitor lacus id hendrerit. Nulla facilisi. Aliquam dictum tempus augue, eu tempor dui varius et. In pretium ultricies dapibus.

04 글자 속성

■ [기본 예제 6-10] 글자를 수직 중앙 정렬

[코드 데모](#)

- 글자 높이 지정하는 line-height 속성을 사용한 글자 수직 중앙 정렬
- 1. 간단한 버튼 만들기(1)

코드 6-31 font_lineHeightBefore.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        .font_big { font-size: 2em; }
        .font_italic { font-style: italic; }
        .font_bold { font-weight: bold; }
        .font_center { text-align: center; }

        .button {
            width: 150px;
            height: 70px;
            background-color: #FF6A00;
            border: 10px solid #FFFFFF;
            border-radius: 30px;
            box-shadow: 5px 5px 5px #A9A9A9;
        }
    </style>
</head>
<body>
    <h1>Hello World</h1>
    <p>This is a sample text.</p>
    <button>Click Me!</button>
</body>
</html>
```

04 글자 속성

■ [기본 예제 6-10] 글자를 수직 중앙 정렬

[코드 데모](#)

- 글자 높이 지정하는 line-height 속성을 사용한 글자 수직 중앙 정렬
- 1. 간단한 버튼 만들기(2)

```
.button > a {  
    display: block;  
}  
</style>  
</head>  
<body>  
    <div class="button">  
        <a href="#" class="font_big font_italic font_bold font_center">Click</a>  
    </div>  
</body>  
</html>
```



04 글자 속성

■ [기본 예제 6-10] 글자를 수직 중앙 정렬

[코드 데모](#)

▪ 2. 글자를 수직 중앙 정렬하기

코드 6-32 font_lineHeightAfter.html

```
.button {  
    width: 150px;  
    height: 70px;  
    background-color: #FF6A00;  
    border: 10px solid #FFFFFF;  
    border-radius: 30px;  
    box-shadow: 5px 5px 5px #A9A9A9;  
}  
  
 .button > a {  
    display: block;  
    line-height: 70px;  
}
```

글자를 감싸는 박스 높이와 같은 크기의
line-height 속성을 적용했습니다.



Click

04 글자 속성

- 링크 글자의 밑줄

- text-decoration - <a> 태그에 href 속성 지정하면 적용되는 밑줄 제거

Lorem ipsum dolor amet

그림 6-15 href 속성이 적용된 a 태그

- [기본 예제 6-11] 링크의 밑줄 제거

[코드 데모](#)

코드 6-33 font_textDecoration.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Font Property</title>
    <style>
        a { text-decoration: none; }
    </style>
</head>
<body>
    <h1>
        <a href="#">Lorem ipsum dolor amet</a>
    </h1>
</body>
</html>
```

Lorem ipsum dolor amet

05 위치 속성

■ 요소의 고정 위치와 상대 위치 지정

- position – 요소의 위치 지정 형식 설정
- 절대 위치 좌표 : absolute, fixed
- 상대 위치 좌표 : relative, static

position:	
position: Example:	div { position: absolute; }
absolute	절대 위치 좌표 설정
fixed	화면을 기준으로 절대 위치 좌표 설정
relative	초기 위치에서 상하좌우로 위치 이동
static	위쪽에서 아래쪽으로 순서대로 배치

그림 6-16 position 속성과 키워드

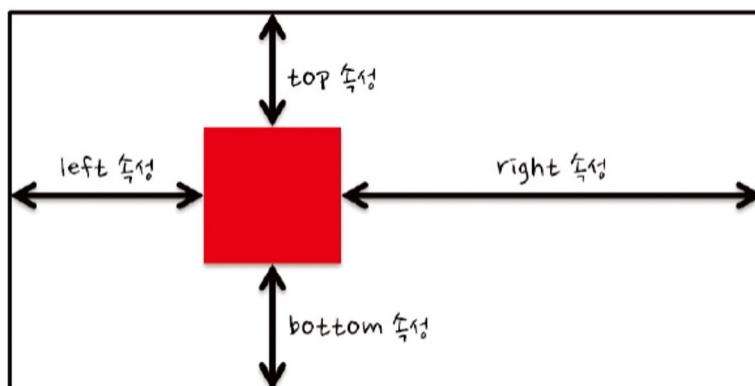


그림 6-17 position 속성과 함께 사용하는 스타일 속성

05 위치 속성

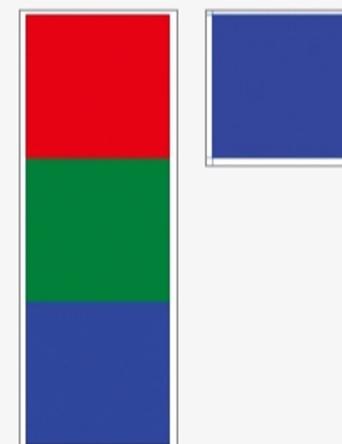
■ [기본 예제 6-12] position 속성

[코드 데모](#)

▪ 1. absolute 키워드 적용하기

코드 6-34 position_absolute.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        .box {
            width: 100px; height: 100px;
            position: absolute;
        }
        .box:nth-child(1) { background-color: red; }
        .box:nth-child(2) { background-color: green; }
        .box:nth-child(3) { background-color: blue; }
    </style>
</head>
<body>
    <div class="box"></div>
    <div class="box"></div>
    <div class="box"></div>
</body>
</html>
```



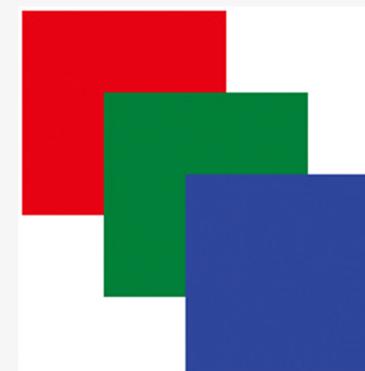
05 위치 속성

- [기본 예제 6-12] position 속성
 - 2. left 속성과 top 속성 적용하기

[코드 데모](#)

코드 6-35 position_absoluteLeftTop.html

```
<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: 90px; top: 90px;
  }
</style>
```



05 위치 속성

■ [기본 예제 6-12] position 속성

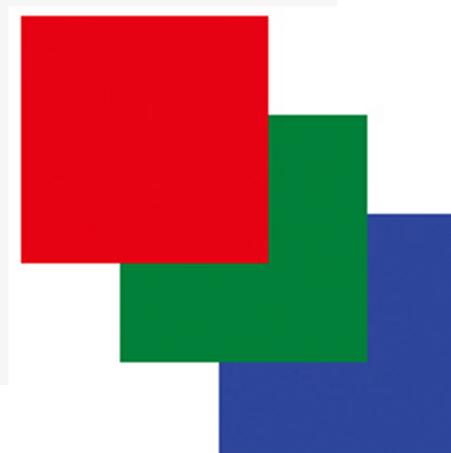
[코드 데모](#)

▪ 3. z-index 속성 적용하기

- 숫자를 입력하면 숫자가 클수록 앞에 위치

코드 6-36 position_zIndex.html

```
<style>
    .box {
        width: 100px; height: 100px;
        position: absolute;
    }
    .box:nth-child(1) {
        background-color: red;
        left: 10px; top: 10px;
        z-index: 100;
    }
    .box:nth-child(2) {
        background-color: green;
        left: 50px; top: 50px;
        z-index: 10;
    }
    .box:nth-child(3) {
        background-color: blue;
        left: 90px; top: 90px;
        z-index: 1;
    }
</style>
```



05 위치 속성

- 위치 속성 공식
 - 위치 속성 공식을 적용하지 않은 코드(1)

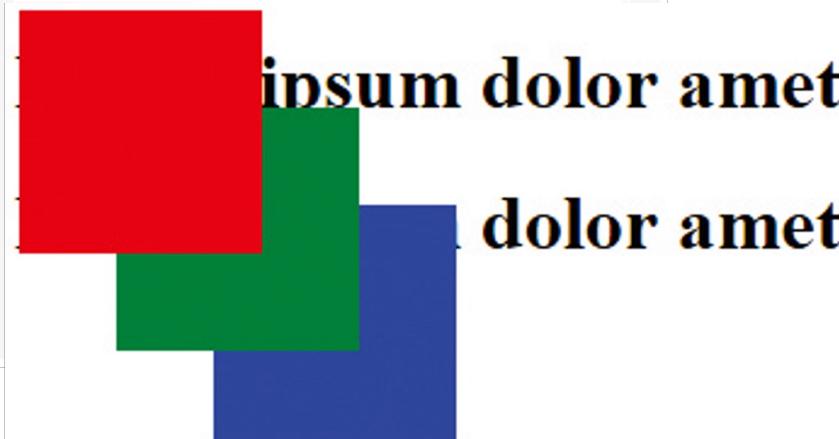
코드 6-37 position_formulaBasic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        .box {
            width: 100px; height: 100px;
            position: absolute;
        }
        .box:nth-child(1) {
            background-color: red;
            left: 10px; top: 10px;
            z-index: 100;
        }
        .box:nth-child(2) {
            background-color: green;
            left: 50px; top: 50px;
            z-index: 10;
        }
    </style>
</head>
<body>
</body>
</html>
```

05 위치 속성

- 위치 속성 공식
- 위치 속성 공식을 적용하지 않은 코드(2)

```
.box:nth-child(3) {  
    background-color: blue;  
    left: 90px; top: 90px;  
    z-index: 1;  
}  
</style>  
</head>  
<body>  
    <h1>Lorem ipsum dolor amet</h1>  
    <div>  
        <div class="box"></div>  
        <div class="box"></div>  
        <div class="box"></div>  
    </div>  
    <h1>Lorem ipsum dolor amet</h1>  
</body>  
</html>
```



05 위치 속성

■ 위치 속성 공식

▪ 문제 파악하기

- ❶ h1 태그 2개가 붙어 있음(div 태그가 영역을 차지하지 않음)
- ❷ 색상이 적용된 상자가 자신의 부모를 기준으로 위치를 잡지 않음

▪ ❶ 문제 해결하기

- “자손의 position 속성에 absolute 키워드를 적용하려면 부모에 height 속성을 입력합니다.”

코드 6-38 position_formulaHeight.html

```
body > div {  
    width: 400px; height: 100px;  
    border: 3px solid black;  
}
```

05 위치 속성

■ 위치 속성 공식

▪ ② 번 문제 해결하기

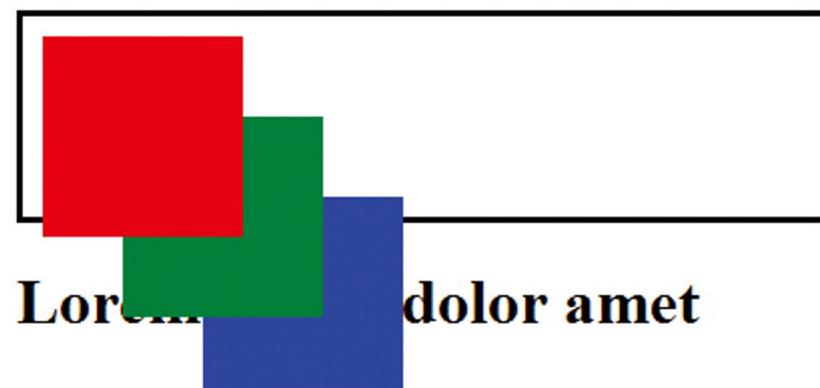
- “자손의 position 속성에 absolute 키워드를 적용하려면 부모의 position 속성에 relative 키워드를 적용합니다.”

코드 6-39 position_formulaRelative.html

```
body > div {  
    width: 400px; height: 100px;  
    border: 3px solid black;  
    position: relative;  
}
```

▪ 위치 속성 공식 적용 후 실행 결과

Lorem ipsum dolor amet



05 위치 속성

■ 내용이 요소 크기를 벗어날 때 처리

▪ overflow

- 내용이 요소 크기를 벗어나 모두 보여 주기 힘들 때 어떻게 보여 줄지 지정
- float 속성과 함께 자주 사용

표 6-4 overflow 속성 키워드

키워드	설명
hidden	영역을 벗어나는 부분 감춤
scroll	영역을 벗어나는 부분을 스크롤로 만듦

05 위치 속성

■ [기본 예제 6-13] overflow 속성

[코드 데모](#)

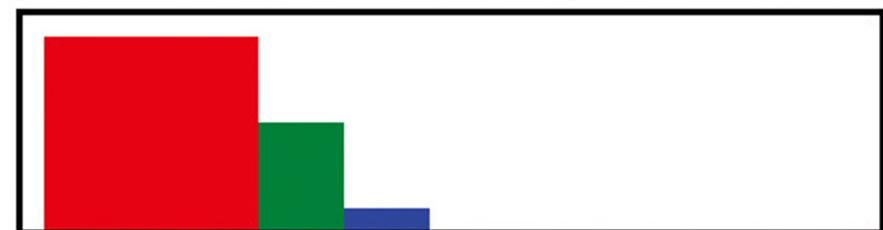
▪ 1. hidden 키워드 적용하기

- 부모 영역을 벗어난 부분을 잘라 모두 감춤

코드 6-40 position_formulaOverflowHidden.html

```
body > div {  
    width: 400px; height: 100px;  
    border: 3px solid black;  
    position: relative;  
    overflow: hidden;  
}
```

Lorem ipsum dolor amet



Lorem ipsum dolor amet

■ [기본 예제 6-13] overflow 속성

[코드 데모](#)

▪ 2. scroll 키워드 적용하기

- 부모 영역을 벗어난 부분은 스크롤을 내리면 볼 수 있음

코드 6-41 position_formulaOverflowScroll.html

```
body > div {  
    width: 400px; height: 100px;  
    border: 3px solid black;  
    position: relative;  
    overflow: scroll;  
}
```

Lorem ipsum dolor amet



Lorem ipsum dolor amet

05 위치 속성

■ [기본 예제 6-13] overflow 속성

[코드 데모](#)

▪ 3. overflow-x와 overflow-y 속성 적용하기

- 특정한 방향으로만 스크롤 생성할 때

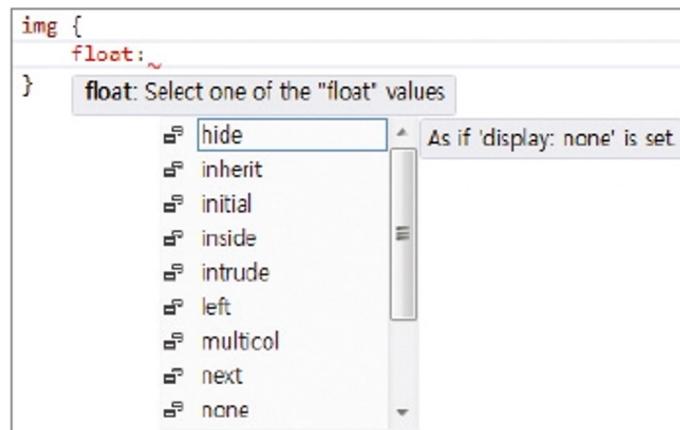
코드 6-42 position_formulaOverflowEach.html

```
body > div {  
    width: 400px; height: 100px;  
    border: 3px solid black;  
    position: relative;  
    overflow-y: scroll;  
}
```

06 유동 속성

■ float 속성

- 웹 페이지의 레이아웃을 잡을 때 많이 사용
- 웹 브라우저 크기에 상관없이 공지 등을 일정한 위치에 고정할 때 적합



The screenshot shows a CSS editor interface. A dropdown menu is open for the 'float' property of an 'img' selector. The menu lists several values: 'hide', 'inherit', 'initial', 'inside', 'intrude', 'left', 'multicol', 'next', and 'none'. The 'left' option is highlighted with a blue selection bar. A tooltip above the menu says 'As if 'display: none' is set'. To the right of the editor, there is a table with two rows, each containing a keyword and its description.

키워드	설명
left	태그를 왼쪽에 붙임
right	태그를 오른쪽에 붙임

그림 6-19 float 속성과 주요 키워드

■ [기본 예제 6-14] float 기본 속성

[코드 데모](#)

■ 1. HTML 페이지 만들기

코드 6-43 float_basic.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Float Style Property</title>
  <style>
    </style>
</head>
<body>
  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <p>In hac habitasse platea dictumst. Donec lobortis augue a metus.</p>
</body>
</html>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam sagittis nulla at dolor ullamcorper mollis sodales risus tincidunt. Vestibulum laoreet facilisis libero at tincidunt. In et tellus dolor. Nulla lectus ligula, venenatis dapibus imperdiet in, fringilla sit amet risus. Ut malesuada convallis turpis, nec molestie sem lobortis vitae. Donec porta, quam at ornare elementum, nisi sapien tempor orci, eu bibendum sapien risus ac eros. Curabitur egestas mi vitae purus eleifend dignissim. Fusce et diam justo, commodo tempor ante. In in nibh erat, eget mollis lorem. Curabitur at neque elit, in tristique lacus. Nulla et velit odio. Quisque non tellus vitae magna consectetur molestie a eu dui.

■ [기본 예제 6-14] float 기본 속성

[코드 데모](#)

▪ 2. float 속성 적용하기

- 이미지와 글이 어우러진 형태로 출력

코드 6-44 float_image.html

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



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam sagittis nulla at dolor ullamcorper mollis sodales risus tincidunt. Vestibulum laoreet facilisis libero at tincidunt. In et tellus dolor. Nulla lectus ligula, venenatis dapibus imperdiet in, fringilla sit amet risus. Ut malesuada convallis turpis, nec molestie sem lobortis vitae. Donec porta, quam at ornare elementum, nisi sapien tempor orci, eu bibendum sapien risus ac eros. Curabitur egestas mi vitae purus eleifend dignissim. Fusce et diam justo, commodo tempor ante. In in nibh erat, eget mollis lorem. Curabitur at neque elit, in tristique lacus. Nulla et velit odio. Quisque non tellus vitae magna consectetur molestie a eu dui.

Curabitur urna eros, sodales nec pretium eget, mollis non nisi. Nullam bibendum, enim vel porttitor rhoncus, nisl enim tristique odio, eget euismod velit turpis sed nisi. Curabitur vitae sem et nulla pellentesque ultricies quis a mi. Phasellus rhoncus tincidunt suscipit. Fusce ac vulputate massa. Nulla congue tincidunt libero, ac commodo orci dapibus id. Quisque in ullamcorper tortor. Nullam ut fringilla lacus.

- [기본 예제 6-15] 수평 정렬

[코드 데모](#)

- 1. float 속성을 적용할 대상 만들기

코드 6-45 float_boxBasic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Float Style Property</title>
    <style>

        </style>
    </head>
    <body>
        <div class="box">1</div>
        <div class="box">2</div>
    </body>
</html>
```

06 유동 속성

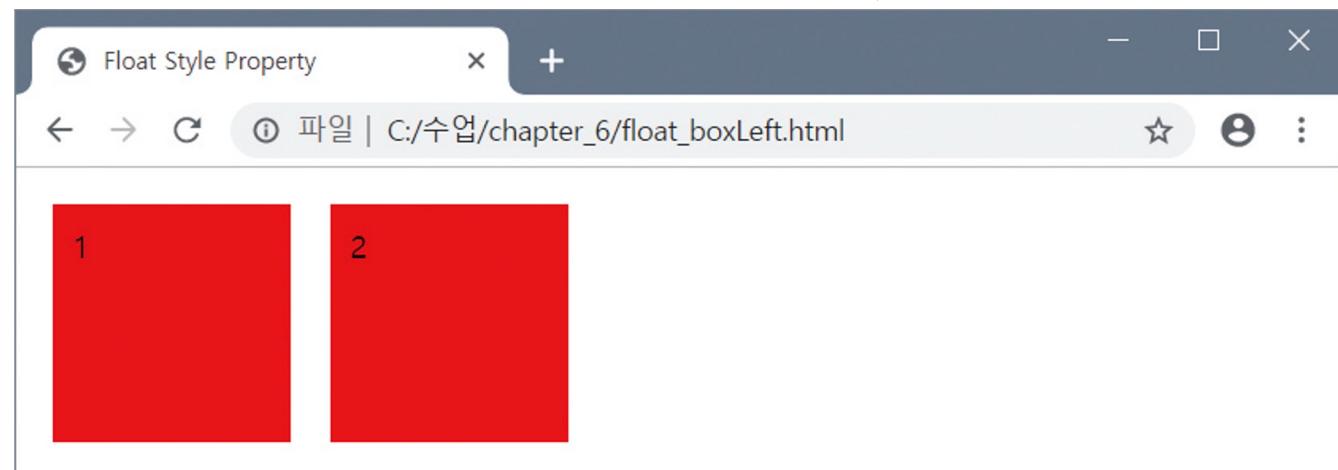
■ [기본 예제 6-15] 수평 정렬

[코드 데모](#)

▪ 2. left 키워드 적용하기

코드 6-46 float_boxLeft.html

```
<style>
  .box {
    width: 100px; height: 100px;
    background-color: red;
    margin: 10px; padding: 10px;
    /* 태그를 왼쪽으로 붙입니다. */
    float: left;
  }
</style>
```



06 유동 속성

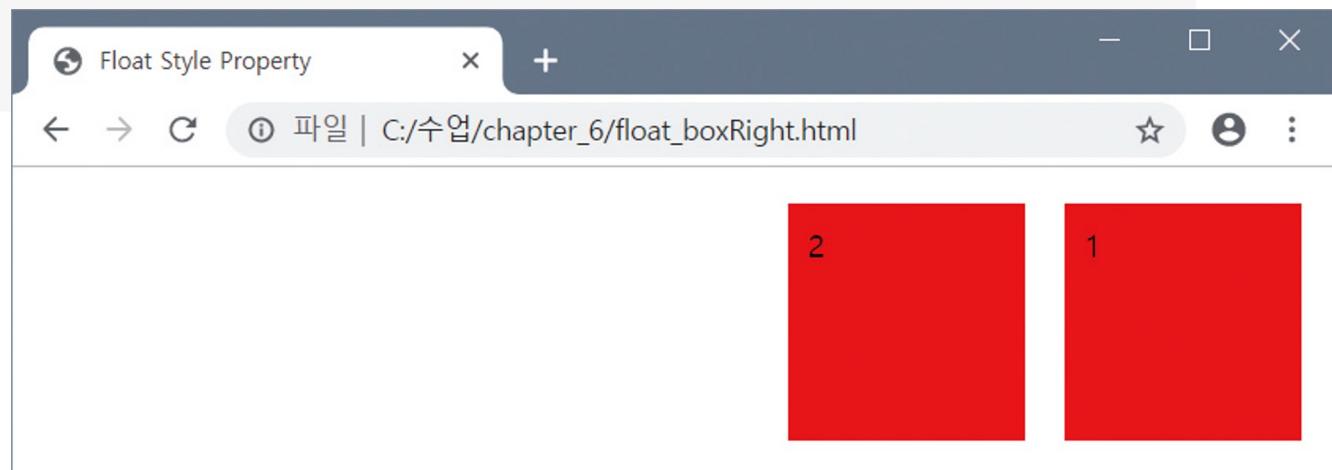
■ [기본 예제 6-15] 수평 정렬

[코드 데모](#)

▪ 3. float 속성이 right 키워드일 때 확인하기

코드 6-47 float_boxRight.html

```
<style>
  .box {
    width: 100px; height: 100px;
    background-color: red;
    margin: 10px; padding: 10px;
    /* 태그를 오른쪽으로 붙입니다. */
    float: right;
  }
</style>
```



07 그림자와 그레이디언트 속성

- 그림자 속성

`text-shadow: 5px 5px 5px black`

오른쪽 아래 흐림도 색상

(a) `text-shadow` 속성: 글자에 그림자 부여

`box-shadow: 5px 5px 5px black`

오른쪽 아래 흐림도 색상

(b) `box-shadow` 속성: 박스에 그림자 부여

그림 6-20 그림자 속성 지정

07 그림자와 그레이디언트 속성

- [기본 예제 6-16] 글자와 박스에 그림자 생성

[코드 데모](#)

- 1. 글자에 그림자 주기

코드 6-48 shadow_textShadow.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        h1 {
            text-shadow: 5px 5px 5px black;
        }
    </style>
</head>
<body>
    <h1>Lorem ipsum dolor amet</h1>
</body>
</html>
```



The image shows a large, bold h1 heading with the text "Lorem ipsum dolor amet". The text is rendered with a black text shadow, which creates a blurred, darker version of the text just below and to the right of the original text. This visual effect demonstrates the use of the CSS3 text-shadow property.

07 그림자와 그레이디언트 속성

- [기본 예제 6-16] 글자와 박스에 그림자 생성
 - 2. 박스에 그림자 만들기

[코드 데모](#)

코드 6-49 shadow_boxShadow.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Property Basic</title>
    <style>
        div {
            border: 3px solid black;
            box-shadow: 10px 10px 30px black;
            text-shadow: 5px 5px 5px black;
        }
    </style>
</head>
<body>
    <div class="box">
        <h1>Lorem ipsum dolor amet</h1>
    </div>
</body>
</html>
```



Lorem ipsum dolor amet

07 그림자와 그레이디언트 속성

- [기본 예제 6-16] 글자와 박스에 그림자 생성
 - 3. 중첩 그림자 만들기

[코드 데모](#)

코드 6-50 shadow_duplication.html

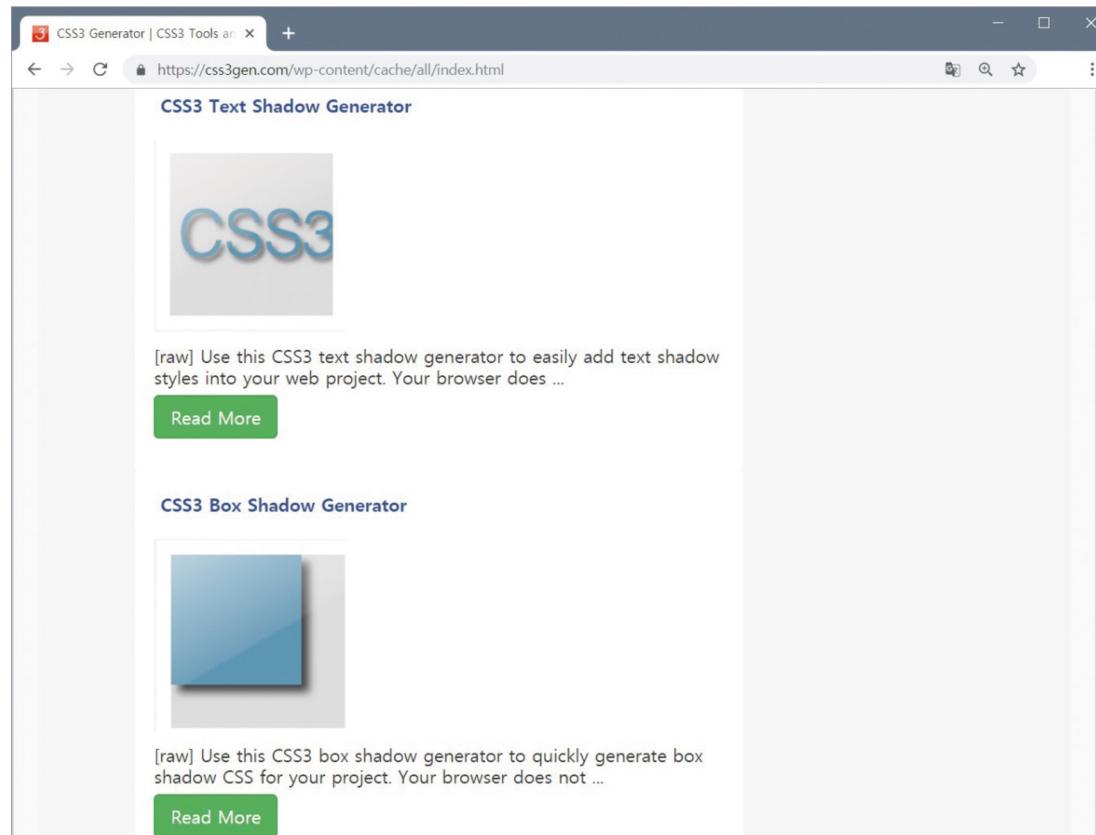
```
<style>
  .box {
    border: 3px solid black;
    box-shadow: 10px 10px 10px black, 10px 10px 20px orange, 10px 10px 30px red;
    text-shadow: 10px 10px 10px black, 10px 10px 20px orange, 10px 10px 30px red;
  }
</style>
```



Lorem ipsum dolor amet

07 그림자와 그레이디언트 속성

- [기본 예제 6-17] CSS3 Generator를 사용해 그림자 생성
 - CSS3 Generator 웹 사이트(<http://css3generator.com/>)에 접속
 - box-shadow와 text-shadow 속성을 쉽게 생성 가능



07 그림자와 그레이디언트 속성

■ 그레이디언트 속성

- 두 가지 이상의 색상을 혼합하는 채색 기능



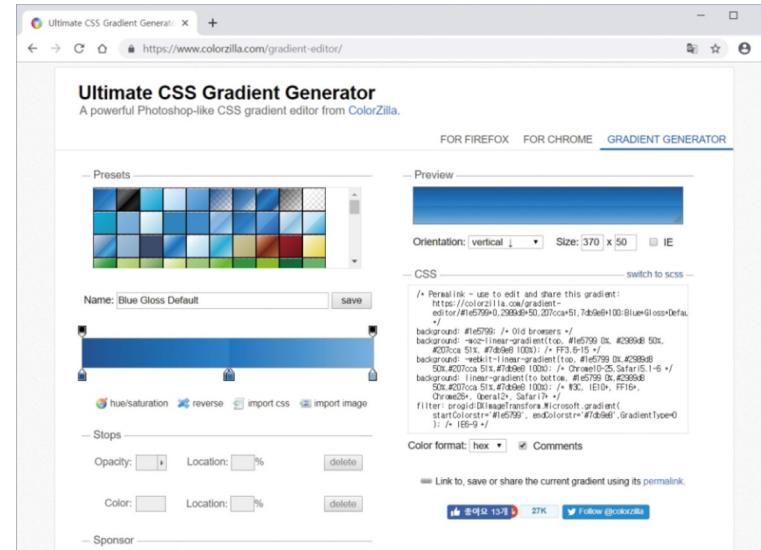
그림 6-21 메인 메뉴 예

■ [기본 예제 6-18] 그레이디언트 생성

코드 데모

▪ 1. 그레이디언트 생성기 사용하기

- <http://www.colorzilla.com/gradient-editor/>에 접속
 - Ultimate CSS Gradient Generator 페이지 왼쪽에서 원하는 형태의 그레이디언트 생성
 - 오른쪽에 생성된 CSS 코드 복사



07 그림자와 그레이디언트 속성

- [기본 예제 6-18] 그레이디언트 생성

[코드 데모](#)

- 2. 복사한 CSS 코드를 사용해 HTML 페이지 만들기(1)

코드 6-51 gradient_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS3 Style Property Basic</title>
    <style>
        div {
            font-family: 'Times New Roman';
            height: 60px;
            line-height: 60px;
            text-align: center;

            /* 복사해서 붙여 넣으세요. */
            /* Old browsers */
            background: #f85032;
            /* FF3.6+ */
            background: -moz-linear-gradient(top, #f85032 0%, #f16f5c 50%,
                #f6290c 51%, #f02f17 71%, #e73827 100%);
            /* Chrome,Safari4+ */
            background: -webkit-gradient(linear, left top, left bottom,
                color-stop(0%,#f85032), color-stop(50%,#f16f5c), color-stop(51%,#f6290c),
                color-stop(71%,#f02f17), color-stop(100%,#e73827));
            /* Chrome10+,Safari5.1+ */
            background: -webkit-linear-gradient(top, #f85032 0%, #f16f5c 50%,
                #f6290c 51%, #f02f17 71%, #e73827 100%);

        }
    </style>
</head>
<body>
    <div>Hello World</div>
</body>
</html>
```

07 그림자와 그레이디언트 속성

- [기본 예제 6-18] 그레이디언트 생성

[코드 데모](#)

- 2. 복사한 CSS 코드를 사용해 HTML 페이지 만들기(2)

```
/* Opera 11.10+ */
background: -o-linear-gradient(top, #f85032 0%, #f16f5c 50%,
#f6290c 51%, #f02f17 71%, #e73827 100%);
/* IE10+ */
background: -ms-linear-gradient(top, #f85032 0%, #f16f5c 50%,
#f6290c 51%, #f02f17 71%, #e73827 100%);
/* W3C */
background: linear-gradient(to bottom, #f85032 0%, #f16f5c 50%,
#f6290c 51%, #f02f17 71%, #e73827 100%);
/* IE6-9 */
filter: progid:DXImageTransform.Microsoft.gradient
(startColorstr='#f85032', endColorstr='#e73827', GradientType=0);
}
</style>
</head>
<body>
<div>
    <h1>CSS3 그레이디언트</h1>
</div>
</body>
</html>
```



CSS3 그레이디언트

07 그림자와 그레이디언트 속성

■ 선형 그레이디언트 분석

- linear-gradient()
 - 선형 그레이디언트가 진행될 각도와 색상 블록을 차례로 입력

색상 블록
linear-gradient(45deg, #f85032 0%, #e73827 100%)
각도 색상 위치

그림 6-22 linear-gradient() 함수



그림 6-23 그레이디언트

- radial-gradient()
 - 원형 그레이디언트

01 수평, 중앙, One True 정렬 레이아웃

■ 수평 정렬 레이아웃

- float와 overflow 속성을 사용해 레이아웃을 만들기
- 규칙
 - 자손에게 float 속성을 지정하고, 부모의 overflow 속성에는 hidden 키워드를 적용

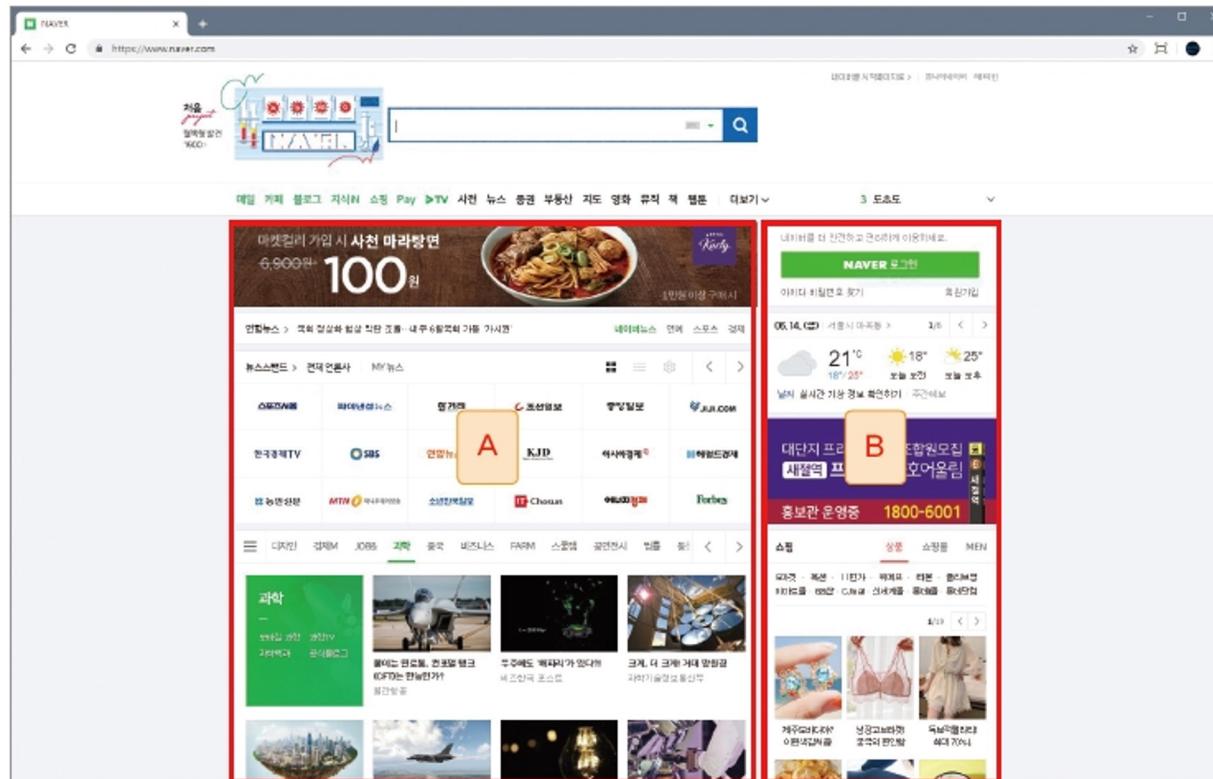


그림 7-1 수평 정렬 레이아웃 활용 예(A와 B가 수평 정렬되어 있음)

01 수평, 중앙, One True 정렬 레이아웃

■ [응용 예제 7-1] 수평 정렬 레이아웃

[코드 데모](#)[온라인 에디터](#)

코드 7-1 example_1.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Float With Overflow</title>
    <style>
        div.container {
            overflow: hidden;
        }

        div.item {
            float: left;
            margin: 0 3px;
            padding: 10px;
            border: 1px solid black;
        }
    </style>
</head>
```

```
<body>
    <div class="container">
        <div class="item">메뉴 - 1</div>
        <div class="item">메뉴 - 2</div>
        <div class="item">메뉴 - 3</div>
        <div class="item">메뉴 - 4</div>
    </div>
</body>
</html>
```

메뉴 - 1

메뉴 - 2

메뉴 - 3

메뉴 - 4

01 수평, 중앙, One True 정렬 레이아웃

- [응용 예제 7-1] 수평 정렬 레이아웃- overflow 속성 지정 이유
 - 1. [코드 7-1]에서 overflow 속성을 제거
 - 2. [그림 7-2]의 (a)와 같이 수평 정렬한 요소의 앞뒤로 글자를 넣어 주기

```
<body>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <div class="container">
    <div class="item">메뉴 - 1</div>
    <div class="item">메뉴 - 2</div>
    <div class="item">메뉴 - 3</div>
    <div class="item">메뉴 - 4</div>
  </div>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
```

01 수평, 중앙, One True 정렬 레이아웃

■ [응용 예제 7-1] 수평 정렬 레이아웃- overflow 속성 지정 이유

- (b) overflow 속성 적용하지 않음
 - 요소가 떠다님

```
<!DOCTYPE html>
<html>
<head>
    <title>Float With Overflow</title>
    <style>
        div.container {
            width: 300px;
            height: 150px;
            border: 1px solid black;
            margin: auto;
        }

        div.item {
            float: left;
            margin: 0 3px;
            padding: 10px;
            border: 1px solid black;
        }
    </style>
</head>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eu justo arcu, sit amet pellentesque tellus. Integer bibendum erat a erat placerat id pharetra dolor pretium. Nullam feugiat sodales laoreet.

메뉴 - 1 | 메뉴 - 2 | 메뉴 - 3 | 메뉴 - 4

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eu justo arcu, sit amet pellentesque tellus. Integer bibendum erat a erat placerat id pharetra dolor pretium. Nullam feugiat sodales laoreet.

- (c) 부모의 overflow 속성을 hidden으로 입력

```
<!DOCTYPE html>
<html>
<head>
    <title>Float With Overflow</title>
    <style>
        div.container {
            width: 300px;
            height: 150px;
            border: 1px solid black;
            overflow: hidden;
            margin: auto;
        }

        div.item {
            float: left;
            margin: 0 3px;
            padding: 10px;
            border: 1px solid black;
        }
    </style>
</head>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eu justo arcu, sit amet pellentesque tellus. Integer bibendum erat a erat placerat id pharetra dolor pretium. Nullam feugiat sodales laoreet.

메뉴 - 1 | 메뉴 - 2 | 메뉴 - 3 | 메뉴 - 4

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean eu justo arcu, sit amet pellentesque tellus. Integer bibendum erat a erat placerat id pharetra dolor pretium. Nullam feugiat sodales laoreet.

01 수평, 중앙, One True 정렬 레이아웃

■ 중앙 정렬 레이아웃

- 특정한 너비의 중앙에 내용이 위치



그림 7-3 중앙 정렬 레이아웃 예

01 수평, 중앙, One True 정렬 레이아웃

■ [응용 예제 7-2] 중앙 정렬 레이아웃

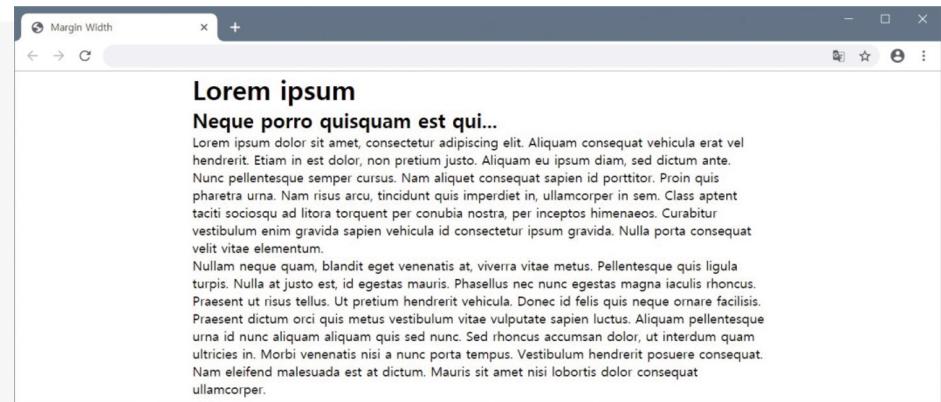
[코드 데모](#)

- width 속성을 부여하고 margin 속성을 '0 auto'로 입력

코드 7-2 example_2.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Margin Width</title>
    <style>
        /* 초기화 */
        * { margin: 0; padding: 0; }

        /* 주제 */
        body {
            margin: 0 auto;
            width: 960px;
        }
    </style>
</head>
<body>
    <h1>Lorem ipsum</h1>
    <h2>Neque porro quisquam est qui...</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p>Quisque cursus sem tortor, eu interdum elit.</p>
</body>
</html>
```



01 수평, 중앙, One True 정렬 레이아웃

■ One True 레이아웃

- 국내 모든 포털 사이트의 메인 페이지 레이아웃
- “행을 독립적으로 생각”해서 공간을 나눔

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu. Suspendisse tincidunt venenatis felis a consectetur. Etiam adipiscing gravida pretium. Nunc pulvinar elementum auctor.

*Lorem ipsum dolor sit amet,
consectetur
adipiscing elit.*

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu. Suspendisse tincidunt venenatis felis a consectetur. Etiam adipiscing gravida pretium. Nunc pulvinar elementum auctor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu.

그림 7-4 One True 레이아웃 예

01 수평, 중앙, One True 정렬 레이아웃

■ [응용 예제 7-3] One True 레이아웃(1)

[코드 데모](#)

- 1. 행 구성하기
- 2. 열 구성하기
- 3. 레이아웃 구성하기
 - ① 부모 태그에 고정된 너비를 입력
 - ② 수평 정렬하는 부모 태그의 overflow 속성에 hidden을 적용
 - ③ 자손 태그에 적당한 너비를 입력하고 float 속성을 적용

코드 7-3 example_3.html

```
<!DOCTYPE html>
<html>
<head>
    <title>One True Layout</title>
    <style>
        body {
            width: 500px;
            margin: 10px auto;
        }
        #middle { overflow: hidden; }
        #left { float: left; width: 150px; background: red; }
        #right { float: right; width: 350px; background: blue; }
    </style>
</head>
<body>
    <div id="middle">
        <div id="left"></div>
        <div id="right"></div>
    </div>
</body>
</html>
```

01 수평, 중앙, One True 정렬 레이아웃

■ [응용 예제 7-3] One True 레이아웃(2)

[코드 데모](#)

```
#top { background: green; }
#bottom { background: purple; }

</style>
</head>
<body>
  <div id="top">Lorem ipsum dolor sit amet, consectetur adipiscing.</div>
  <div id="middle">
    <div id="left">Lorem ipsum dolor sit amet, consectetur adipiscing.</div>
    <div id="right">Lorem ipsum dolor sit amet, consectetur elit.</div>
  </div>
  <div id="bottom">Lorem ipsum dolor sit amet, consectetur.</div>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu. Suspendisse tincidunt venenatis felis a consectetur. Etiam adipiscing gravida pretium. Nunc pulvinar elementum auctor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu. Suspendisse tincidunt venenatis felis a consectetur. Etiam adipiscing gravida pretium. Nunc pulvinar elementum auctor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris eget leo vel massa scelerisque tristique ultrices id arcu.

02 요소 배치

■ 절대 위치를 사용한 요소 배치

- 자손의 position 속성에 absolute 를 적용하려면
부모의 position 속성에 relative를 적용

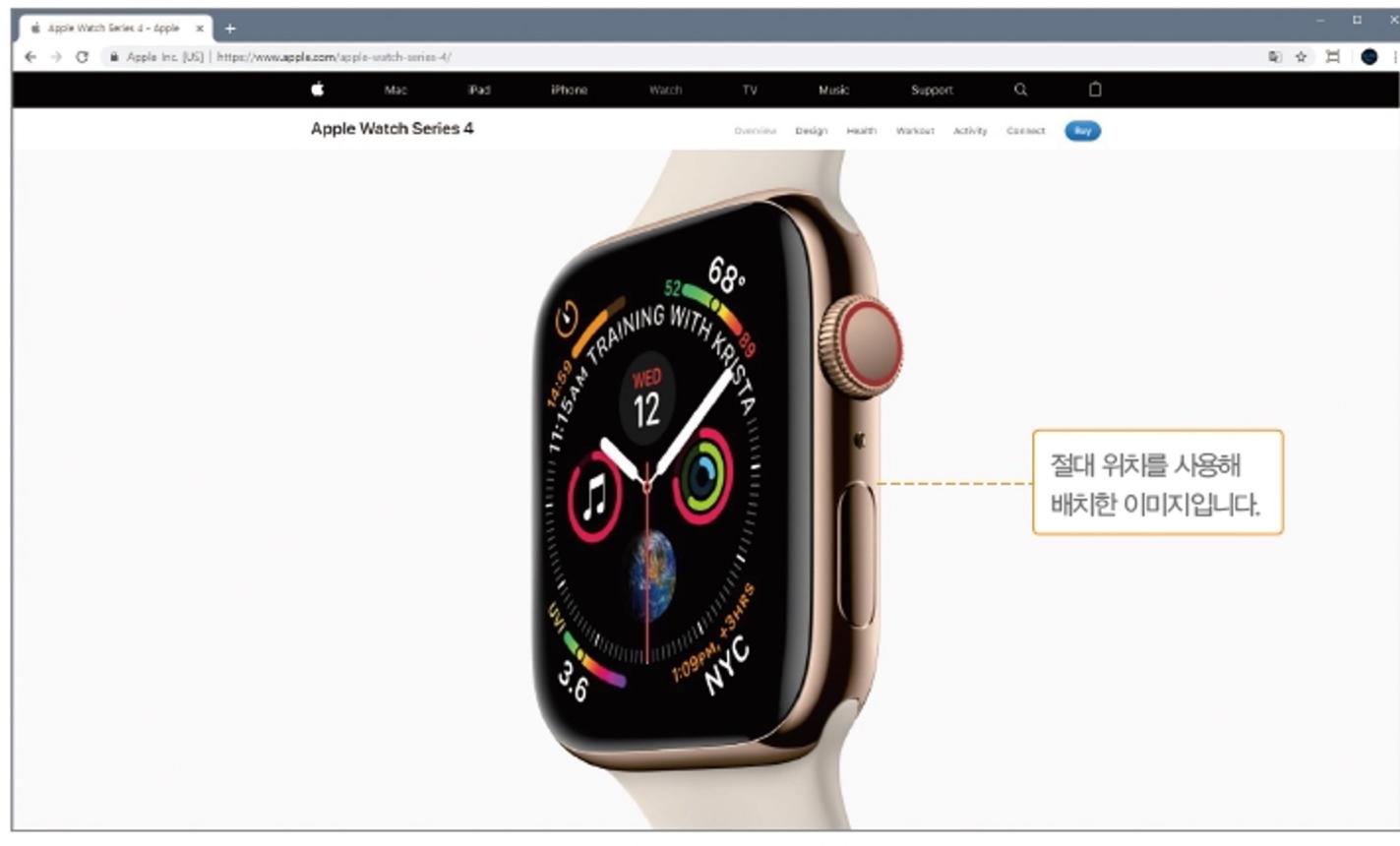


그림 7-6 절대 위치를 사용한 예

02 요소 배치

■ [응용 예제 7-4] 절대 위치를 사용해 요소 배치(1)

[코드 데모](#)

- 자손의 position 속성에 absolute 키워드를 적용하려면
부모에 height 속성을 입력

코드 7-4 example_4.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Absolute Position</title>

    <style>
        #container {
            width: 500px; height: 300px;
            border: 3px solid black;
            overflow: hidden;
            position: relative;
        }
        .circle {
            position: absolute;
            width: 100px; height: 100px;
            border-radius: 50% 50%;
        }
        #red {
            background: red;
            left: 20px; top: 20px;
        }
    </style>

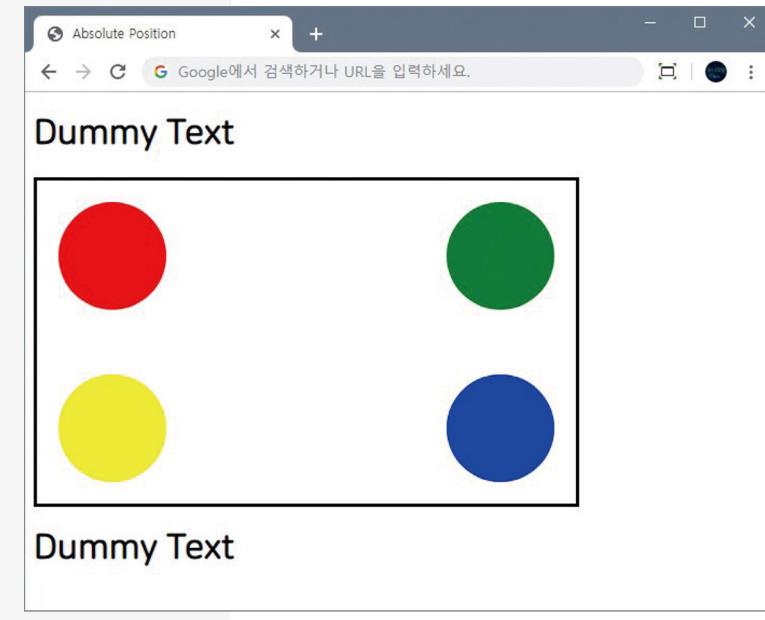
```

02 요소 배치

- [응용 예제 7-4] 절대 위치를 사용해 요소 배치(2)

[코드 데모](#)

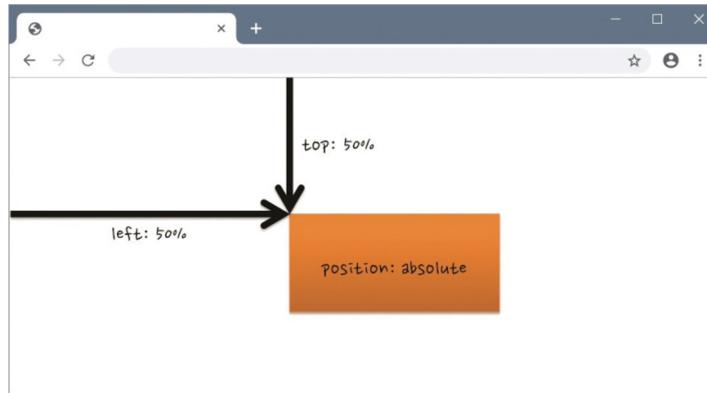
```
#green {  
    background: green;  
    right: 20px; top: 20px;  
}  
  
#blue {  
    background: blue;  
    right: 20px; bottom: 20px;  
}  
  
#yellow {  
    background: yellow;  
    left: 20px; bottom: 20px;  
}  
/  
/style>  
</head>  
<body>  
    <h1>Dummy Text</h1>  
    <div id="container">  
        <div id="red" class="circle"></div>  
        <div id="green" class="circle"></div>  
        <div id="blue" class="circle"></div>  
        <div id="yellow" class="circle"></div>  
    </div>  
    <h1>Dummy Text</h1>  
</body>  
</html>
```



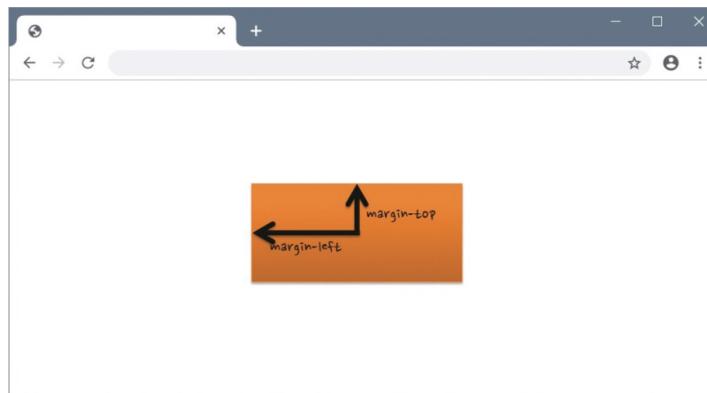
02 요소 배치

■ 요소를 중앙에 배치

- ① 중앙 정렬하려는 div 태그의 position 속성을 absolute로 지정
- ② left 속성과 top 속성을 모두 50%로 지정



- ③ 중앙에 정렬하려는 div 태그의 margin-left 속성과 margin-top 속성에 음수 입력
 - 입력하는 값은 div 태그 너비와 높이의 정확히 반이어야 함



■ [응용 예제 7-5] 요소의 중앙 배치

[코드 데모](#)

코드 7-5 example_5.html

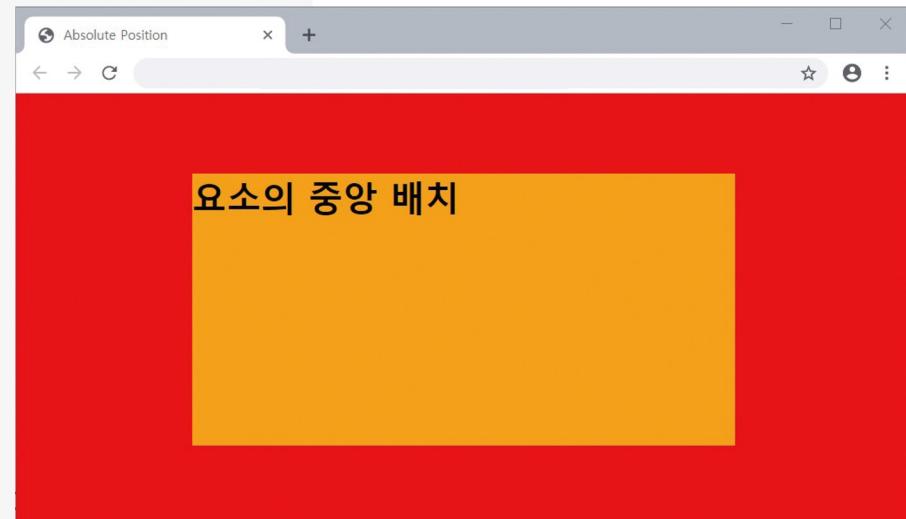
```
<!DOCTYPE html>
<html>
<head>
    <title>Absolute Position</title>
    <style>
        /* 초기화 */
        * { margin: 0; padding: 0; }

        /* 주제 */
        body { background: red; }

        #container {
            /* 색상 및 크기 적용 */
            width: 500px; height: 250px;
            background: orange;

            /* 위치 설정 */
            position: absolute;
            left: 50%; top: 50%;
            margin-left: -250px;
            margin-top: -125px;
        }
    </style>
</head>
```

```
<body>
    <div id="container">
        <h1>요소의 중앙 배치</h1>
    </div>
</body>
</html>
```



02 요소 배치

■ 요소를 고정 위치에 배치

- 웹 페이지의 상하좌우에 붙어서 사용자를 따라다니는 요소

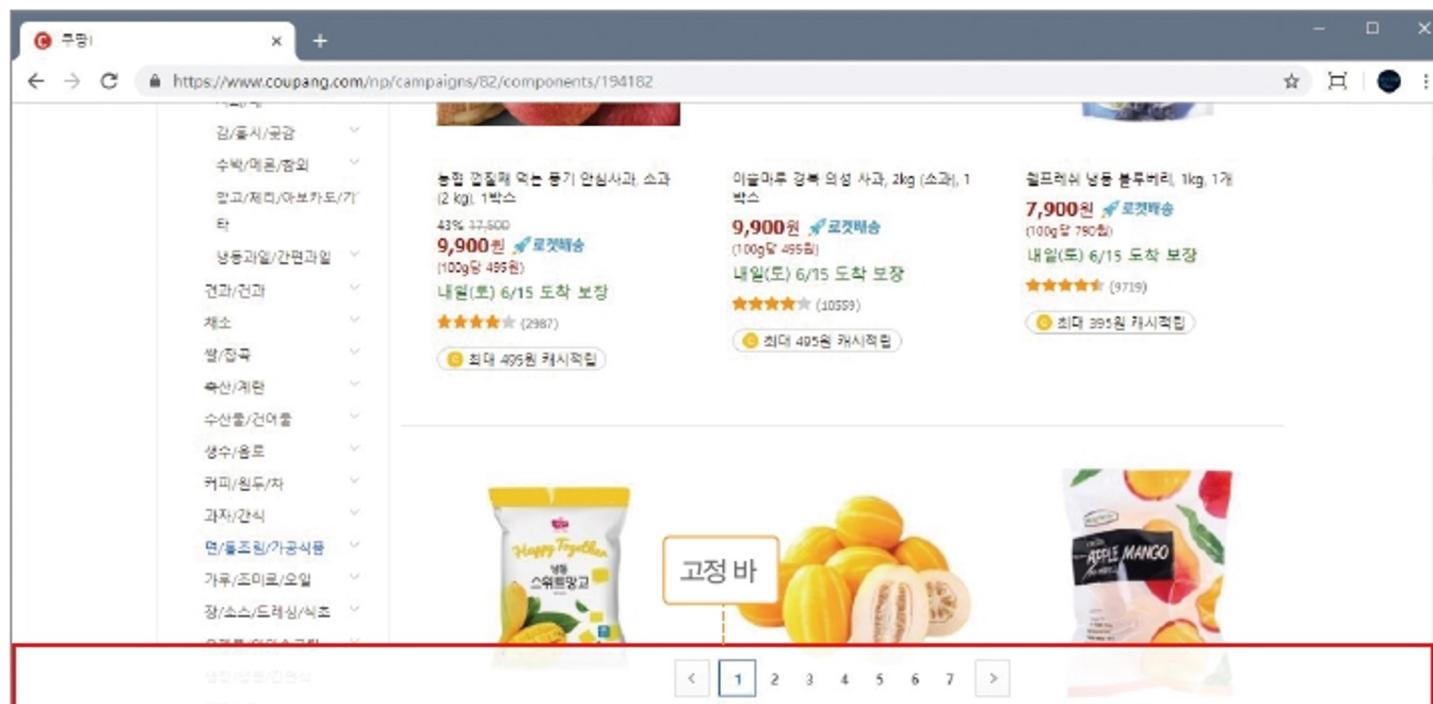


그림 7-7 고정 바 사용 예

■ [응용 예제 7-6] 고정 바 배치(1)

[코드 데모](#)

- ① position 스타일 속성에 fixed를 적용
- ② left 속성, top 속성, right 속성, bottom 속성에 값을 입력해 위치를 설정
- ③ width 속성과 height 속성으로 크기를 설정

코드 7-6 example_6.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Fixed Bar</title>
    <style>
        .container {
            margin-top: 50px;
            margin-left: 50px;
        }

        .top_bar {
            background: red;
            position: fixed;
            left: 0; top: 0; right: 0;
            height: 50px;
        }
    </style>
</head>
<body>
</body>
</html>
```

02 요소 배치

■ [응용 예제 7-6] 고정 바 배치

[코드 데모](#)

```
.left_bar {  
    background: blue;  
    position: fixed;  
    left: 0; top: 50px; bottom: 0;  
    width: 50px;  
}  
  
</style>  
</head>  
<body>  
    <div class="top_bar"></div>  
    <div class="left_bar"></div>  
    <div class="container">  
        <p>Lorem ipsum dolor sit amet.</p>  
        <p>Donec eros neque.</p>  
        <p>Nam et metus vitae tellus mattis bibendum.</p>  
        <p>Pellentesque hendrerit placerat.</p>  
    </div>  
</body>  
</html>
```



03 글자 생략

■ [응용 예제 7-7] 글자 생략

[코드 데모](#)

- ellipsis 클래스로 구현

코드 7-7 example_7.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Ellipsis</title>
    <style>
        h1, p {
            width: 300px;
        }

        .ellipsis {
            white-space: nowrap;
            overflow: hidden;
            text-overflow: ellipsis;
        }
    </style>
</head>
<body>
    <h1 class="ellipsis">Lorem ipsum dolor sit amet, consectetur</h1>
    <p class="ellipsis">Lorem ipsum dolor sit amet, consectetur</p>
</body>
</html>
```

Lorem ipsum dolor...

Lorem ipsum dolor sit amet, consectetur adi...

01 반응형 웹 소개

■ 반응형 웹

- 웹 페이지 하나로도 데스크톱, 태블릿PC, 스마트폰에 맞게 디자인이 자동으로 반응해서 변경되는 웹 페이지
- 장점
 - 개발 효율성, 유지 보수 용이
 - 미디어 쿼리(media query)를 사용해 개발

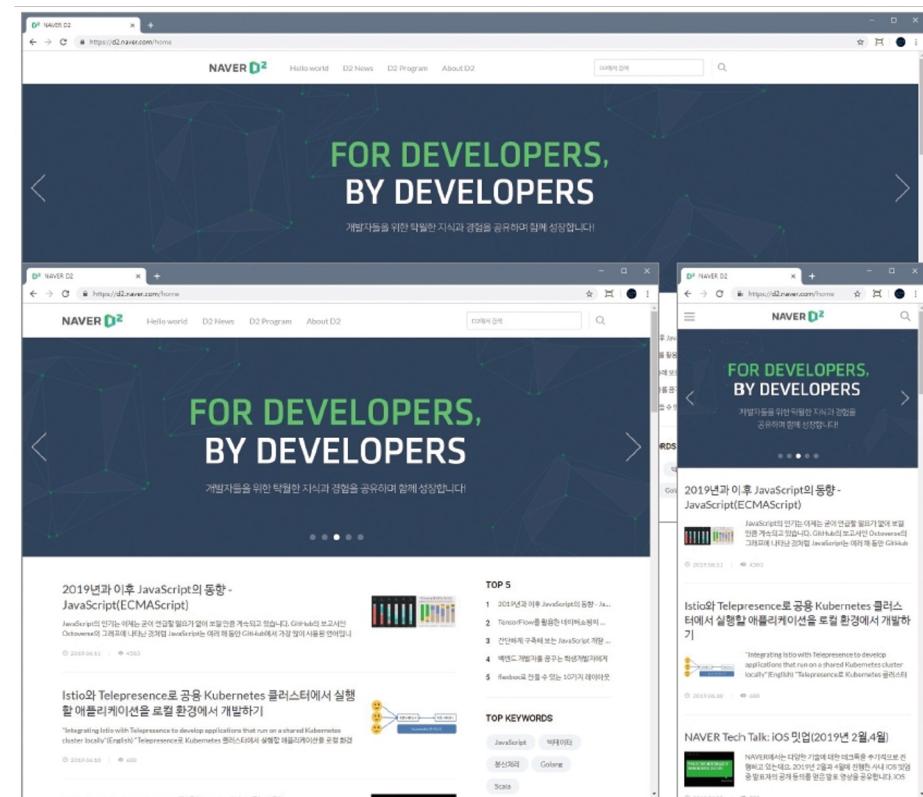


그림 8-1 화면 너비에 따라 레이아웃이 변경되는 반응형 웹 예

02 반응형 웹을 위한 설정

■ 뷰포트 설정

- meta 태그 : 추가적인 웹 페이지 정보를 제공

(a)

```
<meta name="title" content="ITCookbook HTML5 프로그래밍을 위한 페이지">
<meta name="description" content="meta 태그의 title 속성과 description 속성입니다.">
```

(b) **ITCookbook HTML5 프로그래밍을 위한 페이지**
<http://test.test.test.co.kr> ▾
2017.12.09 meta 태그의 title 속성과 description 속성입니다.

그림 8-2 meta 태그를 읽은 검색 엔진

- 뷰포트와 관련한 meta 태그 설정
 - viewport meta 태그 - name 속성에 viewport가 입력된 meta 태그

```
<meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
```

02 반응형 웹을 위한 설정

■ 뷰포트 설정

- viewport meta 태그의 content 속성에 입력할 수 있는 값

표 8-1 viewport meta 태그에 입력할 수 있는 값

값	설명	예
width	화면 너비	width=240
height	화면 높이	height=800
initial-scale	초기 확대 비율	initial-scale=2.0
user-scalable	확대 및 축소 가능 여부	user-scalable=no
minimum-scale	최소 축소 비율	minimum-scale=1.0
maximum-scale	최대 확대 비율	maximum-scale=2.0
target-densitydpi	DPI 지정	target-densitydpi=medium-dpi

02 반응형 웹을 위한 설정

- [기본 예제 8-1] 뷰포트 설정
 - 1. viewport meta 태그 사용하지 않기

[코드 데모](#)[온라인 에디터](#)

코드 8-1 without_meta.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Without Viewport Meta</title>
</head>
<body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec malesuada vel turpis lectus fermentum. Fusce nonummy maximus finibus. Non ac dictum velit. Donec sit amet bibendum enim. Sed varius erat et. Nullam maximus, libero id sollicitudin malesuada, non nisi velut neque, via elephas dui nunc ac tellus. Integer sit amet turpis metus. Nam id efficitur velit. Mauris accumsan, a leo et diam fermentum, nulla orci fermentum quam, a condimentum magna justo dictum sit. Aenean et odio metus, maximus est sed, fringilla est. Sed eget porta ut tincidunt elementum aliquet in eu mi. Donec id scelerisque erat fermentum et, aliquip ex ipsue.



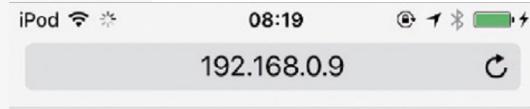
01 수평, 중앙, One True 정렬 레이아웃

- [기본 예제 8-1] 뷰포트 설정
 - 2. viewport meta 태그 사용하기

[코드 데모](#)

코드 8-2 with_meta.html

```
<!DOCTYPE html>
<html>
<head>
  <title>With Viewport Meta</title>
  <meta name="viewport" content="user-scalable=no,initial-scale=1">
</head>
<body>
  <h1>Lorem ipsum dolor sit amet</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec mollis urna sed turpis luctus laoreet. Fusce auctor maximus finibus. Nam ac dictum velit. Donec sit amet bibendum erat. Sed varius arcu est. Nullam maximus, libero id sollicitudin malesuada, nunc nunc varius neque, eu rhoncus dui nunc ac felis. Integer sit amet turpis metus. Nam id efficitur velit. Mauris suscipit, odio et dictum fermentum, nulla orci fermentum quam, a condimentum magna justo dictum leo. Aenean et odio mattis, maximus elit sed, fringilla est. Sed eget purus sit amet arcu elementum aliquet in eu mi. Donec dui tortor, viverra in fermentum ut, aliquam ac ipsum.



02 반응형 웹을 위한 설정

■ 미디어 쿼리 설정

- @-규칙@-rule
 - 스타일시트 내부에서 특정한 규칙을 표현하는 데 사용
 - @import 규칙 - 외부 스타일을 가져옴
 - @font-face 규칙 - 글꼴을 추가로 정의

```
@media (<미디어 쿼리>) {  
    <CSS 코드>  
}
```

(a) @media 규칙

- media 속성

- link 태그에 입력해서 해당 미디어 쿼리 조건에 맞는 장치에서만 CSS 파일을 불러올 때 사용

```
<link rel="stylesheet" href="<파일 이름>" media="<미디어 쿼리>">
```

(b) media 속성

그림 8-3 미디어 쿼리를 사용하는 방법

02 반응형 웹을 위한 설정

■ [기본 예제 8-2] 모니터와 프린터 구분

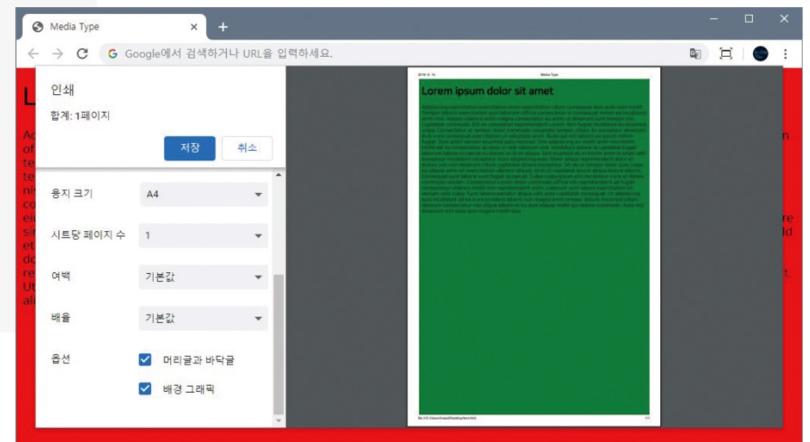
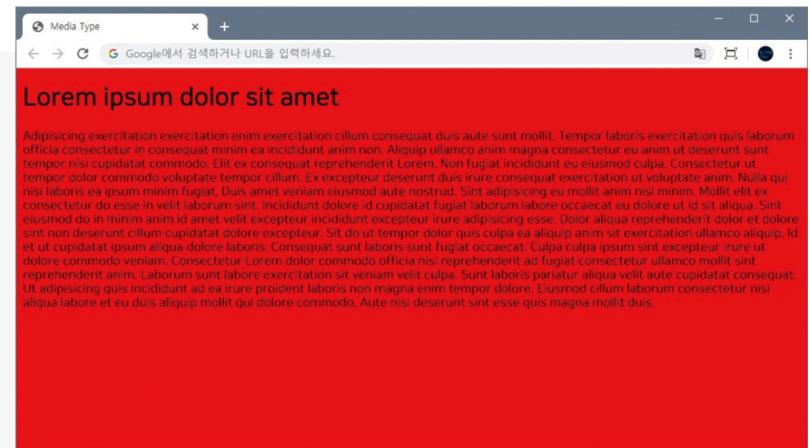
- 1. @ media 규칙을 사용한 미디어 쿼리 사용하기

[코드 데모](#)

코드 8-3 at_media_rule.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Media Type</title>
    <style>
        @media screen {
            body { background-color: red; }
        }

        @media print {
            body { background-color: green; }
        }
    </style>
</head>
<body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



02 반응형 웹을 위한 설정

- [기본 예제 8-2] 모니터와 프린터 구분
 - 2. media 속성을 사용한 미디어 쿼리 사용하기

[코드 데모](#)

코드 8-4 media_attribute.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Media Type</title>
    <link rel="stylesheet" href="screen.css" media="screen">
    <link rel="stylesheet" href="print.css" media="print">
</head>
<body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

코드 8-5 screen.css

```
@media screen {
    body {
        background-color: red;
    }
}
```

코드 8-6 print.css

```
@media print {
    body {
        background-color: green;
    }
}
```

02 반응형 웹을 위한 설정

■ 미디어 타입과 특징

표 8-2 지원하는 미디어 타입

미디어 타입	설명
all	모든 장치
aural	음성 장치
braille	표시 전용 점자 장치
handheld	손으로 들고 다니는 작은 장치
print	프린터
projection	프로젝터
screen	화면
tty	터미널 등 그림을 띄울 수 없는 장치
tv	텔레비전
embossed	인쇄 전용 점자 장치

표 8-3 미디어 타입 연산자

연산자	설명
only	'해당 장치에서만'을 나타냄
not	'해당 장치를 제외한'을 나타냄

02 반응형 웹을 위한 설정

■ 미디어 탑입과 특징

- orientation을 제외한 모든 속성은 min 접두사와 max 접두사를 붙일 수 있음
- and 연산자를 사용해 조건을 합칠 수도 있음

표 8-4 미디어 특징

미디어 특징	설명
width	화면 너비
height	화면 높이
device-width	장치 너비
device-height	장치 높이
orientation	장치 방향
device-aspect-ratio	화면 비율
color	장치 색상 비트
color-index	장치에서 표현 가능한 최대 색상 개수
monochrome	흑백 장치의 픽셀당 비트 개수
resolution	장치 해상도

02 반응형 웹을 위한 설정

■ [기본 예제 8-3] 스마트폰, 태블릿PC, 데스크톱 구분(1)

[코드 데모](#)

- 스마트폰 -빨간색 배경
- 태블릿PC 세로 화면 - 초록색 배경
- 태블릿PC 가로 화면과 데스크톱 - 파란색 배경

코드 8-7 media_feature.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Media Feature</title>
    <meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
    <style>
        /* 스마트폰 */
        @media screen and (max-width: 767px) {
            body { background-color: red; }
        }

        /* 태블릿PC 세로 */
        @media screen and (min-width: 768px) and (max-width: 959px) {
            body { background-color: green; }
        }
    </style>
</head>
<body>
</body>
</html>
```

02 반응형 웹을 위한 설정

- [기본 예제 8-3] 스마트폰, 태블릿PC, 데스크톱 구분(2)

[코드 데모](#)

```
/* 데스크톱 */
@media screen and (min-width: 960px) {
    body { background-color: blue; }
}
</style>
</head>
<body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



02 반응형 웹을 위한 설정

■ [기본 예제 8-4] 화면 방향 전환

[코드 데모](#)

- orientation 속성 사용 - 화면 방향에 따라 다른 배경 색상을 적용

```
<!DOCTYPE html>
<html>
<head>
    <title>Media Feature</title>
    <meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
    <style>
        @media screen and (orientation: portrait) {
            body { background-color: red; }
        }

        @media screen and (orientation: landscape) {
            body { background-color: green; }
        }
    </style>
</head>
<body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



03 반응형 웹 패턴

■ 반응형 웹 패턴

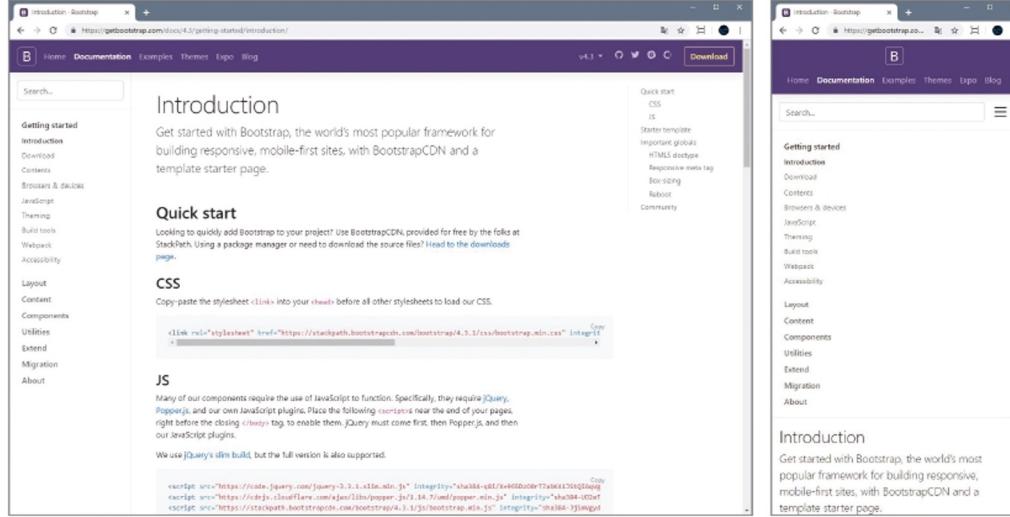


그림 8-4 화면이 작아졌을 때 왼쪽 메뉴가 위쪽으로 이동
(<https://getbootstrap.com/docs/4.3/getting-started/introduction/>)

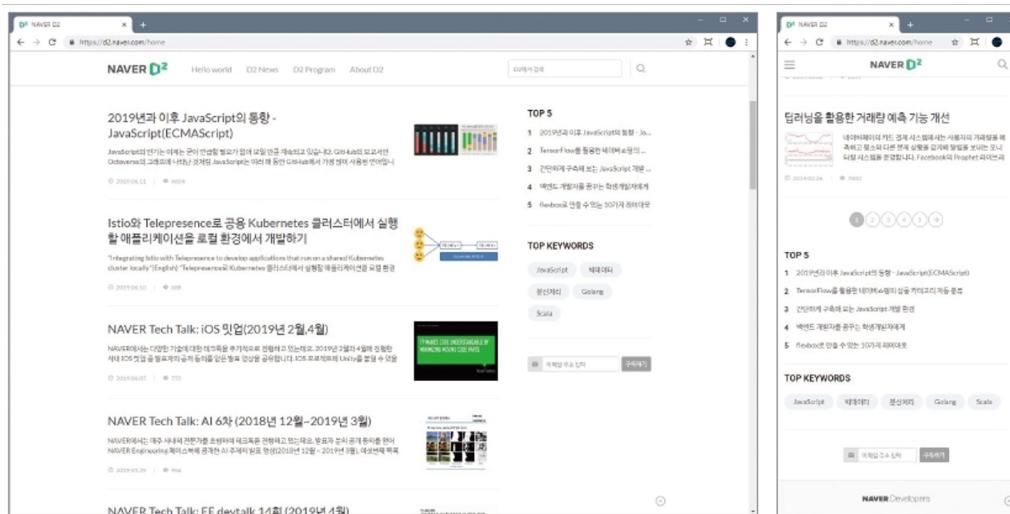


그림 8-5 화면이 작아졌을 때 오른쪽 메뉴가 아래쪽으로 이동(<https://d2.naver.com/home>)

03 반응형 웹 패턴

■ 반응형 웹 패턴

▪ 규칙

- HTML 태그 구성은 모바일 장치를 기준으로 함
- float 태그의 left와 right를 활용해 적당한 위치에 놓음

■ [응용 예제 8-5] 반응형 웹 패턴 ① (1)

[코드 데모](#)

- 1. 데스크톱에서는 메뉴가 왼쪽에 있고, 모바일 장치에서는 메뉴가 위쪽에 있는 패턴

코드 8-9 pattern1.html

```
<!DOCTYPE html>
<html>
<head>
    <title>패턴1</title>
    <meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        body {
            width: 960px;
            margin: 0 auto;
            overflow: hidden;
        }

        #menu {
            width: 260px;
            float: left;
        }
    </style>
</head>
<body>
    <div id="menu"></div>
    <div>사이트 내용</div>
</body>
</html>
```

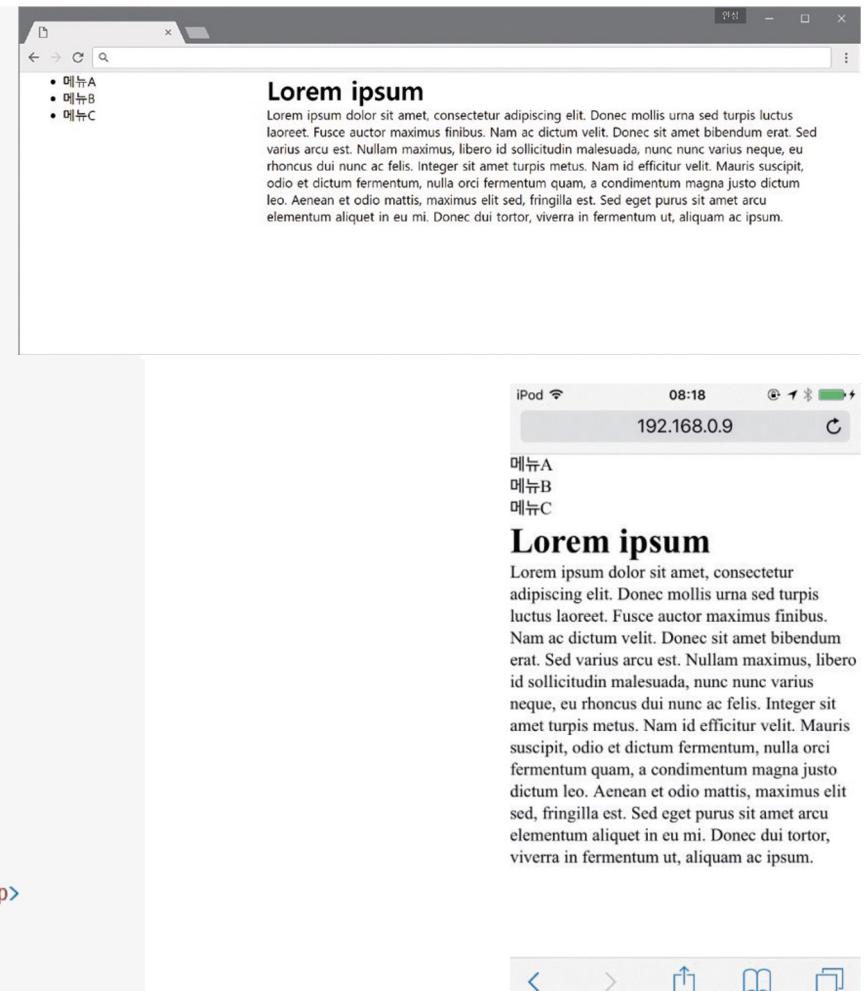
03 반응형 웹 패턴

■ [응용 예제 8-5] 반응형 웹 패턴 ① (2)

[코드 데모](#)

- 1. 데스크톱에서는 메뉴가 왼쪽에 있고, 모바일 장치에서는 메뉴가 위쪽에 있는 패턴

```
#section {  
    width: 700px;  
    float: right; /* left로 입력해도 상관없습니다. */  
}  
  
li { list-style: none; }  
  
@media screen and (max-width: 767px) {  
    /* 스마트폰 사이즈에서는 전부 해제합니다. */  
    body { width: auto; }  
    #menu { width: auto; float: none; }  
    #section { width: auto; float: none; }  
}  
    </style>  
</head>  
<body>  
    <div id="menu">  
        <ul>  
            <li>메뉴A</li>  
            <li>메뉴B</li>  
            <li>메뉴C</li>  
        </ul>  
    </div>  
    <div id="section">  
        <h1>Lorem ipsum</h1>  
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
    </div>  
</body>  
</html>
```



03 반응형 웹 패턴

■ [응용 예제 8-6] 반응형 웹 패턴 ② (1)

[코드 데모](#)

- 2. 데스크톱에서는 메뉴가 오른쪽에 있고, 모바일 장치에서는 메뉴가 위쪽에 있는 패턴

코드 8-10 pattern2.html

```
<!DOCTYPE html>
<html>
<head>
    <title>패턴2</title>
    <meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        body {
            width: 960px;
            margin: 0 auto;
            overflow: hidden;
        }

        #menu {
            width: 260px;
            float: right;
        }

        #section {
            width: 700px;
            float: left;
        }

        li { list-style: none; }
```

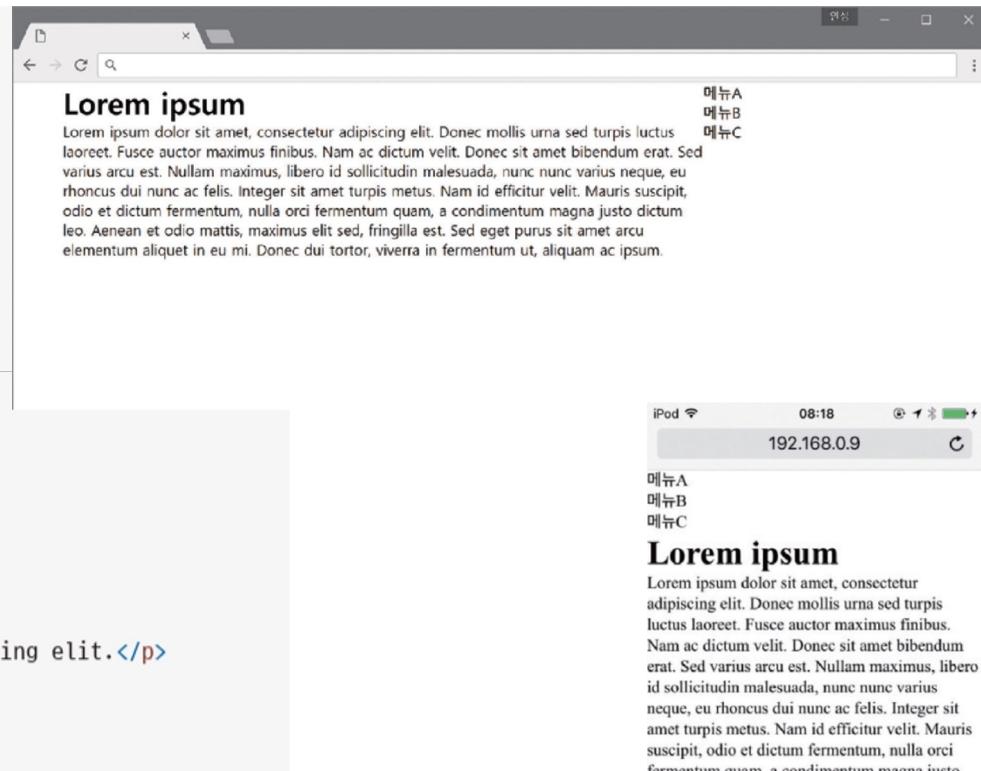
03 반응형 웹 패턴

■ [응용 예제 8-6] 반응형 웹 패턴 ② (2)

[코드 데모](#)

- 2. 데스크톱에서는 메뉴가 오른쪽에 있고, 모바일 장치에서는 메뉴가 위쪽에 있는 패턴

```
@media screen and (max-width: 767px) {  
    /* 스마트폰 사이즈에서는 전부 해제합니다. */  
    body { width: auto; }  
    #menu { width: auto; float: none; }  
    #section { width: auto; float: none; }  
}  
</style>  
</head>  
<body>  
    <div id="menu">  
        <li>메뉴A</li>  
        <li>메뉴B</li>  
        <li>메뉴C</li>  
    </ul>  
    </div>  
    <div id="section">  
        <h1>Lorem ipsum</h1>  
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
    </div>  
</body>  
</html>
```



03 반응형 웹 패턴

■ [응용 예제 8-7] 반응형 웹 패턴 ③ (1)

[코드 데모](#)

- 3. 데스크톱에서는 메뉴가 왼쪽에 있고, 모바일 장치에서는 메뉴가 아래쪽에 있는 패턴

코드 8-11 pattern3.html

```
<!DOCTYPE html>
<html>
<head>
    <title>패턴3</title>
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        body {
            width: 960px;
            margin: 0 auto;
            overflow: hidden;
        }

        #menu {
            width: 260px;
            float: left;
        }

        #section {
            width: 700px;
            float: right; /* left로 입력해도 상관없습니다. */
        }
    </style>
</head>
<body>
    <div id="menu"></div>
    <div id="section"></div>
</body>

```

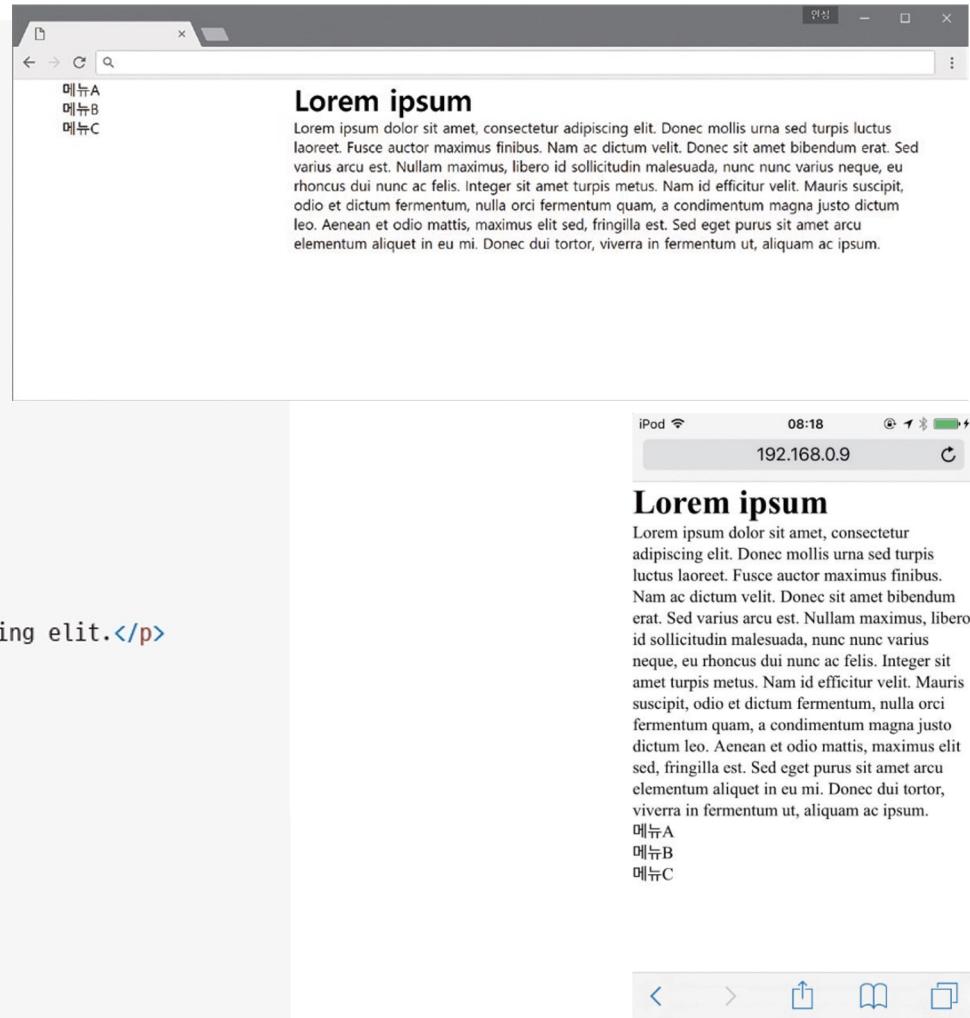
03 반응형 웹 패턴

■ [응용 예제 8-7] 반응형 웹 패턴 ③ (2)

[코드 데모](#)

- 3. 데스크톱에서는 메뉴가 왼쪽에 있고, 모바일 장치에서는 메뉴가 아래쪽에 있는 패턴

```
#section {  
    width: 700px;  
    float: right; /* left로 입력해도 상관없습니다. */  
}  
  
li { list-style: none; }  
  
@media screen and (max-width: 767px) {  
    /* 스마트폰 사이즈에서는 전부 해제합니다. */  
    body { width: auto }  
    #menu { width: auto; float: none; }  
    #section { width: auto; float: none; }  
}  
  
<body>  
    <div id="section">  
        <h1>Lorem ipsum</h1>  
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
    </div>  
    <div id="menu">  
        <ul>  
            <li>메뉴A</li>  
            <li>메뉴B</li>  
            <li>메뉴C</li>  
        </ul>  
    </div>  
</body>  
</html>
```



03 반응형 웹 패턴

■ [응용 예제 8-8] 반응형 웹 패턴 ④ (1)

[코드 데모](#)

- 4. 데스크톱에서는 메뉴가 오른쪽에 있고, 모바일 장치에서는 메뉴가 아래쪽에 있는 패턴

코드 8-12 pattern4.html

```
<!DOCTYPE html>
<html>
<head>
    <title>패턴4</title>
    <meta name="viewport" content="user-scalable=no,initial-scale=1,maximum-scale=1">
    <style>
        * {
            margin: 0;
            padding: 0;
        }
        body {
            width: 960px;
            margin: 0 auto;
            overflow: hidden;
        }
        #menu {
            width: 260px;
            float: right;
        }
    </style>
</head>
<body>
    <div id="menu"></div>
</body>
</html>
```

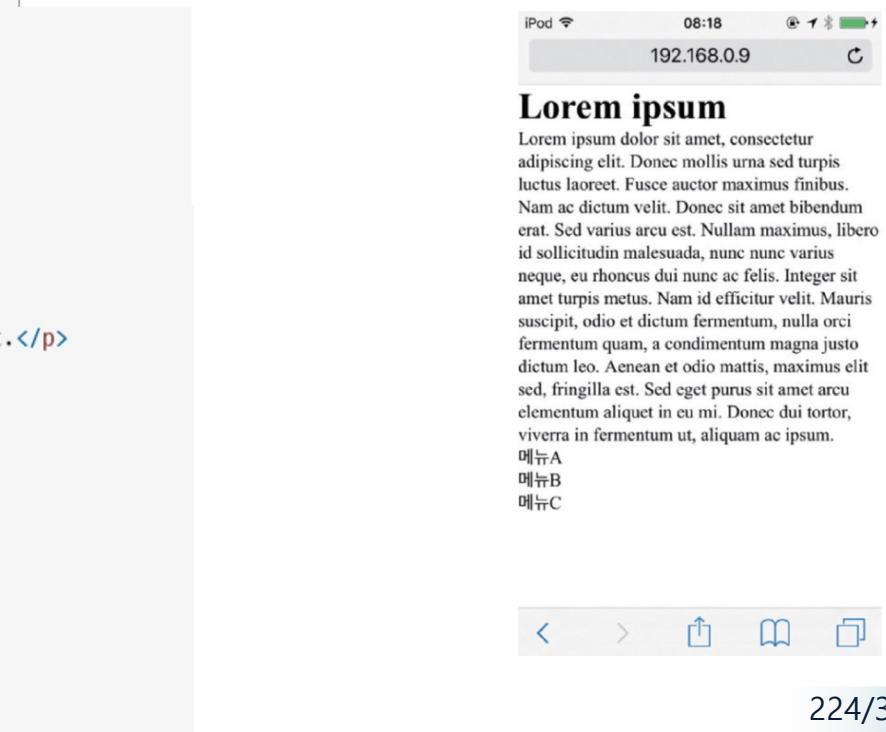
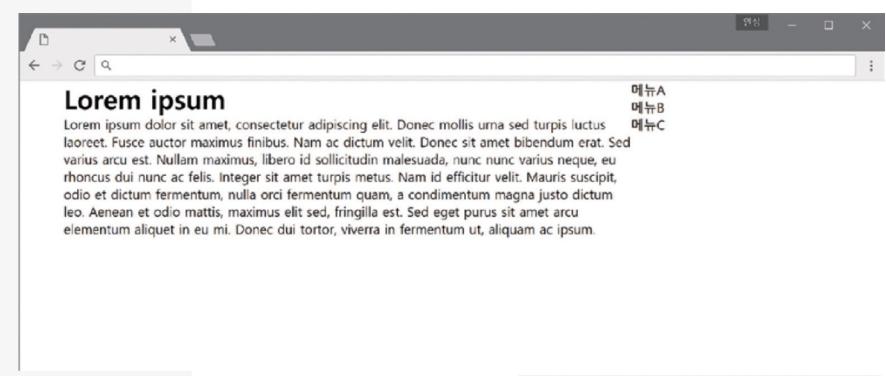
03 반응형 웹 패턴

■ [응용 예제 8-8] 반응형 웹 패턴 ④ (2)

[코드 데모](#)

- 4. 데스크톱에서는 메뉴가 오른쪽에 있고, 모바일 장치에서는 메뉴가 아래쪽에 있는 패턴

```
#section {  
    width: 700px;  
    float: left;  
}  
  
li { list-style: none; }  
  
@media screen and (max-width: 767px) {  
    /* 스마트폰 사이즈에서는 전부 해제합니다. */  
    body { width: auto }  
    #menu { width: auto; float: none; }  
    #section { width: auto; float: none; }  
}  
    </style>  
</head>  
<body>  
    <div id="section">  
        <h1>Lorem ipsum</h1>  
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
    </div>  
    <div id="menu">  
        <ul>  
            <li>메뉴A</li>  
            <li>메뉴B</li>  
            <li>메뉴C</li>  
        </ul>  
    </div>  
</body>  
</html>
```



01 자바스크립트 기본 용어와 출력 방법

■ 자바스크립트 기본 용어

- 표현식 - 값을 만들어 내는 간단한 코드
- 문장 - 프로그래밍 언어에 실행할 수 있는 코드의 최소 단위
- 종결 - 문장 마지막에 세미콜론(;) 또는 줄 바꿈

```
273;  
10 + 20 + 30 * 2;  
var name = '윤' + '인' + '성';  
alert('Hello JavaScript');
```

(a) 표현식 예

```
273  
10 + 20 + 30 * 2  
'JavaScript'
```

(b) 문장 예

그림 9-1 표현식과 문장 예

01 자바스크립트 기본 용어와 출력 방법

■ 자바스크립트 기본 용어

- 키워드 - 자바스크립트를 처음 만들 때 정해진 특별한 의미가 부여된 단어

표 9-1 자바스크립트 키워드

break	else	instanceof	true	case	false
new	try	catch	finally	null	typeof
continue	for	return	var	default	function
switch	void	delete	if	this	while
do	in	throw	with	const	class

- 식별자 - 자바스크립트에서 변수나 함수 등에 이름을 붙일 때 사용하는 단어

사용할 수 있는 예

alpha
alpha10
_alpha
\$alpha
Alpha
ALPHA

사용할 수 없는 예

break
273alpha
has space

• 키워드를 사용하면 안 됩니다.

• 특수 문자는 _과 \$만 허용합니다.
• 숫자로 시작하면 안 됩니다.
• 공백은 입력하면 안 됩니다.

(a) 식별자 예

(b) 식별자 생성 규칙

그림 9-2 식별자 예와 식별자 생성 규칙

01 자바스크립트 기본 용어와 출력 방법

- 자바스크립트 기본 용어
 - 식별자 생성 규칙

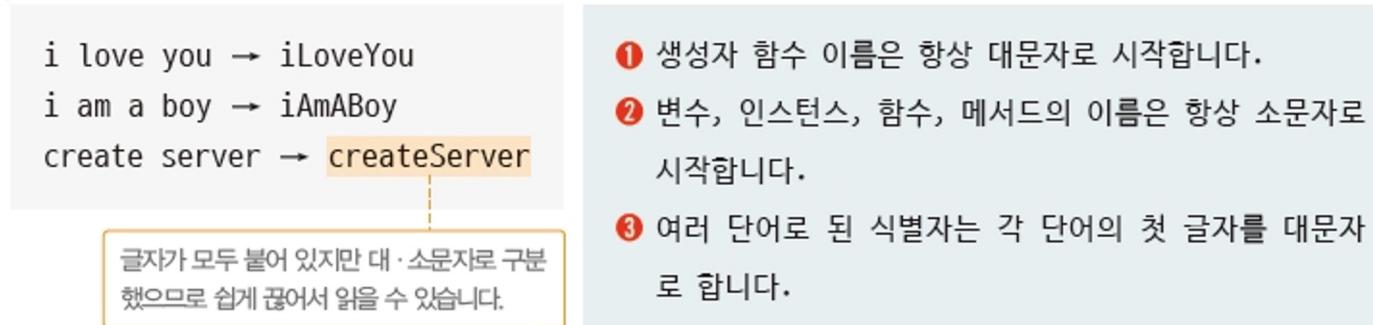


그림 9-3 식별자 생성 관례

표 9-2 자바스크립트의 식별자 종류

구분	단독으로 사용	다른 식별자와 함께 사용
식별자 뒤에 괄호 없음	변수	속성
식별자 뒤에 괄호 있음	함수	메서드

01 자바스크립트 기본 용어와 출력 방법

■ 자바스크립트 기본 용어

- 주석 - 프로그램 진행에 전혀 영향을 주지 않는 코드로, 프로그램을 설명하는 데 사용

표 9-3 주석 처리 방법

방법	형태
① 한 행 주석 처리	// 주석문
② 여러 행 주석 처리	/* 주석문 주석문 */

```
<script>  
    // 주석은 코드 실행에 영향을 주지 않습니다.  
    /*  
     alert('Hello JavaScript .. !');  
     alert('Hello JavaScript .. !');  
     alert('Hello JavaScript .. !');  
    */  
</script>
```

01 자바스크립트 기본 용어와 출력 방법

■ 자바스크립트 출력

- 가장 기본적인 출력 방법 - alert() 함수를 사용해 웹 브라우저에 경고 창을 띄우기

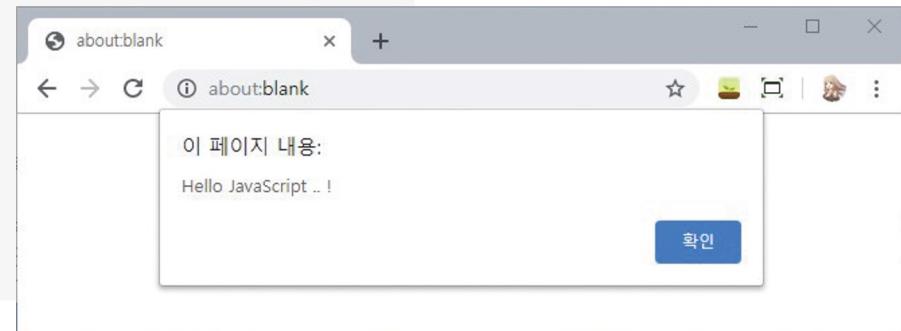
```
alert("메시지")
```

그림 9-4 alert() 함수 형태

■ [기본 예제 9-1] 자바스크립트를 이용한 메시지 출력 [코드 데모](#) [온라인 에디터](#)

코드 9-1 output_alert.html

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Basic</title>
    <script>
        alert('Hello JavaScript .. !');
    </script>
</head>
<body>
</body>
</html>
```



02 자료형과 변수

■ 자료형

■ 숫자

- 가장 기본적인 자료형
- 정수와 실수 구분하지 않음

표 9-4 사칙 연산자

연산자	설명	예
+	덧셈	$> 52 + 273$ 325 $> 52.273 + 103.57$ 155.843
-	뺄셈	$> 52 - 273$ -221 $> 52.273 - 103.57$ -51.29699999999999

자바스크립트는 부동소수점(소수점을 포함한 숫자)을 계산할 때 약간의 오차를 발생시킵니다.

연산자	설명	예
*	곱셈	$> 52 * 273$ 14196 $> 52.273 * 103.57$ 5413.91461

표 9-5 나머지 연산자

연산자	설명	예
%	나머지	$> 10 \% 5$ 0 $> 7 \% 3$ 1

연산자	설명	예
/	나눗셈	$> 52 / 273$ 0.19047619047619047 $> 52.273 / 103.57$ 0.504711789128126 $> 1 / 0$ Infinity

자바스크립트는 어떤 숫자를 0으로 나누었을 때, 무한을 나타내는 값 Infinity이 됩니다.

02 자료형과 변수

■ 자료형

▪ 문자열

- 문자 집합
 - 'abcdefg', 'Hello World', '안녕하세요.'

표 9-6 문자열 생성

방법	예
작은따옴표 사용	> 'Hello JavaScript .. !' "Hello JavaScript .. !" > '"문자열"입니다.' ""문자열"입니다."
큰따옴표 사용	> "Hello JavaScript .. !" "Hello JavaScript .. !" > "'문자열'입니다." "'문자열'입니다."

표 9-8 문자열 연결 연산자

연산자	설명	예
+	문자열 연결	> '가나다' + '라마' + '바사아' + '자자카타' + '파하' "가나다라마바사아자자카타파하"

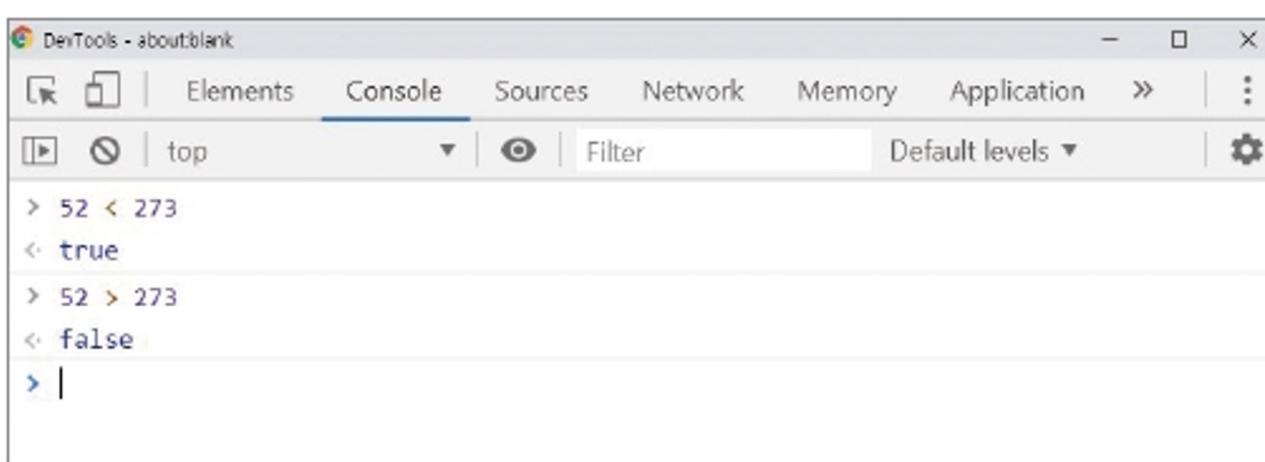
표 9-7 이스케이프 문자

이스케이프 문자	설명	예
\t	수평 탭	> '한빛\t아카데미' "한빛 아카데미"
\n	행 바꿈	> '한빛\n아카데미' "한빛 아카데미"
\w	역 슬래시	> '\\\\\\' \\
\'	작은따옴표	> '\"\""
\"	큰따옴표	> '\"\"\""

02 자료형과 변수

■ 자료형

- 불(bool)
 - 참과 거짓을 표현할 때 사용하는 자료



The screenshot shows the DevTools interface with the 'Console' tab selected. The console output area displays the following text:
> 52 < 273
< true
> 52 > 273
< false
> |

그림 9-5 불 사용 예

02 자료형과 변수

■ 자료형

▪ 비교 연산자

- 두 대상을 비교할 수 있는 연산자

표 9-9 비교 연산자

연산자	설명	예
\geq	좌변이 우변보다 크거나 같음	$> 10 \geq 20$ false $> '가방' > '하마'$ ----- 문자열 순서도 비교 가능합니다. false
\leq	우변이 좌변보다 크거나 같음	$> 10 \leq 20$ true
$>$	좌변이 우변보다 큼	$> 10 > 20$ false
$<$	우변이 좌변보다 큼	$> 10 < 20$ true
\equiv	좌변과 우변이 같음	$> 10 \equiv 20$ false
\neq	좌변과 우변이 다름	$> 10 \neq 20$ true

02 자료형과 변수

■ 자료형

▪ 논리 연산자

표 9-10 논리 연산자

연산자	설명	예
!	논리 부정(참이면 거짓, 거짓이면 참)	> !true false > !(10 == 10) false
&&	논리곱(둘 다 참이어야 참)	> true && true true ----- 둘 다 참일 때만 참입니다. > true && false false > false && true false > false && false false > (10 < 20) && (20 < 30) true ----- 20이 10~30 사이에 있다는 것을 나타낼 때는 이러한 형태로 표현해야 합니다.
	논리합(둘 중 하나만 참이어도 참)	> true true true > true false true > false true true ----- 둘 중 하나만 참이어도 참입니다. > false false false

02 자료형과 변수

■ 변수

- 값을 저장할 때 사용하는 식별자
- ① 변수를 선언합니다.
- ② 변수를 초기화합니다.

코드 9-3 변수 선언과 초기화

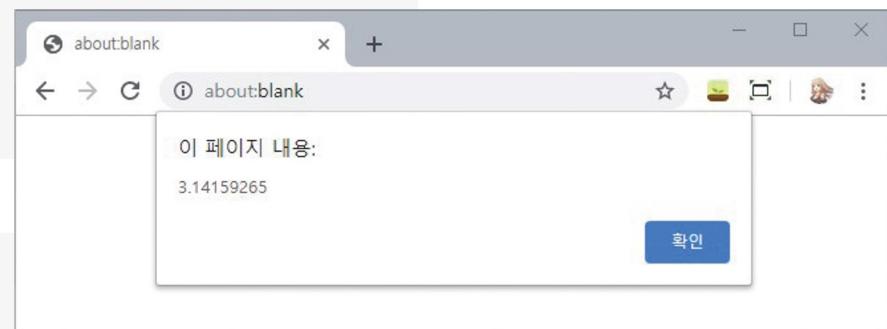
```
> var pi = 3.14159265;  
undefined
```

코드 9-2 변수 선언과 값 할당

```
① > var pi;           // 변수 선언  
undefined  
② > pi = 3.14159265; // 값 할당  
undefined
```

코드 9-4 변수에 저장된 값 출력

```
> var pi = 3.14159265;  
undefined  
> alert(pi);  
undefined
```



02 자료형과 변수

- [기본 예제 9-2] 자바스크립트를 이용한 메시지 출력(1)

[코드 데모](#)

- 1. HTML 페이지 만들기

코드 9-5 variable_use.html

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Basic</title>
  <script>

    </script>
</head>
<body>

</body>
</html>
```

02 자료형과 변수

- [기본 예제 9-2] 자바스크립트를 이용한 메시지 출력(2)

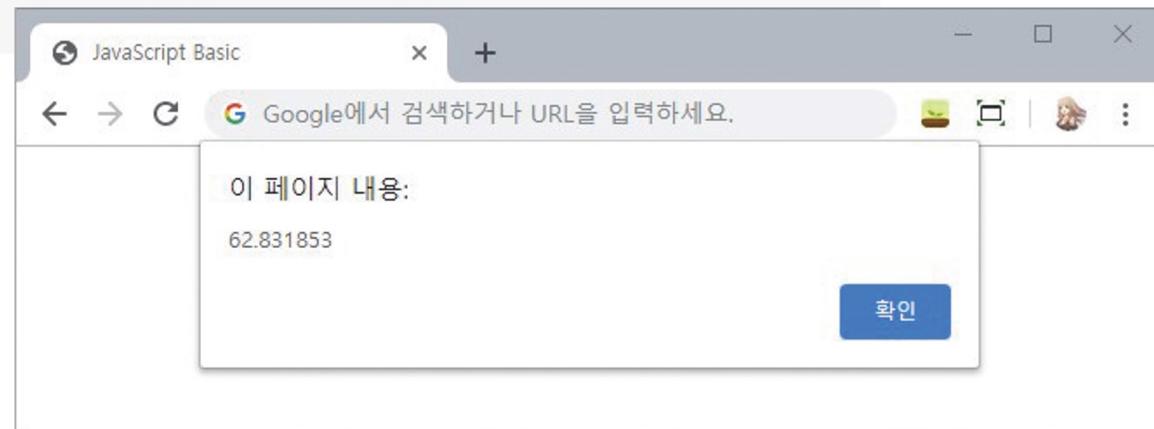
[코드 데모](#)

- 2. 변수 사용하기

코드 9-6 variable_use.html

```
<script>
    // 변수를 선언 및 초기화합니다.
    var radius = 10;
    var pi = 3.14159265;

    // 출력합니다.
    alert(2 * radius * pi);
</script>
```



03 조건문과 반복문

■ 조건문

▪ If 조건문

- 조건이 true이면 문장을 실행하고 false이면 문장 무시
- 실행 문장이 한 행이면 중괄호 생략 가능함
- 실행 문장이 여러 행이라면 중괄호 필요함

```
if (조건) {  
    문장  
}
```

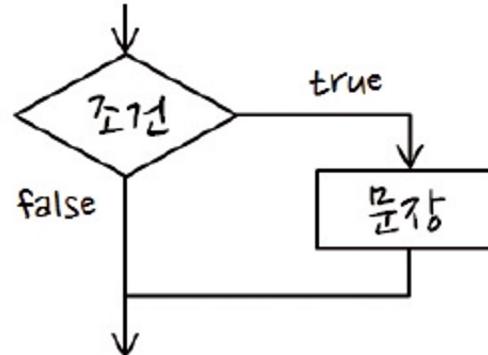


그림 9-6 if 조건문 기본 형태와 순서도

03 조건문과 반복문

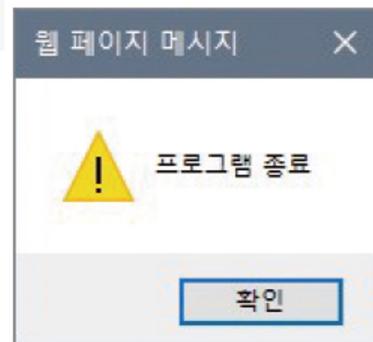
■ [기본 예제 9-3] if 조건문으로 참과 거짓 판별

[코드 데모](#)

코드 9-7 condition_basic.html

```
<script>
    // 조건문
    if (273 < 52) {
        // 표현식 "273 < 52"가 참일 때 실행합니다.
        alert('273 < 52 => true');

    }
    // 프로그램 종료
    alert('프로그램 종료');
</script>
```



크롬은 경고창이 너무 넓게 출력되므로 인터넷 익스플로러의 경고창을 스크린샷으로 넣었습니다. 크롬에서 이러한 모양의 경고창이 출력되지 않는다고 당황하지 마세요.

03 조건문과 반복문

- [기본 예제 9-4] if 조건문으로 오전과 오후 판별

[코드 데모](#)

- 1. 현재 시간 구하기

코드 9-8 condition_getDate.html

```
<script>
    // Date 객체를 선언합니다: 현재 시간 측정
    var date = new Date();

    // 요소를 추출합니다.
    var year = date.getFullYear();
    var month = date.getMonth() + 1;
    var day = date.getDay();
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var seconds = date.getSeconds();
</script>
```

03 조건문과 반복문

- [기본 예제 9-4] if 조건문으로 오전과 오후 판별

[코드 데모](#)

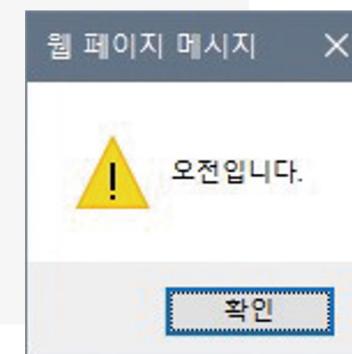
- 2. 오전과 오후 구분하기

코드 9-9 condition_date.html

```
<script>
    // 변수를 선언합니다.
    var date = new Date();
    var hours = date.getHours();

    // 조건문
    if (hours < 12) {
        // 표현식 "hours < 12"가 참일 때 실행합니다.
        alert('오전입니다.');
    }

    if (12 <= hours) {
        // 표현식 "12 <= hours"가 참일 때 실행합니다.
        alert('오후입니다.');
    }
</script>
```



03 조건문과 반복문

■ 조건문

▪ If else 조건문

- 두 가지로 분명하게 나뉘는 상황에서 편리하게 사용 가능
- 실행 문장이 한 행이면 중괄호 생략 가능함
- 실행 문장이 여러 행이라면 중괄호 필요함

```
if (조건) {  
    문장 1  
} else {  
    문장 2  
}
```

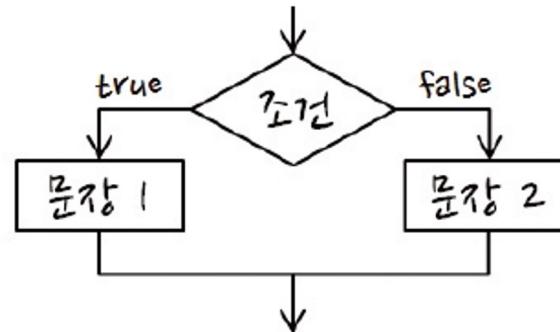


그림 9-7 if else 조건문 기본 형태와 순서도

03 조건문과 반복문

- [기본 예제 9-5] if else 조건문으로 오전과 오후 판별

[코드 데모](#)

코드 9-10 condition_else.html

```
<script>
    // 변수를 선언합니다.
    var date = new Date();
    var hours = date.getHours();

    // 조건문
    if (hours < 12) {
        // 표현식 "hours < 12"가 참일 때 실행합니다.
        alert('오전입니다.');
    } else {
        // 표현식 "hours < 12"가 거짓일 때 실행합니다.
        alert('오후입니다.');
    }
</script>
```

03 조건문과 반복문

■ 조건문

- 중첩 조건문과 if else if 조건문

```
if (조건) {  
    if (조건) {  
        문장  
    } else {  
        문장  
    }  
} else {  
    if (조건) {  
        문장  
    } else {  
        문장  
    }  
}
```

그림 9-8 중첩 조건문 형태

03 조건문과 반복문

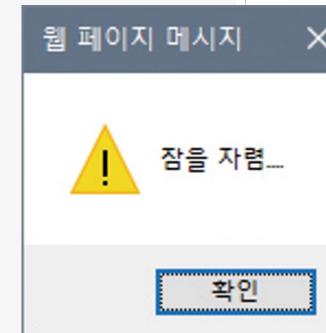
■ [기본 예제 9-6] 중첩 조건문으로 하루 일정 표현

[코드 데모](#)

코드 9-12 condition_ifelseif.html

```
<script>
  // Date 객체를 선언합니다: 현재 시간 측정
  var date = new Date();
  var hours = date.getHours();

  // 조건문
  if (hours < 5) {
    alert('잠을 자렴....');
  } else if (hours < 7) {
    alert('준비');
  } else if (hours < 9) {
    alert('출근');
  } else if (hours < 12) {
    alert('빈둥빈둥');
  } else if (hours < 14) {
    alert('식사');
  } else {
    // 여러 가지 업무를 수행합니다.
  }
</script>
```



03 조건문과 반복문

■ 조건문

- 중첩 조건문의 중괄호를 생략했을 때 만드는 조건문
 - if else if 조건문

```
if (조건) {  
    문장  
} else if (조건) {  
    문장  
} else if (조건) {  
    문장  
} else {  
    문장  
}
```

그림 9-9 if else if 조건문 형태

03 조건문과 반복문

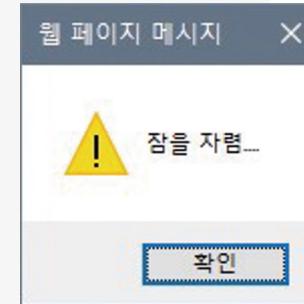
■ [기본 예제 9-7] if else if 조건문으로 하루 일정 표현

[코드 데모](#)

코드 9-11 condition_duplication.html

```
<script>
    // Date 객체를 선언합니다: 현재 시간 측정
    var date = new Date();
    var hours = date.getHours();

    // 조건문
    if (hours < 5) {
        alert('잠을 자렴....');
    } else {
        if (hours < 7) {
            alert('준비');
        } else {
            if (hours < 9) {
                alert('출근');
            } else {
                if (hours < 12) {
                    alert('빈둥빈둥');
                } else {
                    if (hours < 14) {
                        alert('식사');
                    } else {
                        // 여러 가지 업무를 수행합니다.
                    }
                }
            }
        }
    }
</script>
```



03 조건문과 반복문

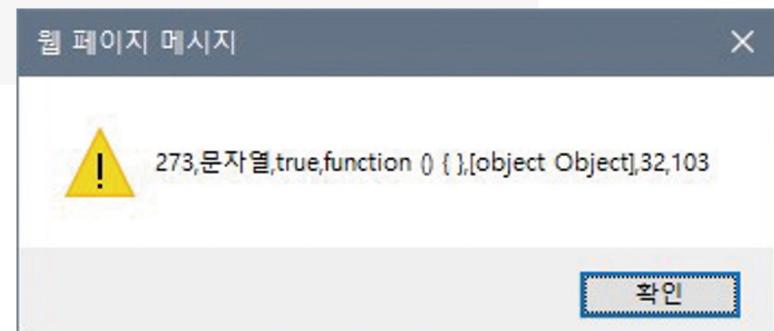
■ 반복문

▪ 배열

- 변수 여러 개를 한꺼번에 다룰 수 있는 자료형
- 요소 - 배열 내부에 입력된 자료 하나하나
- 배열 내부에 다양한 자료형을 입력 가능
- 배열 전체를 출력하면 요소가 순서대로 표시

코드 9-14 array_basic.html

```
<script>
  // 변수를 선언합니다.
  var array = [273, '문자열', true, function () { }, {}, [32, 103]];
  alert(array);
</script>
```



03 조건문과 반복문

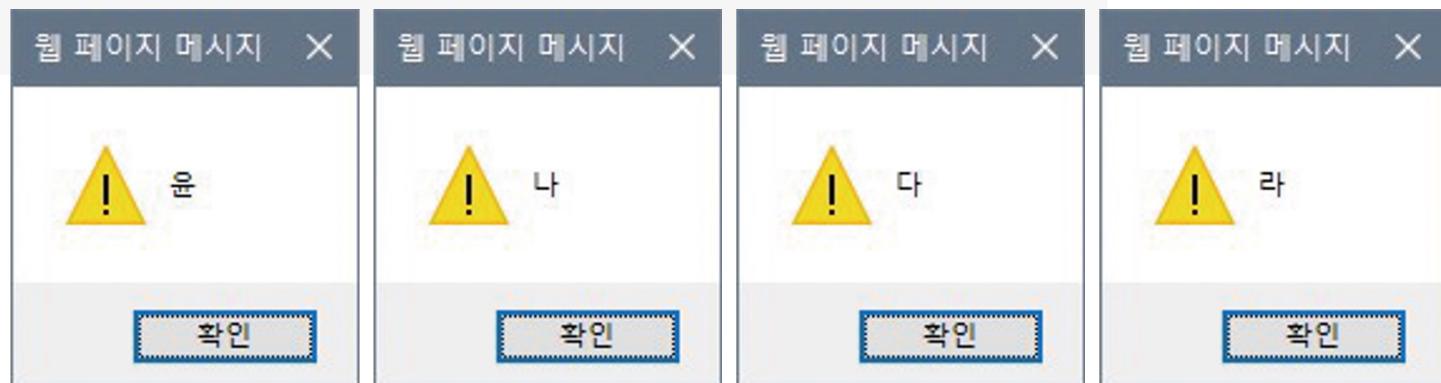
■ [기본 예제 9-8] 배열 생성과 배열 요소 접근

[코드 데모](#)

코드 9-15 loop_array.html

```
<script>
    // 변수를 선언합니다.
    var array = ['가', '나', '다', '라'];

    // 배열 요소를 변경합니다.
    array[0] = '윤';
    // 요소를 출력합니다.
    alert(array[0]);
    alert(array[1]);
    alert(array[2]);
    alert(array[3]);
</script>
```



03 조건문과 반복문

■ 반복문

▪ while 반복문

- 가장 기본적인 반복문
- if 조건문과 형식이 비슷하지만,
if 조건문과 달리 불(bool) 표현식이 참이면 중괄호 안 문장을 계속 실행

```
while (조건) {  
    문장  
}
```

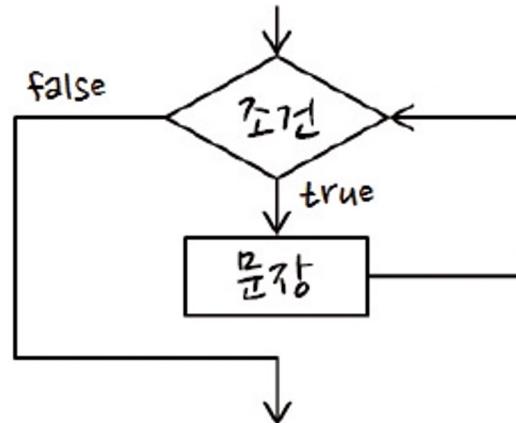


그림 9-10 while 반복문 기본 형태와 순서도

03 조건문과 반복문

■ [기본 예제 9-9] while 반복문

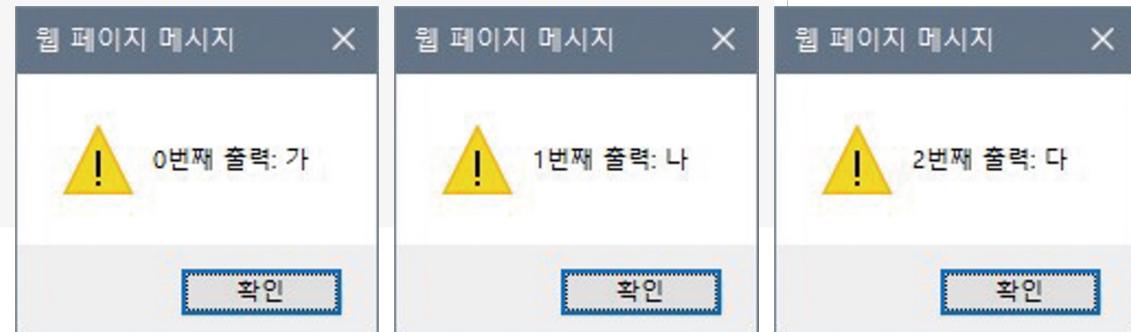
[코드 데모](#)

코드 9-17 loop_while.html

```
<script>
    // 변수를 선언합니다.
    var i = 0;
    var array = ['가', '나', '다'];

    // 반복을 수행합니다. i가 배열 원소 개수인 3보다 작을 때 반복합니다.
    while (i < array.length) {
        // 출력합니다.
        alert(i + '번째 출력: ' + array[i]);

        // 탈출하려고 변수를 더합니다.
        i++;
    }
</script>
```



03 조건문과 반복문

■ 반복문

▪ for 반복문

- for 반복문은 원하는 횟수만큼 반복하고 싶을 때 사용
- ① 초기식을 비교합니다.
- ② 조건식을 비교합니다. 조건이 거짓이면 반복문을 종료합니다.
- ③ 문장을 실행합니다.
- ④ 종결식을 실행합니다.
- ⑤ 앞의 ② 단계로 이동합니다.

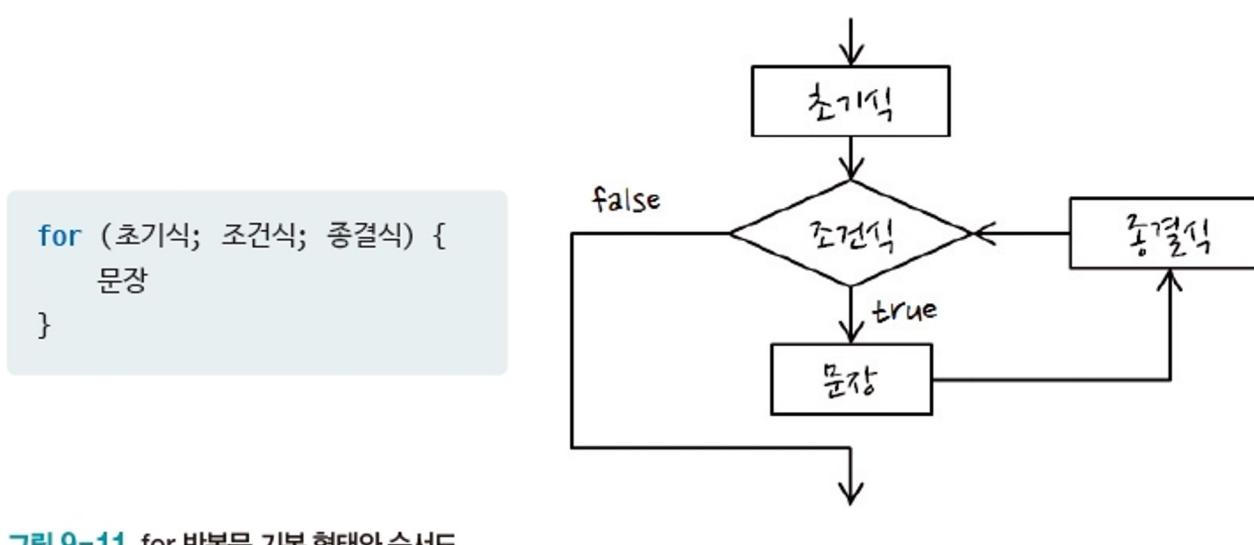


그림 9-11 for 반복문 기본 형태와 순서도

03 조건문과 반복문

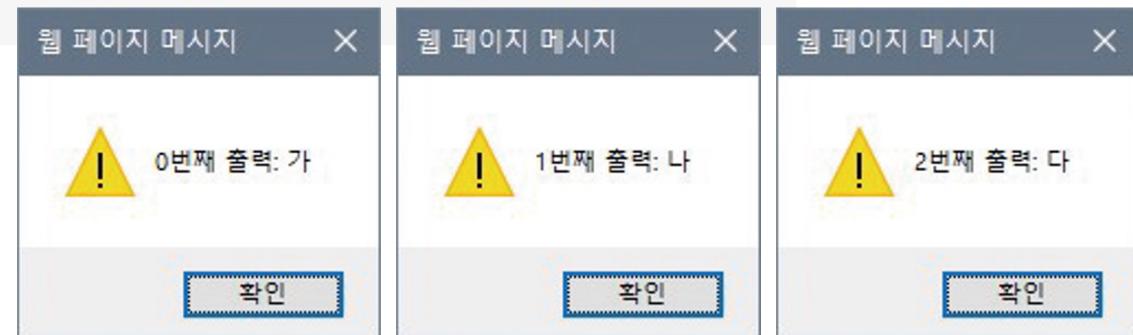
■ [기본 예제 9-10] for 반복문

[코드 데모](#)

코드 9-18 loop_for.html

```
<script>
  // 변수를 선언합니다.
  var array = ['가', '나', '다'];

  // 반복을 수행합니다.
  for (var i = 0; i < 3; i++) {
    // 출력합니다.
    alert(i + '번째 출력: ' + array[i]);
  }
</script>
```



03 조건문과 반복문

- [기본 예제 9-11] for 반복문을 사용한 0부터 100까지 합 계산

[코드 데모](#)

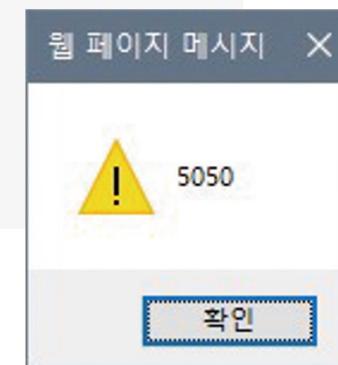
코드 9-20 loop_forPlus.html

```
<script>
    // 변수를 선언합니다.
    var output = 0;

    // 반복을 수행합니다.
    for (var i = 0; i <= 100; i++) {
        output += i;
    }

    // 출력합니다.
    alert(output);
</script>
```

100까지 더해야 하므로 \leq 연산자를 사용합니다.



04 함수

■ 선언과 호출, 실행 우선순위

▪ 선언과 호출

- 함수 - 코드 집합을 나타내는 자료형
- 익명 함수 생성 - 함수 이름을 입력하지 않고 만들기
- 선언적 함수 생성 - 함수 이름을 입력해서 만들기

표 9-11 함수 생성 방법

방법	표현
익명 함수	<code>function () { }</code>
선언적 함수	<code>function 함수() { }</code>

03 조건문과 반복문

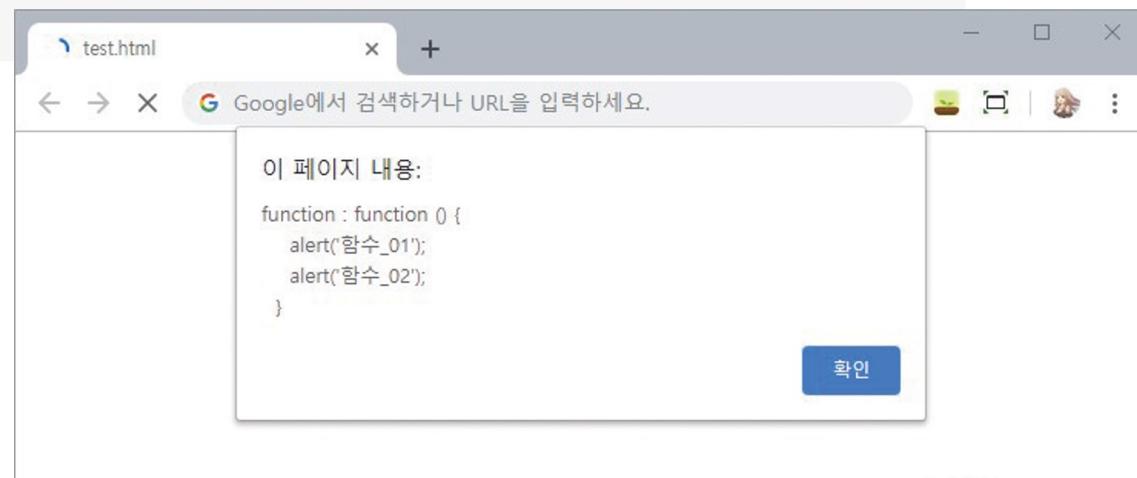
■ [기본 예제 9-12] 함수 선언

[코드 데모](#)

▪ 1. 익명 함수 선언하기

코드 9-21 function_noname.html

```
<script>
    // 함수를 선언합니다.
    var 함수 = function () {
        alert('함수_01');
        alert('함수_02');
    };
    // 출력합니다.
    alert(typeof (함수) + ' : ' + 함수);
</script>
```



03 조건문과 반복문

- [기본 예제 9-12] 함수 선언

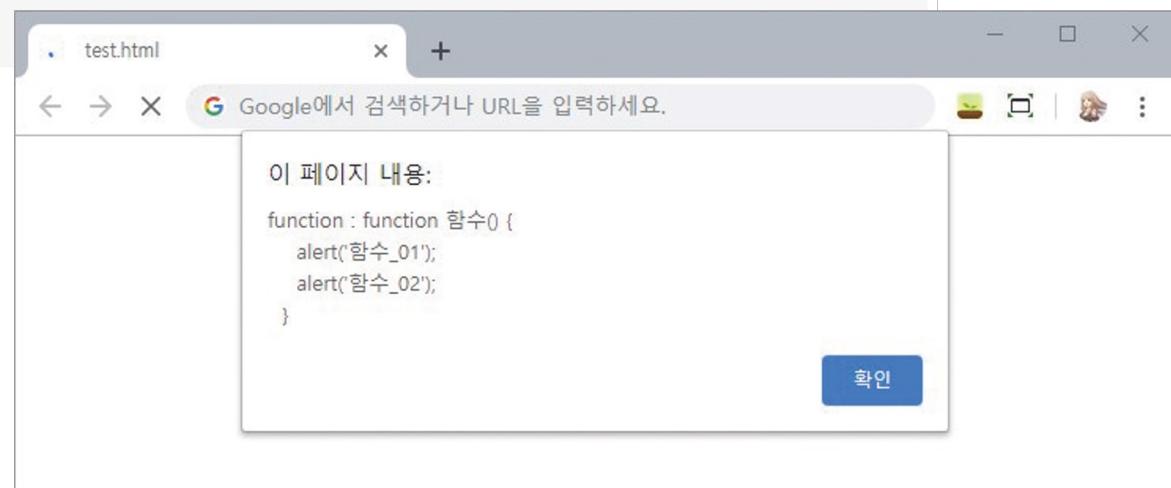
[코드 데모](#)

- 2. 선언적 함수 선언하기

코드 9-22 function_name.html

```
<script>
    // 함수를 선언합니다.
    function 함수() {
        alert('함수_01');
        alert('함수_02');
    }

    // 출력합니다.
    alert(typeof (함수) + ' : ' + 함수);
</script>
```



04 함수

■ 선언과 호출, 실행 우선순위

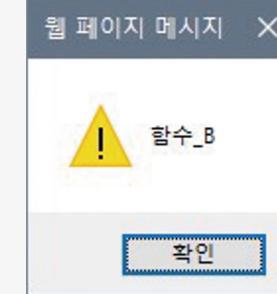
■ 실행 우선순위

- 가장 마지막에 입력된 값이 저장

코드 9-23 function_priorityBetweenNoname.html

```
<script>
    // 함수를 선언합니다.
    var 함수 = function () { alert('함수_A'); };
    var 함수 = function () { alert('함수_B'); };

    // 함수를 호출합니다.
    함수();
</script>
```



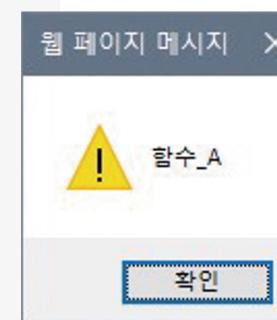
- ※주의 ※ 선언적 함수와 익명 함수를 함께 사용할 때

□ 모든 코드를 읽기 전에 선언적 함수를 먼저 읽음

코드 9-24 function_priority.html

```
<script>
    // 함수를 선언합니다.
    var 함수 = function () { alert('함수_A'); };
    function 함수() { alert('함수_B'); };

    // 함수를 호출합니다.
    함수();
</script>
```



04 함수

■ 매개변수와 반환 값

■ 매개변수

- 함수의 괄호 안에 집어넣어 함수 쪽에 추가적인 정보를 전달하는 것

■ 리턴 값

- 함수를 실행한 결과를 반환 값

```
function 함수 이름(매개변수, 매개변수, 매개변수) {  
    // 함수 코드  
    // 함수 코드  
    // 함수 코드  
    return 반환 값;  
}
```

그림 9-12 매개변수와 반환 값을 가지는 함수 형식

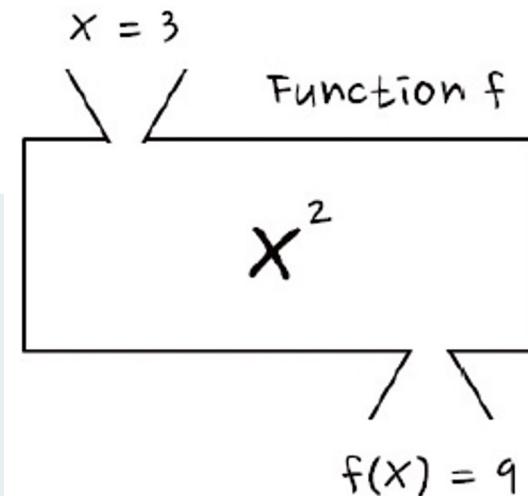


그림 9-13 함수

03 조건문과 반복문

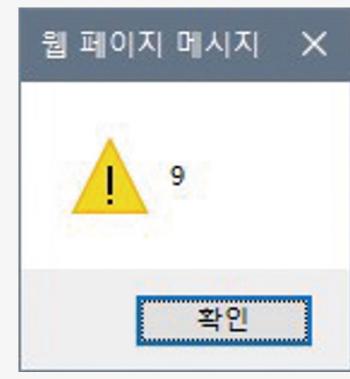
- [기본 예제 9-13] 매개변수와 반환 값이 있는 함수

[코드 데모](#)

코드 9-25 function_return.html

```
<script>
    // 함수를 선언합니다.
    function f(x) {
        return x * x;
    }

    // 함수를 호출합니다.
    alert(f(3));
</script>
```



04 함수

■ 콜백 함수

- 매개변수로 전달되는 함수

■ [기본 예제 9-14] 콜백 함수

[코드 데모](#)

코드 9-26 function_callback.html

```
<script>
    // 함수를 선언합니다.
    function callTenTimes(callback) {
        // 10회 반복합니다.
        for (var i = 0; i < 10; i++) {
            callback(); // 매개변수로 전달된 함수를 호출합니다.
        }
    }

    // 변수를 선언합니다.
    var callback = function () {
        alert('함수 호출');
    };

    callTenTimes(callback); // 함수를 호출합니다.
</script>
```

열 번 출력됩니다.

04 함수

- [기본 예제 9-14] 콜백 함수
- [코드 9-26]을 익명 함수로 실행하기

[코드 데모](#)

코드 9-27 reference_nonameCallback.html

```
<script>
    // 함수를 선언합니다.
    function callTenTimes(callback) {
        for (var i = 0; i < 10; i++) {
            callback();
        }
    }

    // 함수를 호출합니다.
    callTenTimes(function () {
        alert('함수 호출');
    });
</script>
```

05 객체

■ 객체 개요

- 객체는 자료형 여러 개를 한 번에 저장
- 배열은 요소에 접근할 때 인덱스를 사용하지만, 객체는 키를 사용

코드 9-28 object_create.html

```
<script>
  // 객체를 선언합니다.
  var product = {
    제품명: '7D 건조 망고',
    유형: '당절임',
    성분: '망고, 설탕, 메타중아황산나트륨, 치자황색소',
    원산지: '필리핀'
  };
</script>
```

키	속성
제품명	7D 건조 망고
유형	당절임
성분	망고, 설탕, 메타중아황산나트륨, 치자황색소
원산지	필리핀

그림 9-15 객체 생성 예

05 객체

■ 객체 개요

- 객체 뒤의 대괄호를 사용해 키를 입력하면 객체 속성에 접근

```
product['제품명'] → '7D 건조 망고'  
product['유형']   → '당절임'  
product['성분']   → '망고, 설탕, 메타중아황산나트륨, 치자황색소'  
product['원산지'] → '필리핀'
```

그림 9-16 [코드 9-28]에서 생성한 객체 속성에 접근하는 예 1

- 객체 뒤의 점(.)을 찍어 객체 속성에 접근

```
product.제품명 → '7D 건조 망고'  
product.유형   → '당절임'  
product.성분   → '망고, 설탕, 메타중아황산나트륨, 치자황색소'  
product.원산지 → '필리핀'
```

그림 9-17 [코드 9-28]에서 생성한 객체 속성에 접근하는 예 2

05 객체

■ 객체 개요

▪ for in 반복문

- 객체 요소를 하나씩 살펴볼 수 있음

```
for (var 키 in 객체) {  
    문장  
}
```

그림 9-18 for in 반복문 형태

코드 9-30 object_withForIn.html

```
<script>  
    // 객체를 선언합니다.  
    var product = {  
        제품명: '7D 건조 망고',  
        유형: '당절임',  
        성분: '망고, 설탕, 메타중아황산나트륨, 치자황색소',  
        원산지: '필리핀'  
    };  
  
    // 출력합니다.  
    for (var i in product) {  
        alert(i + ':' + product[i]);  
    }  
</script>
```

05 객체

■ 속성과 메서드

- 요소(element)
 - 배열에 있는 값 하나하나
- 속성(property)
 - 객체에 있는 값 하나하나

```
// 객체를 선언합니다.  
var object = {  
    number: 273,  
    string: 'rintiantta',  
    boolean: true,  
    array: [52, 273, 103, 32],  
    method: function () {  
    }  
};
```

그림 9-19 다양한 자료형의 객체 속성 예

05 객체

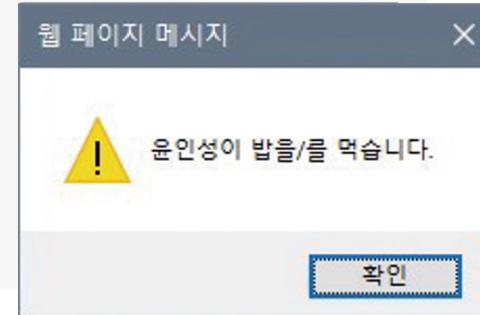
■ 속성과 메서드

- `this` 키워드
 - 객체에 있는 속성을 메서드에서 사용하고 싶을 때는 자신이 가진 속성임을 분명하게 표시해야 함

코드 9-31 object_this.html

```
<script>
    // 객체를 선언합니다.
    var person = {
        name: '윤인성',
        eat: function (food) {
            alert(this.name + '이 ' + food + '을/를 먹습니다.');
        }
    };

    // 메서드를 호출합니다.
    person.eat('밥');
</script>
```



01 문서 객체 모델의 기본 용어와 개념

■ 문서 객체 모델 기본 용어

- 문서 객체
 - 태그를 자바스크립트에서 사용할 수 있는 객체로 만든 것
- 문서 객체를 조작한다는 말은 결국 태그를 조작한다는 의미

코드 10-1 dom_basic.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>

        </script>
    </head>
    <body>
        <h1>Header</h1>
        <p>Lorem ipsum dolor amet</p>
    </body>
</html>
```

01 문서 객체 모델의 기본 용어와 개념

- 문서 객체 모델 기본 용어

- 요소 노드
 - <h1> 태그와 <script> 태그처럼 요소를 생성하는 노드
- 텍스트 노드 : 화면에 출력되는 문자열

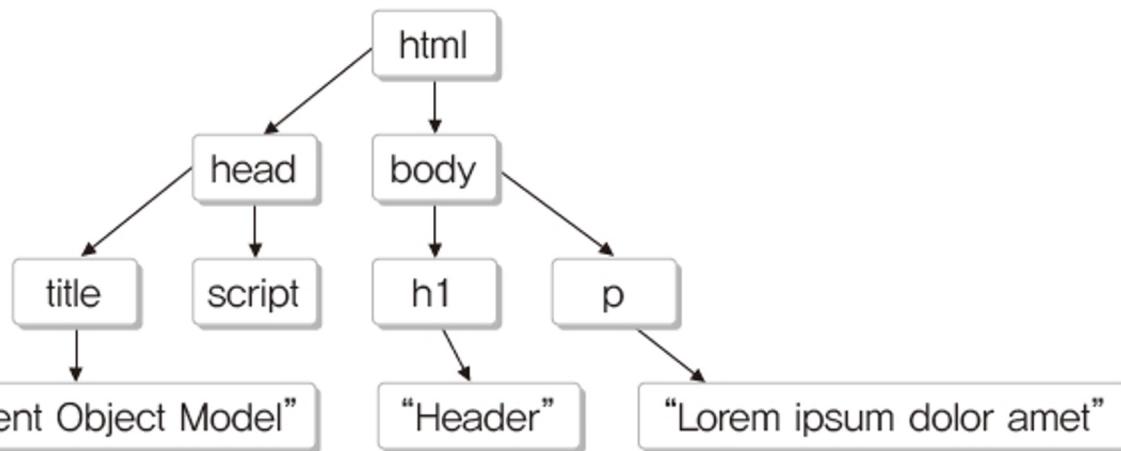


그림 10-1 노드

01 문서 객체 모델의 기본 용어와 개념

■ 문서 객체 모델 기본 용어

- 정적 생성

- 웹 페이지를 처음 실행할 때 HTML 태그로 적힌 문서 객체를 생성하는 것

- 동적 생성

- 웹 페이지를 실행 중에 자바스크립트를 사용해 문서 객체를 생성하는 것

- 문서 객체 모델 Document Object Model, DOM

- 웹 브라우저가 HTML 파일을 분석하고 출력하는 방식

01 문서 객체 모델의 기본 용어와 개념

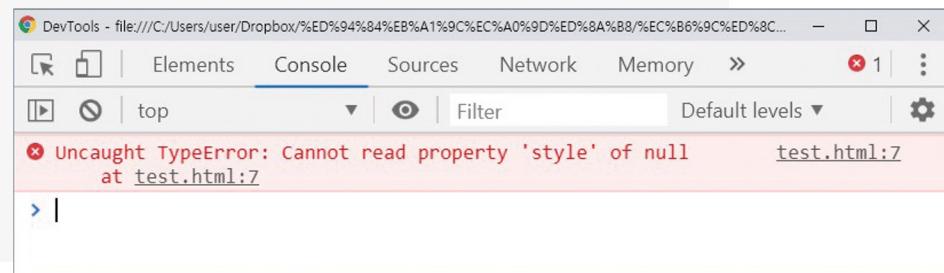
- 웹 페이지 실행 순서
- [기본 예제 10-1] 실행 순서에 따른 문서 객체 사용 오류
 - 1. 문서 객체 사용 오류

[코드 데모](#)

[온라인 에디터](#)

코드 10-2 dom_fault.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>
        // h1 태그의 배경 색상을 변경합니다.
        document.querySelector('h1').style.backgroundColor = 'red';
        // h2 태그의 글자 색상을 변경합니다.
        document.querySelector('h2').style.color = 'red';
    </script>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>
</body>
</html>
```

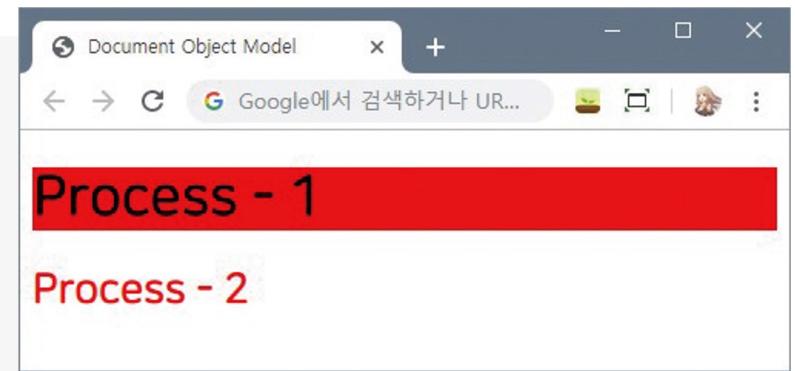


01 문서 객체 모델의 기본 용어와 개념

- 웹 페이지 실행 순서
- [기본 예제 10-1] 실행 순서에 따른 문서 객체 사용 오류 [코드 데모](#)
- 2. 해결 방법 1: script 태그 위치 옮기기

코드 10-3 dom_correct.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>
    <script>
        // h1 태그의 배경 색상을 변경합니다.
        document.querySelector('h1').style.backgroundColor = 'red';
        // h2 태그의 글자 색상을 변경합니다.
        document.querySelector('h2').style.color = 'red';
    </script>
</body>
</html>
```



01 문서 객체 모델의 기본 용어와 개념

- 웹 페이지 실행 순서
- [기본 예제 10-1] 실행 순서에 따른 문서 객체 사용 오류
 - 3. 해결 방법 2: 이벤트 활용하기

[코드 데모](#)

코드 10-4 dom_event.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Document Object Model</title>
    <script>
        window.onload = function () {
            // h1 태그의 배경 색상을 변경합니다.
            document.querySelector('h1').style.backgroundColor = 'red';
            // h2 태그의 글자 색상을 변경합니다.
            document.querySelector('h2').style.color = 'red';
        };
    </script>
</head>
<body>
    <h1>Process - 1</h1>
    <h2>Process - 2</h2>
</body>
</html>
```

02 문서 객체 선택

- ‘문서 객체를 선택한다’.
- 이미 존재하는 HTML 태그를 자바스크립트에서 문서 객체로 변환하는 것

표 10-1 문서 객체를 선택하는 메서드

구분	메서드	설명
1개 선택	document.getElementById(아이디)	아이디로 1개 선택
	document.querySelector(선택자)	선택자로 1개 선택
여러 개 선택	document.getElementsByName(이름)	name 속성으로 여러 개 선택
	document.getElementsByClassName(클래스)	class 속성으로 여러 개 선택
	document.querySelectorAll(선택자)	선택자로 여러 개 선택

02 문서 객체 선택

■ [기본 예제 10-2] 문서 객체 선택

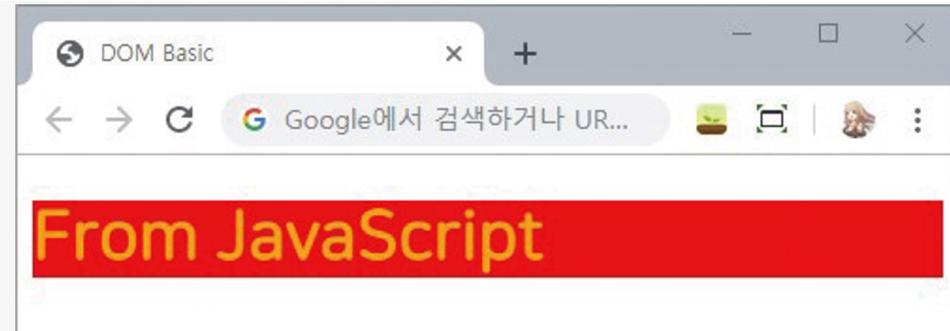
[코드 데모](#)

- 1. getElementById() 메서드를 사용해 문서 객체 1개 선택하기

코드 10-6 select_id.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var header = document.getElementById('header');

            // 문서 객체를 조작합니다.
            header.style.color = 'orange';
            header.style.background = 'red';
            header.innerHTML = 'From JavaScript';
        };
    </script>
</head>
<body>
    <h1 id="header">Header</h1>
</body>
</html>
```



태그 내부를 의미하는 속성입니다.

02 문서 객체 선택

■ [기본 예제 10-2] 문서 객체 선택

[코드 데모](#)

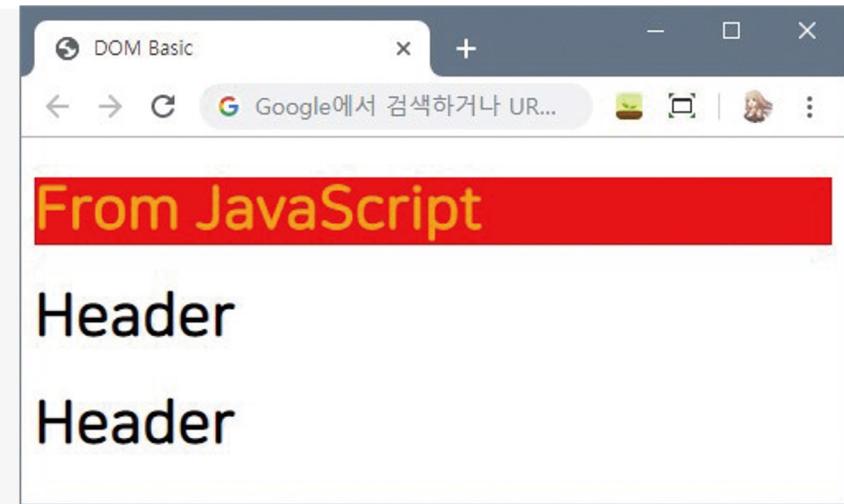
▪ 2. querySelector() 메서드를 사용해 문서 객체 1개 선택하기

코드 10-7 select_query.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var header = document.querySelector('h1');

            // 문서 객체를 조작합니다.
            header.style.color = 'orange';
            header.style.background = 'red';
            header.innerHTML = 'From JavaScript';

        };
    </script>
</head>
<body>
    <h1>Header</h1>
    <h1>Header</h1>
    <h1>Header</h1>
</body>
</html>
```



02 문서 객체 선택

■ [기본 예제 10-2] 문서 객체 선택

[코드 데모](#)

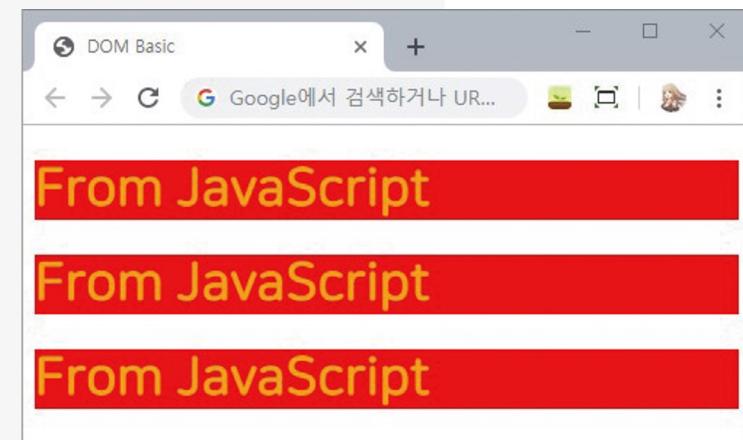
▪ 3. querySelectorAll() 메서드를 사용해 문서 객체 여러 개 선택하기

코드 10-8 select_all.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var headers = document.querySelectorAll('h1');

            for (var i = 0; i < headers.length; i++) {
                // 변수를 선언합니다.
                var header = headers[i];
                // 문서 객체를 조작합니다.
                header.style.color = 'orange';
                header.style.background = 'red';
                header.innerHTML = 'From JavaScript';
            }
        };
    </script>
</head>
```

```
<body>
    <h1>Header</h1>
    <h1>Header</h1>
    <h1>Header</h1>
</body>
</html>
```



03 문서 객체 조작

■ 글자 조작

▪ 글자 속성

- 문서 객체 내부에 있는 글자를 조작할 때 사용

표 10-2 글자 속성

속성	설명
textContent	문서 객체 내부 글자를 순수 텍스트 형식으로 가져오도록 변경
innerHTML	문서 객체 내부 글자의 HTML 태그를 반영해 가져오도록 변경

03 문서 객체 조작

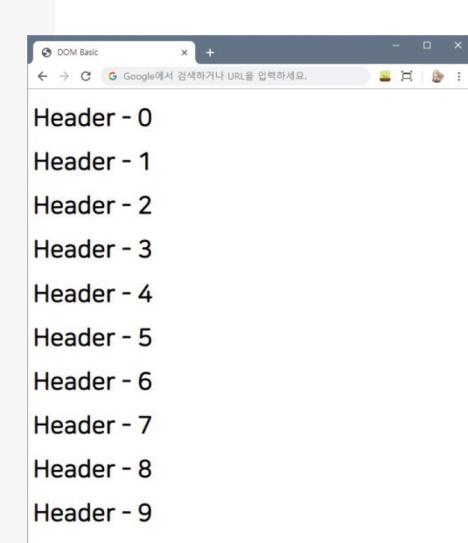
■ [기본 예제 10-3] 내부 글자 변경

[코드 데모](#)

코드 10-9 control_text.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 변수를 선언합니다.
            var output = '';
            for (var i = 0; i < 10; i++) {
                output += '<h1>Header - ' + i + '</h1>';
            }
            ❶ // 문서 객체 내부의 글자를 변경합니다.
            document.body.textContent = output;
            ❷ // document.body.innerHTML = output;
        };
    </script>
</head>
<body>

</body>
</html>
```



03 문서 객체 조작

■ 스타일 조작

- 자바스크립트로 CSS 속성 값을 추가·제거·변경 가능
- 1. 자바스크립트는 특수 문자 '-'을 식별자에 사용할 수 없으므로 오류 출력

```
① var header = document.getElementById('header');
    header.style.backgroundColor = 'red';
```

- 2. - 로 연결된 단어의 첫 글자를 대문자로 변경

```
② var header = document.getElementById('header');
    header.style.backgroundColor = 'red';
```

표 10-3 스타일 식별자 변환

스타일시트의 스타일 속성	자바스크립트의 스타일 식별자
background-image	backgroundImage
background-color	backgroundColor
box-sizing	boxSizing
list-style	listStyle

03 문서 객체 조작

■ [기본 예제 10-4] 스타일 조작(1)

[코드 데모](#)

코드 10-10 control_style.html

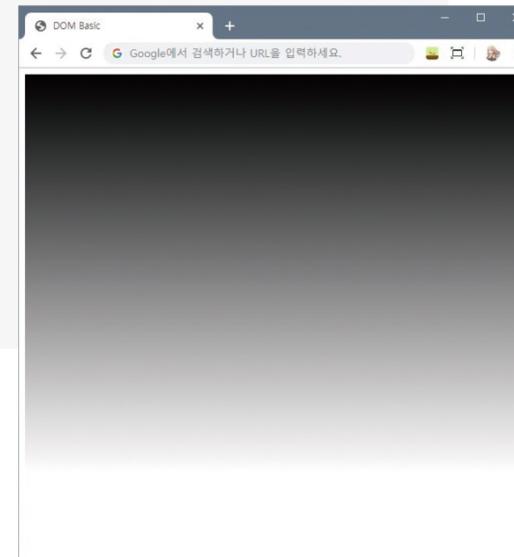
```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 추가합니다.
            var output = '';
            for (var i = 0; i < 256; i++) {
                output += '<div></div>';
            }
            document.body.innerHTML = output;
```

03 문서 객체 조작

■ [기본 예제 10-4] 스타일 조작(2)

[코드 데모](#)

```
// 문서 객체를 선택합니다.  
var divs = document.querySelectorAll('div');  
for (var i = 0; i < divs.length; i++) {  
    // 변수를 선언합니다.  
    var div = divs[i];  
  
    // 스타일을 적용합니다.  
    div.style.height = '2px';  
    div.style.background = 'rgb(' + i + ', ' + i + ', ' + i + ')';  
}  
};  
</script>  
</head>  
<body>  
  
</body>  
</html>
```



03 문서 객체 조작

■ 속성 조작

표 10-4 문서 객체의 속성 조작 메서드

메서드	설명
setAttribute(속성 이름, 속성 값)	속성 지정
getAttribute(속성 이름)	속성 추출

- setAttribute(), getAttribute() 메서드
 - 웹 표준에서 지정하지 않은 속성에 접근할 때 사용

```
<body>
  <div data-role="page">
    <div data-role="header"></div>
    <div data-role="content">

      </div>
      <div data-role="footer"></div>
    </div>
  </body>
```

그림 10-2 jQuery Mobile 코드

03 문서 객체 조작

- [기본 예제 10-5] 문서 객체 속성 조작
 - 1. img 태그 속성 조작하기

[코드 데모](#)

코드 10-11 control_attribute.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 변수를 선언합니다.
            var image = document.getElementById('image');

            // 속성을 변경합니다.
            image.src = 'http://placehold.it/300x200';
            image.width = 300;
            image.height = 200;
        };
    </script>
</head>
<body>
    <img id="image">
</body>
</html>
```



300 x 200

03 문서 객체 조작

- [기본 예제 10-5] 문서 객체 속성 조작
 - 2. body 태그 속성 조작하기

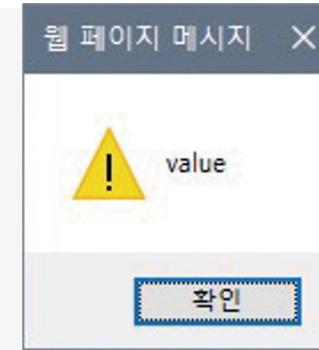
[코드 데모](#)

코드 10-12 control_body.html

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Basic</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 속성을 지정합니다.
            document.body.setAttribute('data-custom', 'value');

            // 속성을 추출합니다.
            var dataCustom = document.body.getAttribute('data-custom');
            alert(dataCustom);
        };
    </script>
</head>
<body>

</body>
</html>
```



```
<!DOCTYPE html>
▼<html>
  <head>...</head>
  <body data-custom="value"></body>
</html>
```

그림 10-3 검사로 속성 조작 확인

03 문서 객체 조작

- [기본 예제 10-6] 문서 객체를 사용한 시간 표시
 - 2. body 태그 속성 조작하기

[코드 데모](#)

코드 10-13 dom_clock.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Clock</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var clock = document.getElementById('clock');
            // 1초마다 함수를 실행합니다.
            setInterval(function () {
                var now = new Date();
                clock.innerHTML = now.toString();
            }, 1000);
        };
    </script>
</head>
<body>
    <h1 id="clock"></h1>
</body>
</html>
```



04 이벤트

■ 이벤트 용어 정리

- 이벤트 속성 (event property) - onload
- 이벤트 이름(event name) 또는 이벤트 타입(event type) - on을 제외한 load
- 이벤트 리스너(event listener) 또는 이벤트 핸들러(event handler)
 - 이벤트 속성에 넣는 함수

■ 기본적으로 지원하는 이벤트

- 마우스 이벤트
- 키보드 이벤트
- HTML 프레임 이벤트
- HTML 입력 양식 이벤트
- 사용자 인터페이스 이벤트
- 구조 변화 이벤트
- 터치 이벤트

```
onblur onfocus onfocusin onfocusout  
onload onresize onscroll onunload  
onclick ondblclick onmousedown  
onmouseup onmousemove onmouseover  
onmouseout onmouseenter  
onmouseleave onchange onselect  
onsubmit onkeydown onkeypress  
onkeyup onerror
```

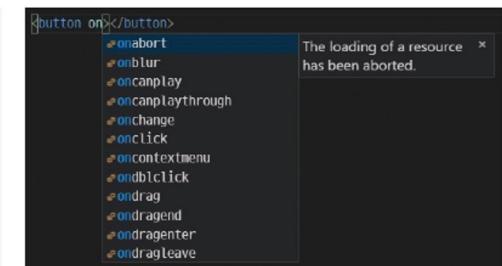


그림 10-4 이벤트 속성 종류와 통합 개발 환경의 속성 자동 완성

04 이벤트

■ 이벤트 연결

- 이벤트 모델(event model) - 문서 객체에 이벤트를 연결하는 방식

표 10-5 이벤트 모델 종류(이벤트 연결 방식)

구분	종류
DOM 레벨 0	<ul style="list-style-type: none">• 인라인 이벤트 모델• 고전 이벤트 모델
DOM 레벨 2	<ul style="list-style-type: none">• 마이크로소프트 인터넷 익스플로러 이벤트 모델• 표준 이벤트 모델

▪ 인라인 이벤트 모델

- HTML 태그 내부에 자바스크립트 코드를 넣어 이벤트를 연결하는 방식

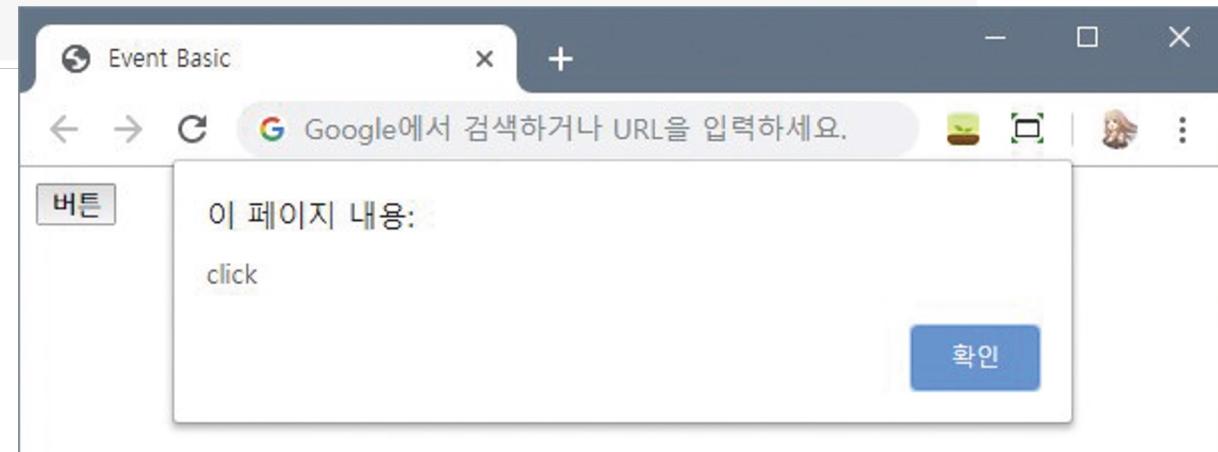
03 문서 객체 조작

- [기본 예제 10-7] 이벤트 연결
 - 1. 인라인 이벤트 모델 사용하기

[코드 데모](#)

코드 10-14 event_inline.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Basic</title>
</head>
<body>
    <button onclick="alert('click')">버튼</button>
</body>
</html>
```



03 문서 객체 조작

■ [기본 예제 10-7] 이벤트 연결

[코드 데모](#)

▪ 1. script 태그에 인라인 이벤트 모델 사용하기

코드 10-15 event_inlineWithScript.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Inline Event</title>
    <script>
        function buttonClick() {
            alert('click');
        }
    </script>
</head>
<body>
    <button onclick="buttonClick()">버튼</button>
</body>
</html>
```

04 이벤트

■ 이벤트 연결

- 고전 이벤트 모델 - 과거에 표준으로 정의되어 많이 사용하던 이벤트 모델

■ [기본 예제 10-8] 고전 이벤트 연결

[코드 데모](#)

코드 10-16 event_tradition.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Traditional Event</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var button = document.getElementById('button');

            // 이벤트를 연결합니다.
            button.onclick = function () {----- 이벤트 리스너
                alert('click');
            };
        };
    </script>
</head>
<body>
    <button id="button">버튼</button>
</body>
</html>
```

04 이벤트

- 이벤트 사용
 - 이벤트 객체를 사용하면 이벤트와 관련한 정보를 알아낼 수 있음
- [기본 예제 10-9] 이벤트 정보

[코드 데모](#)

코드 10-18 event_object.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Event Object</title>
    <script>
        window.onload = function (event) {
            alert(event);
        };
    </script>
</head>
<body>

</body>
</html>
```

이벤트 객체

04 이벤트

■ 이벤트 사용

- 기본 이벤트(default event) - 특정 태그가 가진 기본적인 이벤트

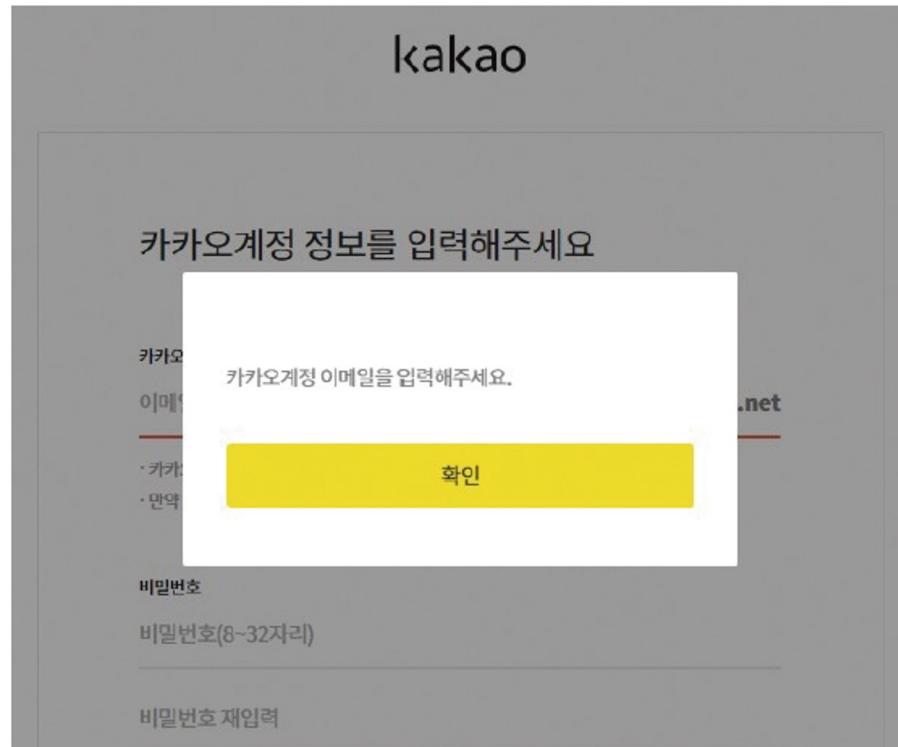


그림 10-5 기본 이벤트 제거를 사용한 유효성 검사

04 이벤트

■ [기본 예제 10-10] 기본 이벤트 제거

[코드 데모](#)

코드 10-19 event_default.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Traditional Event - Default Event</title>
    <script>
        // 이벤트를 연결합니다.
        window.onload = function () {
            // 문서 객체를 선택합니다.
            var button = document.getElementById('button');

            // 이벤트를 연결합니다.
            button.onclick = function () {
                // 기본 이벤트를 제거합니다.
                return false;
            };
        };
    </script>
</head>
<body>
    <a id="button" href="http://hanbit.co.kr">버튼</a>
</body>
</html>
```

