

# 게임 프로그래밍

Web

# Game Programming



# HTML5 문서

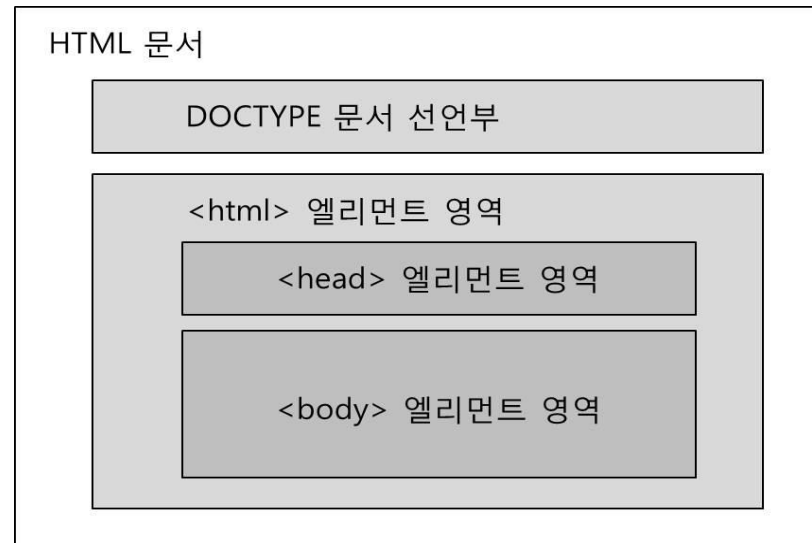
- HTML5 문서의 구조

- DOCTYPE

- 'Document Type'의 줄임말로 문서 유형을 지정
    - HTML5 태그는 아니지만 HTML5 문서라면 반드시 명세
    - 어떤 HTML5 태그보다도 먼저 문서의 맨 앞에 선언됨

- 문서 선언부를 제외하고는 <html> 시작 태그로 시작해서 문서 내용들이 명세되고 마지막 </html> 종료 태그로 끝남

- <html> 태그 안에는 <head>와 </head> 태그 쌍이 먼저 오고, <body>와 </body> 태그 쌍이 그 다음에 한 번씩만 올 수 있음



# HTML5 문서 예

- <!DOCTYPE html> 선언 : 웹 브라우저에게 HTML5로 작성되었음을 알림
- HTML5 문서 자체는 아스키(ASCII) 코드로 구성된 일반 텍스트 파일임
  - 문서 작성은 일반 텍스트 편집기를 이용하면 가능
  - 파일 확장자를 .html이나 .htm으로 저장
  - 웹 브라우저로 읽어 들이면 해석하고 변환한 HTML5 문서 결과를 확인할 수 있음

## [예제2-1] HTML5 문서 작성하기

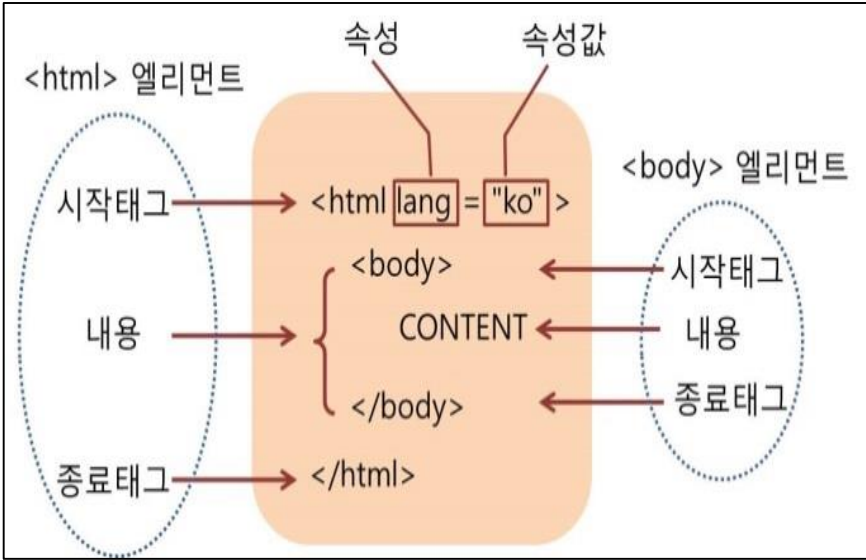
/ch02/html5-tag.html

<!DOCTYPE html>	<!-- 문서 유형 선언 -->
<html>	<!-- HTML5 문서의 시작 -->
<head>	<!-- 머리글(문서 설명 정보)의 시작 -->
<meta charset="utf-8">	<!-- 문서의 인코딩 설정 -->
<title>HTML5 문서 제목</title>	<!-- 문서 제목(브라우저 상단에 표시됨) -->
</head>	<!-- 머리글의 종료 -->
<body>	<!-- 문서 본문 시작 -->
<h1>HTML이란?</h1>	<!-- 문서 본문(글 제목) -->
HyperText Markup Language의 약어	<!-- 문서 본문(글 내용) -->
</body>	<!-- 문서 본문 종료 -->
</html>	<!-- HTML5 문서의 끝 -->



# HTML5의 기본 요소

요소	의미	코드 예
태그(tag)	'<'와 '>'로 둘러싸인 문자열 시작태그(< >)와 종료태그(</ >)로 구성 문서 표현 방식을 지시	<title>웹문서내용</title>
내용(content)	태그로 둘러싸인 문자열	<title>웹문서내용</title>
엘리먼트(element)	태그와 내용을 포함한 전체 문자열 HTML 문서의 기본 구성 단위 (상위)엘리먼트 안에 (하위)엘리먼트가 계층적으로 포함될 수 있음	<title>웹문서내용<title>
속성(attribute)	엘리먼트의 상세한 표현(기능) 설정 사항을 지시 시작 태그 안에 사용	<title color="red"></title>
속성값(value)	속성값(' '또는 " "로 감싸야 함)	<title color="red"></title>



# 공간 분할 태그

- 공간 분할 태그

- 태그 자체의 특별한 의미나 기능이 없이 단지 페이지 공간 분할의 단위로 사용
- 같은 공간에 위치할 엘리먼트들을 담는 컨테이너(container)의 역할
- 여러 하위 태그들을 특정 공간에 그룹화하여 공간 배치를 쉽게 하도록 함
- 예) <div>와 <span> 태그

- <div>와 <span> 태그

- 그룹화된 공간을 하나의 단위로 하여 스타일을 적용하면 웹 문서 전체의 레이아웃을 쉽게 구성할 수 있음
- <div> 태그
  - 공간 분할을 위해 가장 많이 사용되는 태그
  - 태그들을 묶어 그룹화하는 컨테이너로서 전체 문서를 구조화함
  - 스타일시트와 결합하면 그룹에 속한 태그들을 한번에 모두 같은 스타일로 적용 가능
- <span> 태그
  - 내용 일부를 다른 엘리먼트들과 분리하고자 할 때 사용
  - 내용을 강조하거나 각기 다른 스타일을 적용할 수 있음



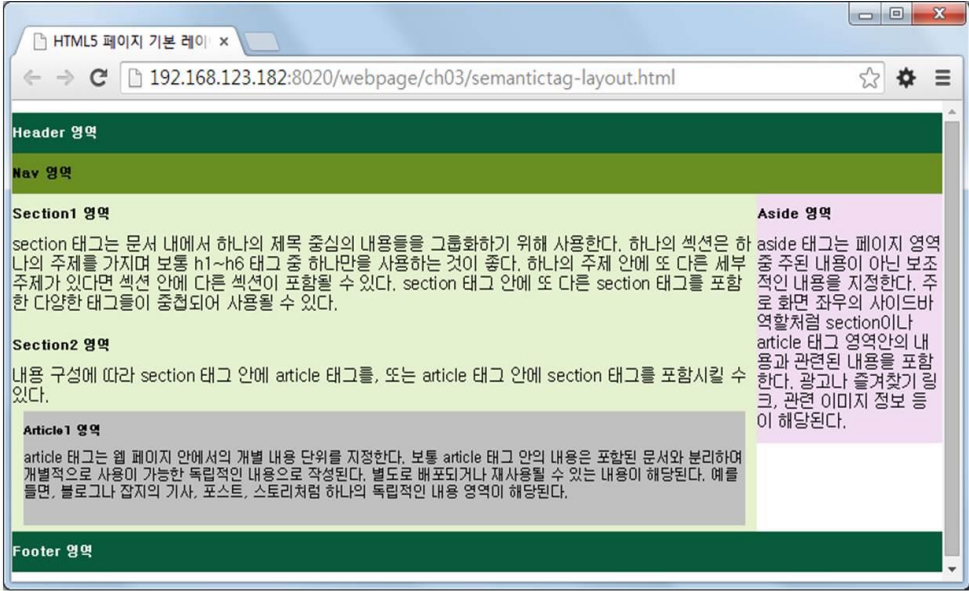
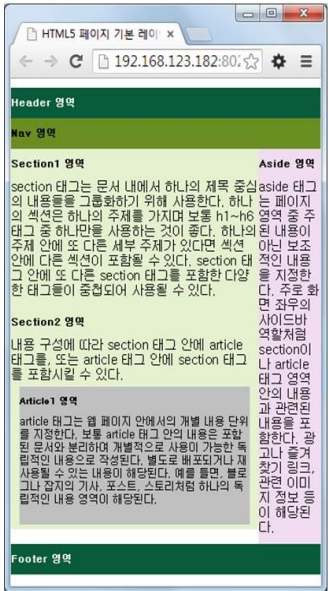
# 공간 분할 형식

- 보통 블록 형식 태그들을 하나로 묶어 감쌀 때는 <div> 태그를, 내부에 인라인 형식 태그들을 감쌀 때는 <span> 태그를 사용

형식	태그	기능
블록 형식	<div>	블록(block) 형식으로 공간을 분할하는 영역을 정의 영역 공간이 매번 줄바꿈이 되어 새로운 행에 생성되는 방식 하위에 다른 블록 형식 태그나 인라인 형식 태그를 포함할 수 있음
	일반태그	<div>, <h1>~<h6>, <p>, <table>, <ul>, <ol>, <li>, <form>
인라인 형식	<span>	인라인(inline) 형식으로 공간을 분할하는 영역을 정의 영역 공간이 입력 내용의 길이만큼만 차지하면서 줄바꿈 없이 같은 줄에 계속 추가되는 방식 반드시 블록 형식 태그의 하위에 포함되어야 하며 하위에 블록 형식 태그를 포함할 수 없음
	일반태그	<span>, <img>, <a>, <input>, <i>, <b>, <strong>, <small>, <sub>, <sup>, <ins>, <del> (단, HTML5에서는 <a> 태그 내부에 <h1>~<h6>,<p>,<ul>,<ol> 등의 블록 형식 태그를 포함할 수 있음)

# 시맨틱 태그

- HTML5 : 웹 페이지 구성 방식의 변화도 포함
  - 문서에 정보의 의미도 함께 고려하도록 함
  - <div> 태그를 세분화하여 각 태그에 의미를 부여함으로써 페이지를 의미 단위로 분할
  - 시맨틱 태그(semantic tag)
    - 태그만으로 문서를 쉽게 이해할 수 있도록 의미를 부여한 새로운 태그
    - 어떤 표현이나 행위를 지시하기 보다는 웹 문서의 내용이 담고 있는 의미를 명확히 전달할 목적으로 만들어진 태그



# Semantic tag

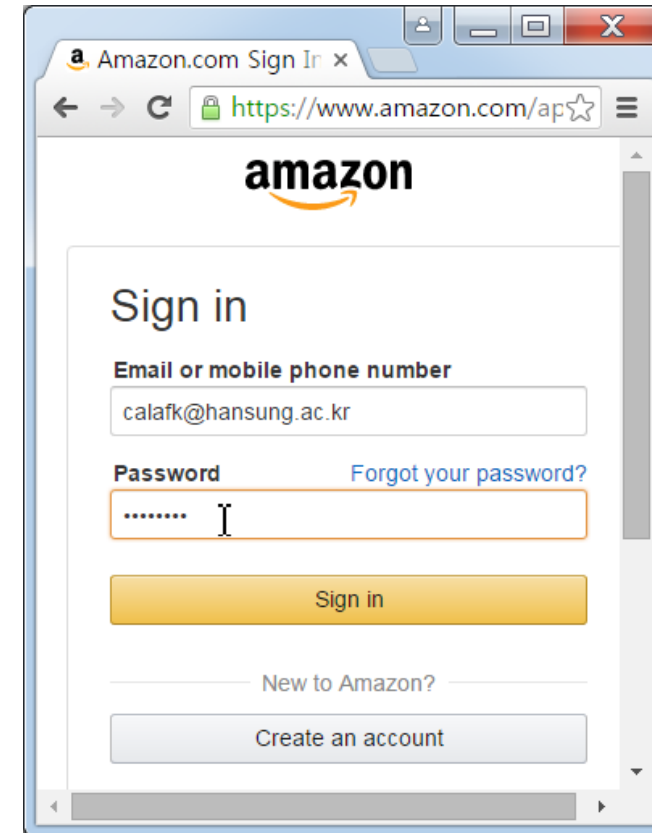
시맨틱 태그	의미	기능
<header>	머리말 영역	페이지(혹은 내용) 상단의 머리말을 지정 페이지 제목, 소개 글이나 로고 이미지 등이 포함 <body>, <section>, <article> 태그 안에도 여러 번 포함 가능
<nav>	메뉴 영역	페이지 이동을 위한 메뉴 영역을 지정 이전 페이지, 다음 페이지, 특정 페이지나 사이트에 대한 연결 등 네비게이션 요소로 구성 <header>, <aside>, <footer> 태그 안에도 여러 번 포함 가능
<aside>	보조 영역	페이지의 좌,우 측면 공간 같은 보조 영역을 지정 광고나 즐겨찾기 링크, 관련 이미지 정보 등을 제공 <section>, <article> 태그 안의 내용과 간접적으로 관련된 내용을 포함
<section>	형식적 구분 영역	제목을 갖는 연관된 내용 영역을 지정 하나의 제목(<h1>~<h6> 태그)을 중심으로 내용들을 그룹화 <section> 태그 안에 또 다른 <section> 태그를 포함한 다양한 태그들이 중첩되어 사용
<article>	내용적 구분 영역	독립적인 개별 내용 영역을 지정 블로그나 댓글, 신문, 잡지의 기사 등을 제공 별도로 배포되거나 재사용 가능한 내용으로 그룹화 <section> 태그와 <article> 태그 사이의 중첩 가능
<footer>	꼬리말 영역	페이지(혹은 내용) 하단의 꼬리말을 지정 작성자과 작성 날짜, 저작권 등 웹 페이지 관련 추가 정보들을 포함





## 웹 폼

- 웹 폼
  - ▣ 웹 페이지에서 사용자 입력을 받는 폼
  - ▣ 로그인, 등록, 검색, 예약, 쇼핑 등
- 폼 요소
  - ▣ 폼을 만드는 다양한 태그
  - ▣ `<input>`, `<textarea>`, `<select>` 등



# 폼 요소의 종류

폼 요소	설명
<input type="text">	한 줄 텍스트 입력 창
<input type="password">	암호 입력을 위한 한 줄 텍스트 입력 창
<input type="button">	단순 버튼
<input type="submit">	웹 서버로 폼 데이터를 전송시키는 버튼
<input type="reset">	입력된 폼 데이터를 초기화시키는 버튼
<input type="image">	이미지 버튼
<input type="checkbox radio">	체크박스과 라디오버튼
<select>	드롭다운 리스트를 가진 콤보박스
<input type="month week date time datetime-local">	년, 월, 일, 시간 등의 시간 정보 입력 창
<input type="number range">	스핀 버튼과 슬라이드바로 편리한 숫자 입력 창
<input type="color">	색 입력을 쉽게 하는 컬러 다이얼로그
<input type="email url tel search">	이메일, URL, 전화번호, 검색키워드 등 형식 검사 기능을 가진 텍스트 입력 창
<input type="file">	로컬 컴퓨터의 파일을 선택하는 폼 요소
<button type="button reset submit">	단순 버튼, reset, submit 버튼
<textarea>	여러 줄의 텍스트 입력 창



## 스타일시트 기초

- 스타일(style)
  - 문서를 멋지게 장식하거나 정해진 양식에 맞게 내용을 다듬는 역할
  - 문서 작성 프로그램에서 작성하는 모든 문서는 스타일을 적용함
- 스타일시트(stylesheet)
  - 문서의 스타일을 정의
  - 화면에 출력하기 위한 외형적인 모양과 형식에 관한 스타일 규칙들의 모임
  - 스타일시트에 의해 웹 문서가 브라우저에 어떻게 보여지는지가 결정됨
- 예) 똑같은 HTML5 웹 페이지에 다른 디자인의 스타일시트를 적용한 결과



# CSS(Cascading Style Sheet)

- CSS(Cascading Style Sheet)

- HTML 문서에 적용하는 표준 스타일시트
- W3C 웹 컨소시엄에서 개발한 스타일시트 언어로 1996년 CSS1, 1998년 CSS2, 2007년 CSS2.1 현재는 CSS3가 개발 중임
- 기존의 스타일시트 개념을 웹에 적용한 것
- 마크업 문서의 스타일을 정의하는 속성(property)들을 모아 놓은 언어이기 때문에 HTML 문서뿐만 아니라 XML 문서에도 적용 가능
- 화면 구성이나 표현 형식 등 디자인 부분은 HTML 대신에 CSS에 위임함으로써 문서 구조와 문서 표현을 자연스럽게 분리할 수 있다.

- CSS 사용의 이점

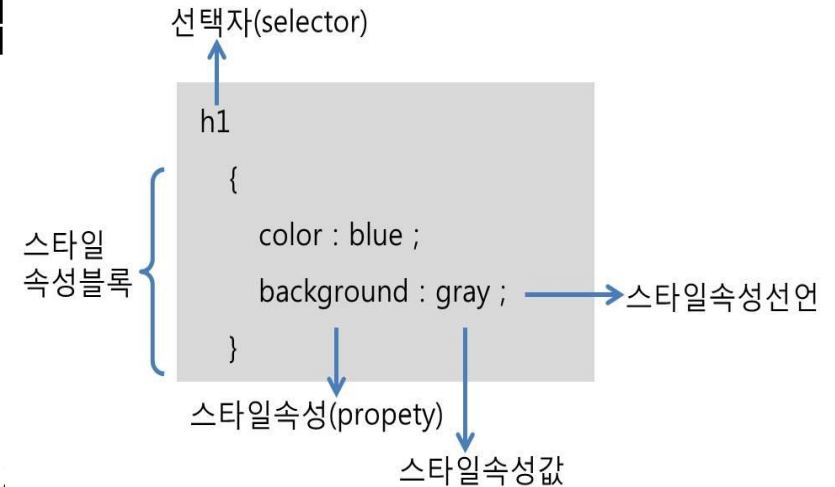
- 확장성 : 표현을 더욱 다양하게 확장하거나 표현 기능의 변경이 가능
- 편의성 : 훨씬 간편하게 레이아웃 등의 스타일을 구성
- 재사용성 : 독립된 스타일 모듈을 작성, 여러 HTML 문서에 공통으로 활용
- 생산성 : 역할 분담에 따른 전문화, 모듈 단위의 협업과 생산성 향상이 가능



# CSS3 스타일 선언 방법

## ● CSS3 스타일 선언 형식

- 선택자(selector)
  - 스타일을 적용할 대상을 지정
  - 특정 태그 영역만 원하는 스타일 또는 특별한 기능(제이쿼리 사용)을
- 스타일 속성(property) 블록
  - 선택자에 의해 선택된 영역에 적용할 색상, 크기 등의 구체적인 스타일을 명세
  - 스타일 속성 블록은 중괄호({ })로 둘러싸며 하나 이상의 스타일 속성 선언을 포함
- 스타일 속성 이름
  - 선택자에 대해 어떤 스타일을 부여할지 적용할 실질적인 기능에 해당하는 이름을 지정(기능을 쉽게 유추할 수 있는 단어로 구성)
- 스타일 속성 값
  - 스타일 속성의 종류에 따라 설정 값의 유형과 범위가 제한됨
  - 의미를 표현하는 키워드나 숫자를 단위와 함께 명세



# 선택자 유형

종류	형식	내용
전체 선택자	*	모든 태그에 스타일 적용
태그 선택자	태그명	지정한 이름의 태그에만 스타일 적용
클래스 선택자	.클래스명	지정한 클래스 속성을 갖는 태그에만 스타일 적용
아이디 선택자	#아이디명	지정한 아이디 속성을 갖는 태그에만 스타일 적용
계층 선택자	선택자 태그명 선택자 > 태그명 선택자 + 태그명 선택자 ~ 태그명	태그들의 계층 구조에서 특정 위치의 태그에만 스타일 적용 자손 선택자 자식 선택자 형제(근접후행) 선택자 형제(후행) 선택자
상태 선택자	선택자:상태조건	지정된 특정 상태 조건을 충족하는 태그에만 스타일 적용



# 확장 선택자 유형

- 상태 선택자(state selector)
  - 콜론(:) 뒤에 엘리먼트를 선택할 수 있는 특별한 상태 즉, 조건을 나타내는 예약어를 명세
  - 단독으로 사용되기 보다는 다른 선택자와 함께 짝을 이루어 선택자를 꾸며주는 역할을
  - 가상 클래스(pseudo class) 선택자라고도 함
- 상태 선택자

종류	상태	기능
링크선택자	선택자:link	href 속성을 가진(방문 전) <a> 태그를 선택
	선택자:visited	방문했던 링크를 가진 <a> 태그를 선택
반응선택자	선택자:hover	마우스 포인터를 올려놓은(누르지 않고) 태그를 선택
	선택자:active	마우스 포인터로 클릭한 태그를 선택
부정선택자	선택자:not	선택 대상을 반대로 적용



# 스타일시트 선언 방식

- CSS3 스타일시트를 선언하는 4가지 방식
  - 스타일시트의 내용이 어디에 위치해 있는지에 따라 결정
- 스타일시트 선언 방식

선언 방식	내용	특성
내부 방식	HTML5 문서 내부에 스타일시트를 삽입하는 형식(임베디드 형식)	<style> 태그 안에 스타일시트를 명세
외부 방식	HTML5 문서 외부에 독립된 스타일시트 파일을 작성하여 연결하는 형식(링크 형식)	<link> 태그를 통해 외부 CSS3 파일을 연결
인라인 방식	HTML5 문서 안의 각 엘리먼트에 속성으로 스타일을 끼어 넣는 형식	HTML5 태그 안에 style 속성을 설정
임포트 방식	스타일시트 안에서 또 다른 스타일시트 파일을 포함시키는 방식	스타일시트 안에 @import 규칙을 사용하여 외부 CSS3 파일을 포함



# 스타일 적용의 우선 순위

- 4가지 스타일시트 선언 방식을 함께 사용하여 스타일 충돌(conflict)이 발생할 경우  
⇒ 우선 순위와 상속의 의해서 결정  
⇒ 우선 순위는 가장 제한적으로, 가장 하위 수준에서, 가장 나중에 정의할수록 높음
- 여러 스타일시트 방식이 결합 적용된 경우의 적용 우선 순위

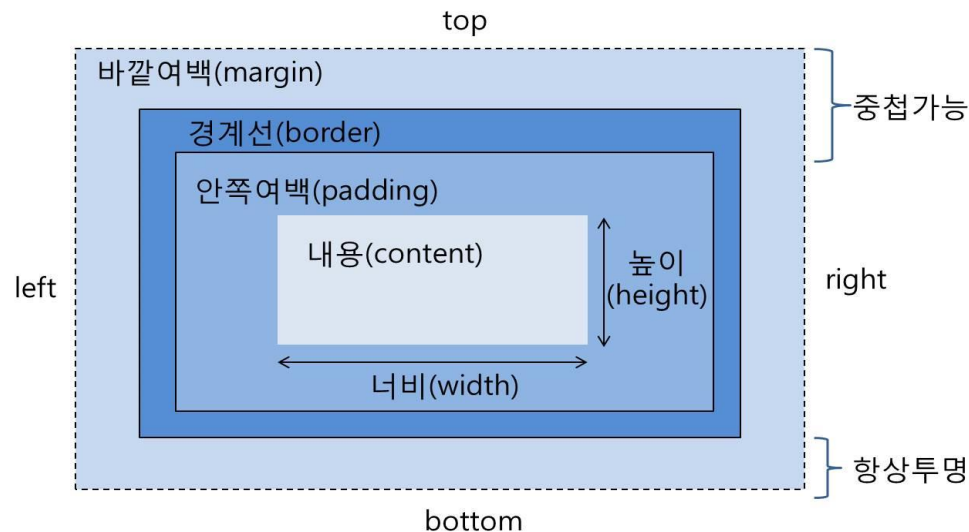


- 같은 선언 방식 안에서의 상속의 개념을 고려한 규칙
  - 상위 엘리먼트에서 정의한 스타일은 하위 엘리먼트에도 상속될 수 있음
  - 상속받은 스타일과 하위 엘리먼트에서 정의한 스타일이 겹쳐서 충돌이 발생할 경우, 상속받은 스타일보다 직접 정의한 스타일이 우선
  - 만약, 직접 정의한 스타일 속성도 없고 상속받은 속성도 없다면 기본값으로 설정
- 직접 스타일을 정의하는 경우의 선택자 유형에 따른 적용 우선 순위



# 박스 모델 구조

- 박스 모델(box model)
  - 다양한 스타일 속성들이 적용되는 박스 모양의 스타일 관련 공간 개념
  - CSS3를 사용할 때 가장 기본이 되는 개념
  - 모든 엘리먼트들은 표시될 때 줄의 일부, 줄 전체 또는 여러 줄에 걸친 사각형 모양의 공간을 차지 -> 웹 브라우저가 모든 엘리먼트들을 작은 박스로 간주, 공간 배치하기 때문
- 박스 모델의 구조

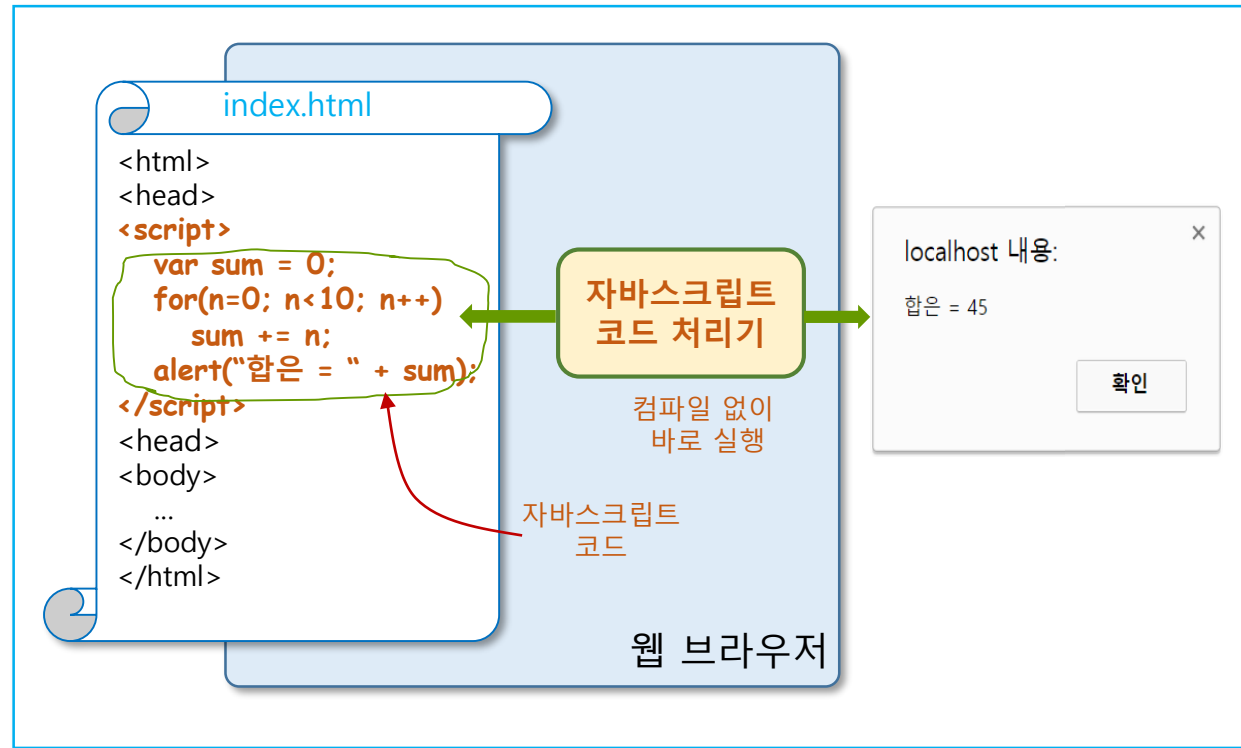


- 스타일 측면에서 문서는 하나 이상의 엘리먼트 박스들로 구성됨
- 안쪽 박스 모델 영역은 바깥쪽 상위 박스 모델의 내용 영역 안으로만 제한됨



# 자바스크립트 언어

- Javascript
  - ▣ 1995년 넷스케이프 개발
  - ▣ Netscape Navigator 2.0 브라우저에 최초 탑재
  - ▣ 웹 프로그래밍 개념 창시
- 특징
  - ▣ HTML 문서에 내장
    - 조각 소스 코드
  - ▣ 스크립트 언어
    - 인터프리터 실행
    - 컴파일 필요 없음
  - ▣ 단순
    - C언어 구조 차용
    - 배우기 쉬움



# 자바스크립트 언어의 발전

## ❖ 세계에서 가장 오해를 많이 받는 프로그래밍 언어

- 부수적인 프로그래밍 언어로 취급

## ❖ 풍부한 경험을 제공하는 인터넷 애플리케이션(RIA)

- 구글 지도의 등장
  - 자바스크립트만으로 만든 웹 페이지가 데스크톱에서 사용하는 애플리케이션의 형태를 가졌으며 기존의 지도에 비해 강력한 지도 제공
  - RIA : 풍부한 경험을 선사하는 웹 애플리케이션



## ❖ Node.js

- 2008년 9월 : 구글은 크롬 웹 브라우저의 베타 버전을 발표
- 2009년 1월 이후 : 자바스크립트를 웹 브라우저가 아닌 곳에서도 사용할 수 있는 표준안 제시
- CommonJS 표준 발표 이후 : CommonJS 표준과 V8 자바스크립트 엔진을 기반으로 Node.js를 개발



# 자바스크립트로 할 수 있는 일

## ❖ 웹 클라이언트 애플리케이션 개발

- 웹 브라우저에서 실행되는 웹 클라이언트 애플리케이션 개발이 목적
- 웹 브라우저에서 실행할 수 있는 유일한 프로그래밍 언어

## ❖ 웹 서버 개발

- 기존에 웹 개발은 두 가지 이상의 프로그래밍 언어가 필요했음
  - 웹 클라이언트, 웹 서버를 다른 언어로 개발
- Node.js가 등장하면서 웹 서버도 자바스크립트로 개발 가능



- 웹 페이지를 출력하지 않아도 웹 프로토콜(HTTP 또는 HTTPS 등)을 활용하면 웹 서버로 칭함.  
페이팔 결제 시스템에도 Node.js를 활용
- Node.js는 웹 개발과 관련해서 간단한 모듈들만 제공해서 데이터 처리와 예외 처리 등이 복잡하나, 빠르다는 장점이 있음



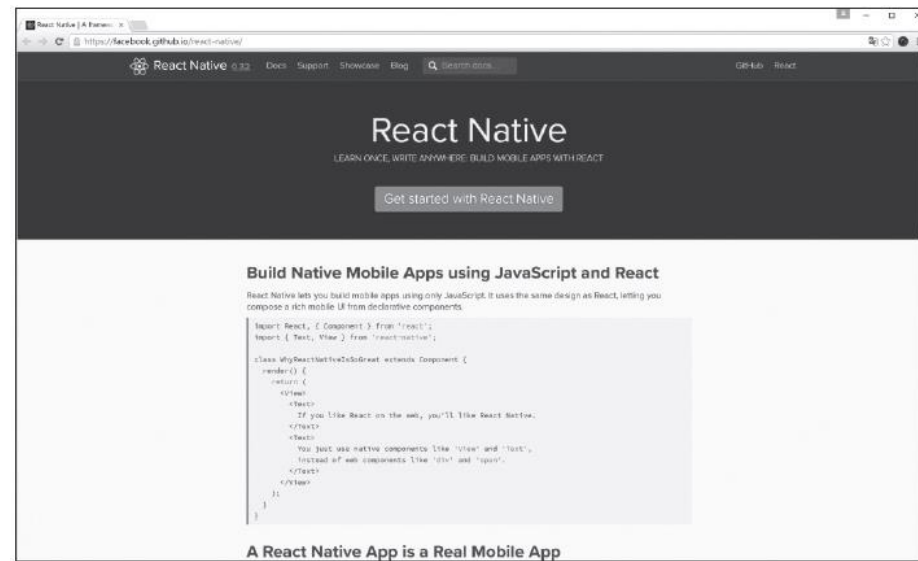
## ❖ 모바일 애플리케이션 개발

### ■ 네이티브 애플리케이션

- 스마트폰에서 인식할 수 있는 프로그래밍 언어(자바, 스위프트 등)로 만든 애플리케이션
- 기업에서 애플리케이션을 만들 경우 2가지 언어로 만들기에 비용이 2배가 됨
- 자바스크립트를 사용하면 1개의 애플리케이션만 개발해도 스마트폰 동작 가능

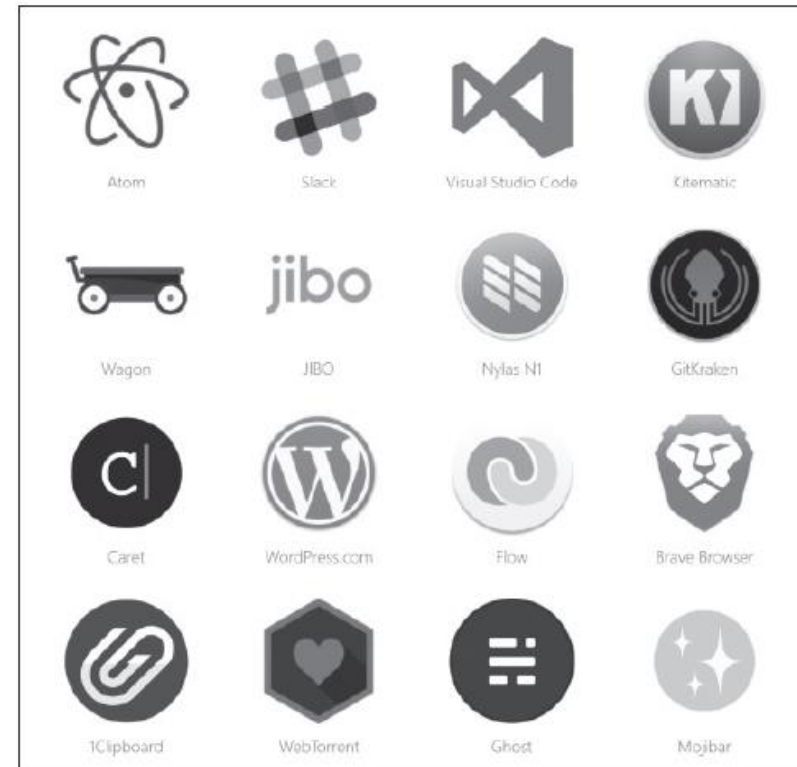
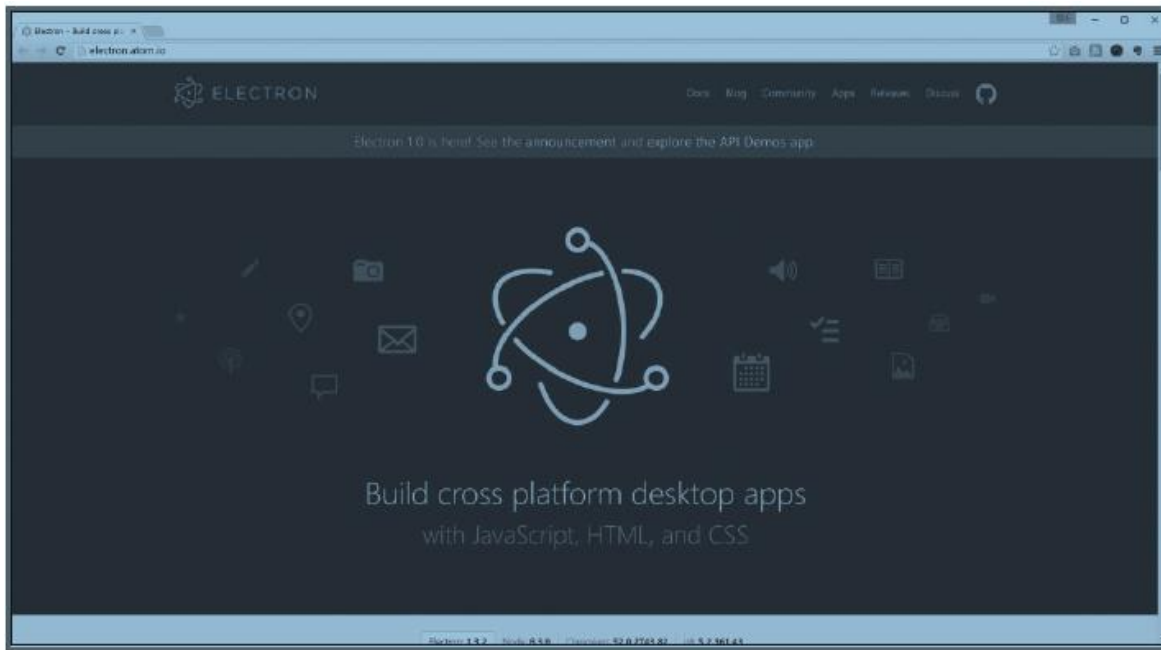
### ■ 페이스북의 React Native

- 자바스크립트로 네이티브 애플리케이션을 개발(내부적으로 프로그래밍 언어를 변환)



## ❖ 데스크톱 애플리케이션 개발

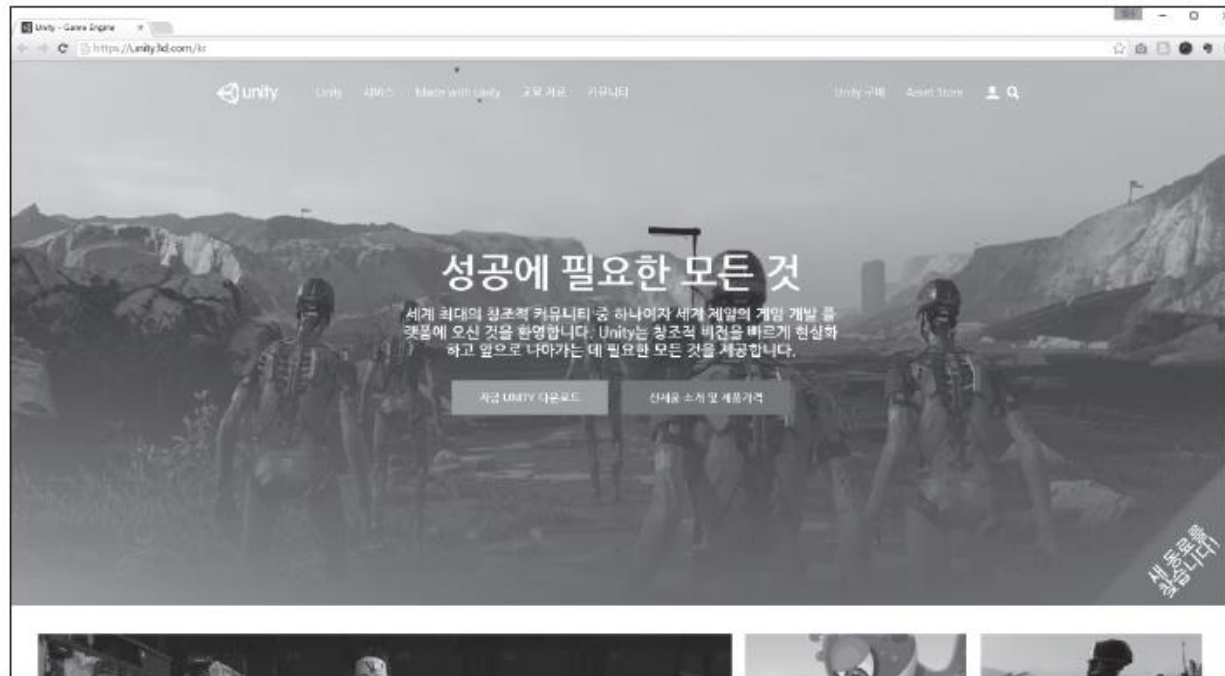
- 일렉트론Electron 모듈 : 자바스크립트로 개발 전용 텍스트 에디터를 만들어 배포, 본격적으로 데스크톱 애플리케이션 개발





## ❖ 게임 개발

- 원래 게임은 서버와 클라이언트 모두 C++로 제작(속도가 빠름)
- 스마트폰이 활성화 되면서 '한 번에 여러 스마트폰 운영체제에서 실행할 수 있는 애플리케이션을 개발하는 것'이 경제적으로 이득이 됨
- 유니티Unity 게임 엔진 등장 : 자바스크립트 기반



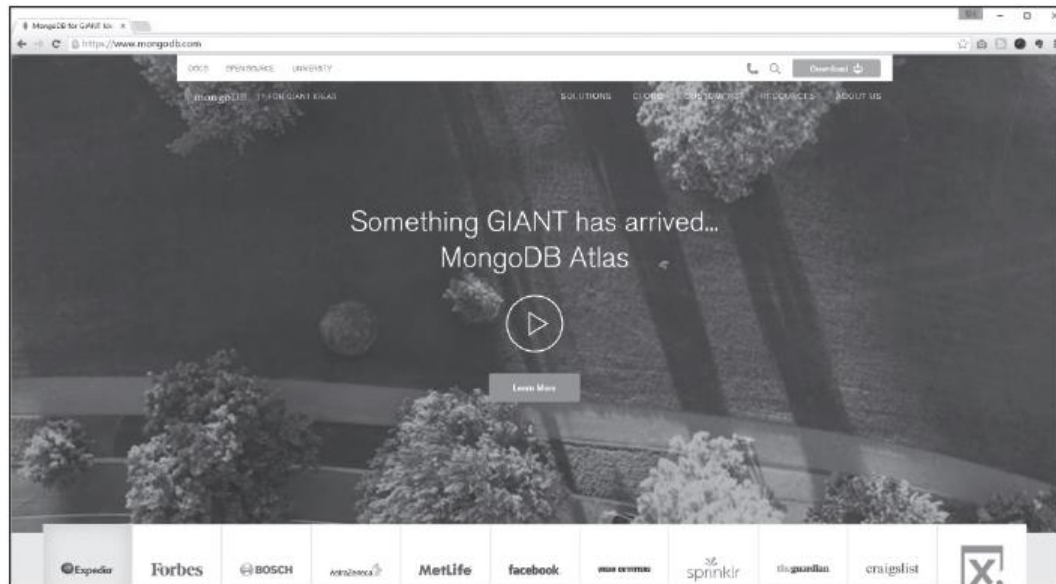
## ❖ 데이터베이스 관리

### ■ 데이터베이스

- 데이터를 저장할 때 사용하는 프로그램 (보통 SQL 프로그래밍 언어 사용)
- NoSQL : 기존의 SQL은 복잡하고 무거워 사용하기 쉬운 데이터베이스 등장

### ■ 2016년 9월 기준으로 주사용 데이터베이스 엔진

- Oracle, MySQL, Microsoft SQL Server, MongoDB, PostgreSQL
- MongoDB : 데이터베이스를 관리할 때 자바스크립트를 활용하는 NoSQL



# 웹 페이지에서 자바스크립트의 역할

- 사용자의 입력 및 계산
  - ▣ 마우스와 키보드 입력은 오직 자바스크립트로만 가능
  - ▣ 계산 기능
- 웹 페이지 내용 및 모양의 동적 제어
  - ▣ HTML 태그의 속성, 콘텐츠, CSS 프로퍼티 값 동적 변경
- 브라우저 제어
  - ▣ 브라우저 윈도우 크기와 모양 제어
  - ▣ 새 윈도우 열기/닫기
  - ▣ 다른 웹 사이트 접속
  - ▣ 히스토리 제어
- 웹 서버와의 통신
- 웹 애플리케이션 작성
  - ▣ 캔버스 그래픽, 로컬/세션 스토리지 저장, 위치정보서비스 등



# 자바스크립트 코드의 위치

## 자바스크립트 코드 작성이 가능한 위치

1. HTML 태그의 이벤트 리스너 속성에 작성
2. `<script></script>` 태그에 작성
3. 자바스크립트 파일에 작성
4. URL 부분에 작성

```
<!DOCTYPE html>
<html>
<head>
<title>이벤트 리스너 속성에 자바스크립트 코드</title>
</head>
<body>
<h3>마우스 올려 보세요</h3>
<hr>

</body>
</html>
```

이벤트 리스너  
속성

this는 현재 img 태그를  
가리키는 자바스크립트 키워드

자바스크립트  
코드

onclick 이벤트  
리스너 속성

자바스크립트 코드  
(이미지를 banana.png로 교체)

```

```



# Game Programming

```
<!DOCTYPE html>
<html>
<head> <title>script 태그에 자바스크립트 작성</title>
<script>
function over(obj) {
  obj.src="media/banana.png";
}
function out(obj) {
  obj.src="media/apple.png";
}
</script>
</head>
<body>
<h3>마우스 올려 보세요</h3>
<hr>

</body>
</html>
```

obj는 전달받은  
img 태그를 가리킴

this는 현재 img 태그를  
가리키는 자바스크립트키워드

```
/* 자바스크립트 파일 lib.js */
function over(obj) {
  obj.src="media/banana.png";
}
function out(obj) {
  obj.src="media/apple.png";
}
```

```
<!DOCTYPE html>
<html>
<head> <title>외부 파일에 자바스크립트 작성</title>
<script src="lib.js">
</script>
</head>
<body>
<h3>마우스 올려 보세요</h3>
<hr>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head> <title>URL에 자바스크립트 작성</title>
</head>
<body>
<h3>링크의 href에 자바스크립트 작성</h3>
<hr>
<a href="javascript:alert('클릭하셨어요?')">
  클릭해보세요</a>
</body>
</html>
```

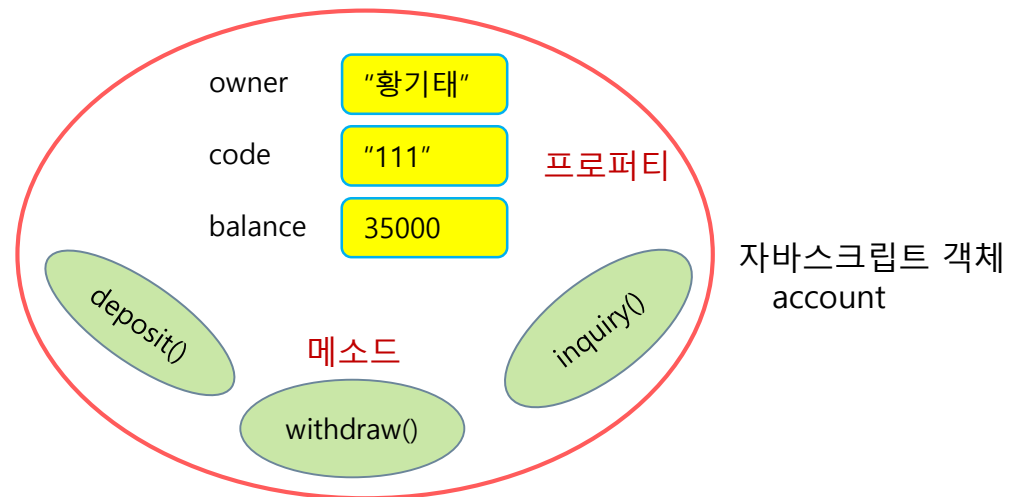
# 자바스크립트 객체 종류

- 자바스크립트는 객체 기반 언어
  - ▣ 자바스크립트는 객체 지향 언어 아님
- 자바스크립트 객체의 유형
  1. 코어 객체
    - 자바스크립트 언어가 실행되는 어디서나 사용 가능한 기본 객체
    - 기본 객체로 표준 객체
    - Array, Date, String, Math 타입 등
    - 웹 페이지 자바스크립트 코드에서 혹은 서버에서 사용 가능
  2. HTML DOM 객체
    - HTML 문서에 작성된 각 HTML 태그들을 객체화한 것들
    - HTML 문서의 내용과 모양을 제어하기 위한 목적
    - W3C의 표준 객체
  3. 브라우저 객체
    - 자바스크립트로 브라우저를 제어하기 위해 제공되는 객체
    - BOM(Browser Object Model)에 따르는 객체들
    - 비표준 객체



# 사용자 객체 만들기

- 사용자가 새로운 타입의 객체 작성 가능 : 3 가지 방법
  - ▣ 1. 직접 객체 만들기
    - new Object() 이용
    - 리터럴 표기법 이용
  - ▣ 2. 객체의 틀(프로토타입)을 만들고 객체 생성하기
- 샘플
  - ▣ 은행 계좌를 표현하는 account 객체



# new Object()로 객체 만들기

## □ 과정

- ▣ 1. new Object()로 빈 객체 생성
- ▣ 2. 빈 객체에 프로퍼티 추가
  - 새로운 프로퍼티 추가(프로퍼티 이름과 초기값 지정)
- ▣ 3. 빈 객체에 메소드 추가
  - 메소드로 사용할 함수 미리 작성
  - 새 메소드 추가(메소드 이름에 함수 지정)

```
var account = new Object();  
account.owner = "황기태"; // 계좌 주인 프로퍼티 생성 및 초기화  
account.code = "111"; // 코드 프로퍼티 생성 및 초기화  
account.balance = 35000; // 잔액 프로퍼티 생성 및 초기화  
account.inquiry = inquiry; // 메소드 작성  
account.deposit = deposit; // 메소드 작성  
account.withdraw = withdraw; // 메소드 작성
```





# 리터럴 표기법으로 만들기

## □ 과정

- ▣ 중괄호를 이용하여 객체의 프로퍼티와 메소드 지정

```
var account = {  
  // 프로퍼티 생성 및 초기화  
  owner : "황기태",    // 계좌 주인 프로퍼티 추가  
  code : "111",        // 계좌 코드 프로퍼티 추가  
  balance : 35000,     // 잔액 프로퍼티 추가  
  
  // 메소드 작성  
  inquiry : function () { return this.balance; }, // 잔금 조회  
  deposit : function(money) { this.balance += money; }, // 저금. money 만큼 저금  
  withdraw : function (money) { // 예금 인출, money는 인출하고자 하는 액수  
    // money가 balance보다 작다고 가정  
    this.balance -= money;  
    return money;  
  }  
};
```



# 프로토타입

- 프로토타입(prototype)이란?
  - ▣ 객체의 모양을 가진 틀
  - ▣ 붕어빵은 객체이고, 붕어빵을 찍어내는 틀은 프로토타입
  - ▣ C++, Java에서는 프로토타입을 클래스라고 부름
  - ▣ Array, Date, String : 자바스크립트에서 제공하는 프로토타입
  - ▣ 객체 생성시 'new 프로토타입' 이용
    - `var week = new Array(7);` // Array는 프로토타입임
    - `var hello = new String( "hello" );` // String은 프로토타입임



## 프로토타입으로 객체 만들기

- ▣ 프로토타입은 함수로 만든다
  - 프로토타입 함수를 생성자 함수라고도 함

```
// 프로토타입 Student 작성
function Student(name, score) {
  this.univ = "한국대학"; // this.univ을 이용하여 univ 프로퍼티 작성
  this.name = name; // this.name을 이용하여 name 프로퍼티 작성
  this.score = score; // this.score를 이용하여 score 프로퍼티 작성
  this.getGrade = function () { // getGrade() 메소드 작성
    if(this.score > 80) return "A";
    else if(this.score > 60) return "B";
    else return "F";
  }
}
```

- ▣ new 연산자로 객체를 생성한다

```
var kitae = new Student("황기태", 75); // Student 객체 생성
var jaemoon = new Student("이재문", 93); // Student 객체 생성
document.write(kitae.univ + ", " + kitae.name + "의 학점은 " + kitae.getGrade() + "<br>");
document.write(jaemoon.univ + ", " + jaemoon.name + "의 학점은 " + jaemoon.getGrade() + "<br>")
```



## Reference

명품 HTML5+CSS3+Javascript 웹프로그래밍, 황기태, 생능출판  
jQuery Mobile·Cordova를 활용한 모바일 웹+웹앱+하이브리드앱 입문, 박성진, 생능출판  
자바스크립트 프로그래밍 입문, 유인성, 한빛아카데미

BOM

<https://opentutorials.org/course/1375/6628>

[https://www.w3schools.com/js/js\\_window.asp](https://www.w3schools.com/js/js_window.asp)

<https://heecheolman.tistory.com/35>

<https://idlecomputer.tistory.com/42>

<http://www.deadfire.net/jscript/jscript22.html>

<https://ko.javascript.info/browser-environment>

코딩을 편하게 해주는 템플릿 제작 툴

<https://aperitif.io/>

10분만에 개인 웹사이트 만들기

<https://brunch.co.kr/@llun/14>



## Reference

### Document

[https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)

[http://tcpschool.com/javascript/js\\_dom\\_document](http://tcpschool.com/javascript/js_dom_document)

<https://opentutorials.org/course/1375/6740>

<https://m.blog.naver.com/magnking/220972680805>

<https://iamawebdeveloper.tistory.com/category/javascript/Document%20%EA%B0%9D%EC%B2%B4>

<https://im-developer.tistory.com/107>

<https://poiemaweb.com/js-dom>

<http://jeonghwan-kim.github.io/2018/01/25/before-jquery.html>

