

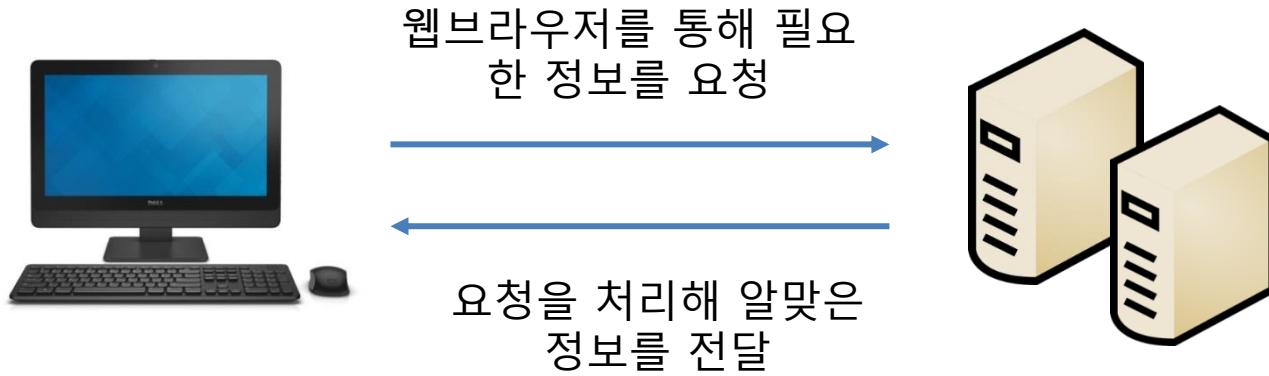
1강. 자바스크립트 기초 문법



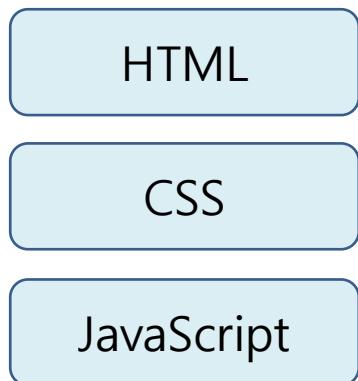
웹 프로그래밍이란?

- 프로그래밍이란?
 - 컴퓨터 프로그램을 만드는 일
 - 프로그램이란? 컴퓨터에게 일을 시키는 명령의 집합
- 웹 프로그래밍이란?
 - 웹에 관련된 프로그램을 만드는 일
 - 웹브라우저와 관련된 프로그램을 작성하는 일
- 웹 프로그래밍의 구분
 - **프런트엔드(Front-end)** 개발
 - > 서버에서 받아 온 정보를 웹 브라우저에 어떻게 보여줄 것인지를 프로그래밍
 - **백엔드(Back-end)** 개발
 - > 서버에서 사용자의 요청을 처리하거나 데이터를 관리하는 프로그래밍

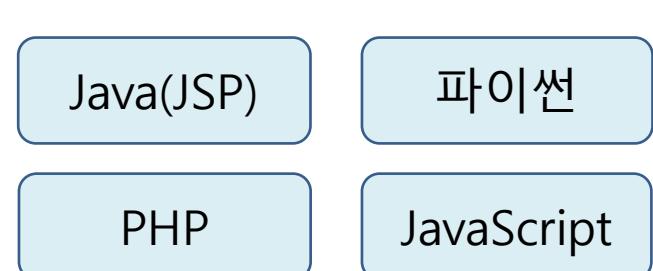
클라이언트와 서버



클라이언트(Client)측 언어



서버(Server)측 언어



자바스크립트로 할 수 있는 일

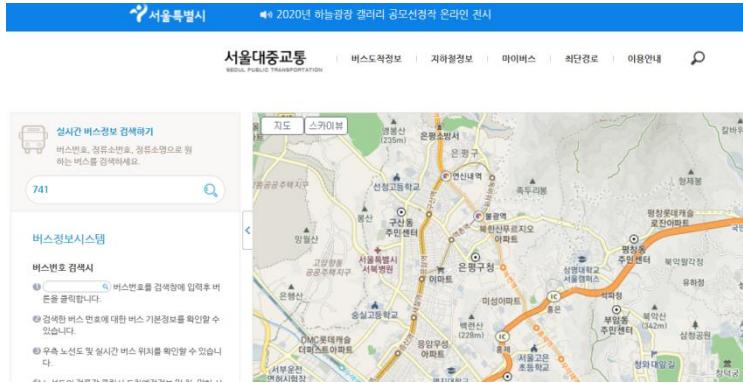
❖ JavaScript(자바스크립트)



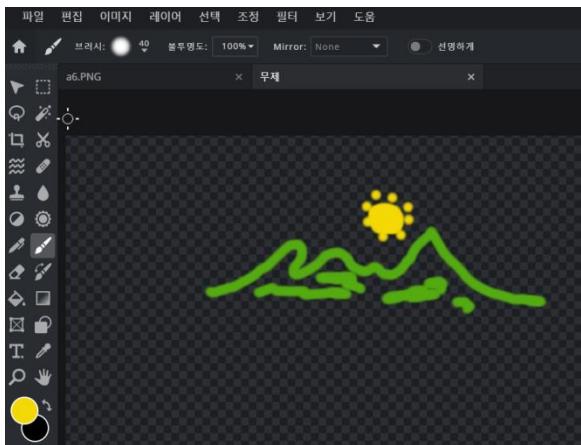
- HTML에 소속된 프로그램으로 동적인 웹사이트 제작(웹 브라우저에서만 작동)

※ 자바스크립트의 탄생 – 1995년 네스케이프(netscape)의 브랜던 아이크가 창시

자바스크립트로 할 수 있는 일



서울 대중교통 지도웹



픽사(Pixlr) 그림판

- 웹 애플리케이션 제작
 - 지도관련, 웹 오피스 등
- 게임제작
- 서버용 프로그램 제작

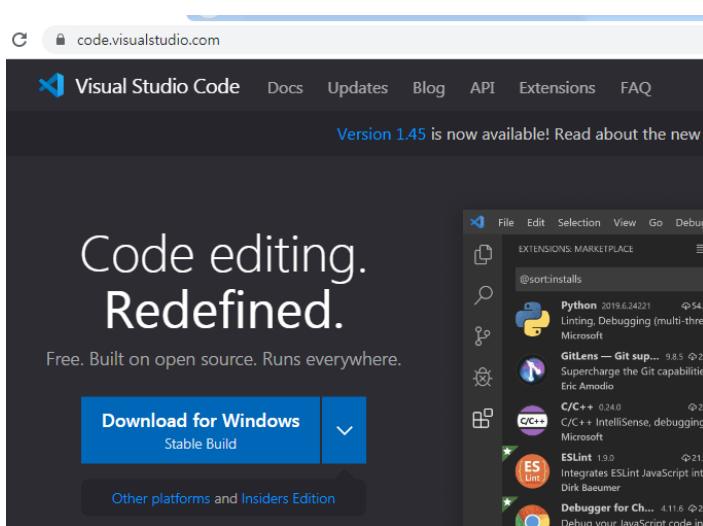
node.js (자바스크립트 실행환경)로 가능

구글 크롬의 해석엔진(V8)을 확장한 프로그램

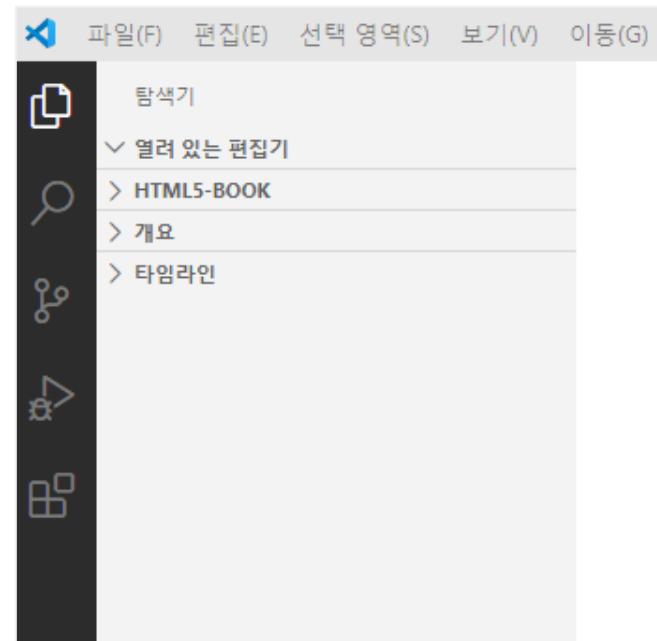
편집기 – 비주얼 스튜디오 코드

비주얼 스튜디오 코드

- 비주얼 스튜디오 코드 다운로드 -> 설치



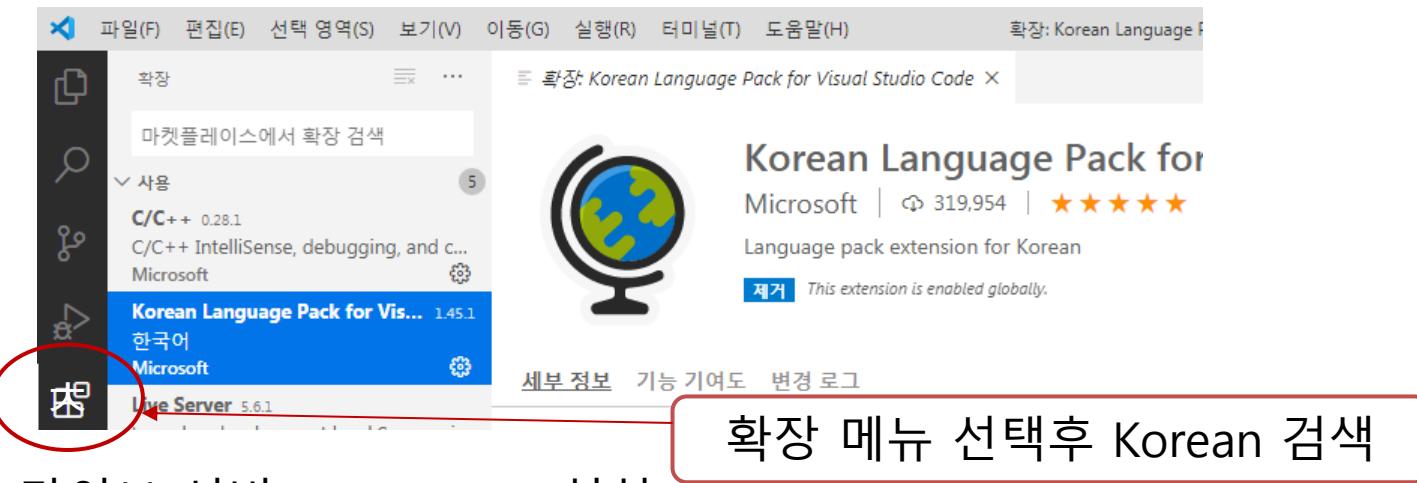
다운로드



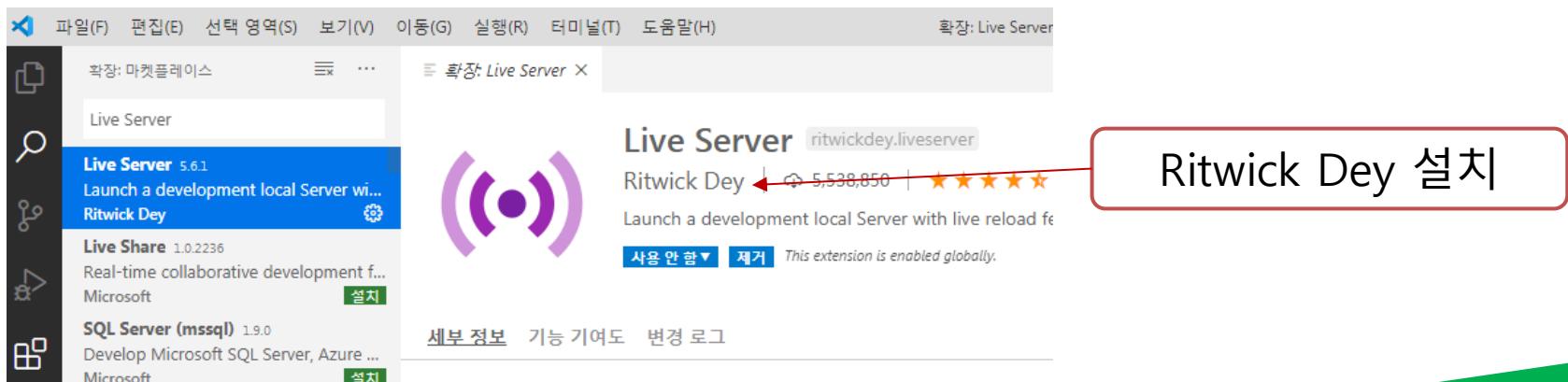
VS code 화면

비주얼 스튜디오 코드 환경 설정

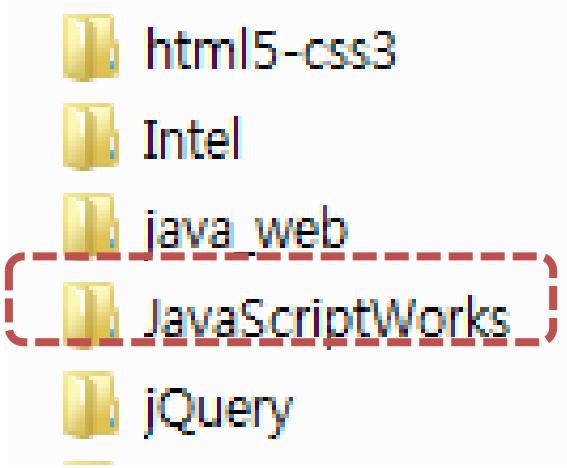
- 한국어 팩 설치



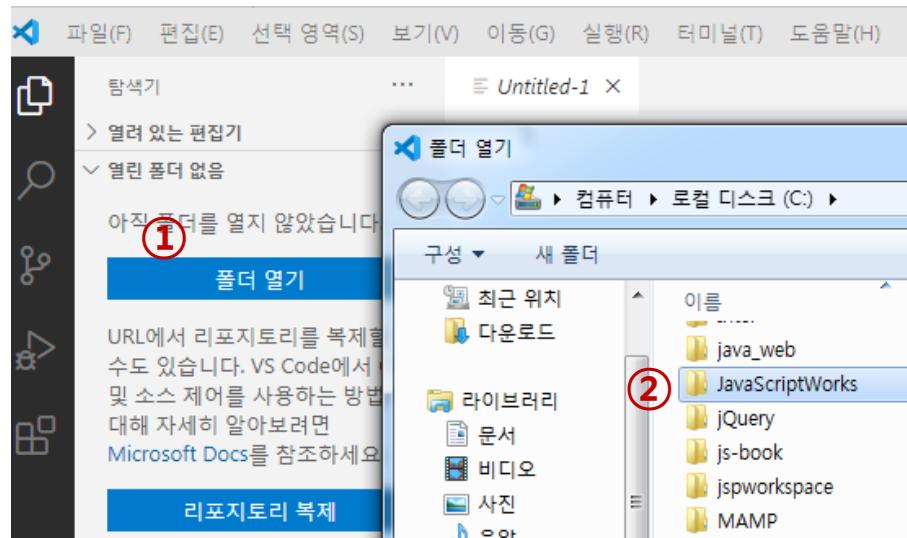
- 라이브 서버(Live Server) 설치



비주얼 스튜디오 코드 환경 설정

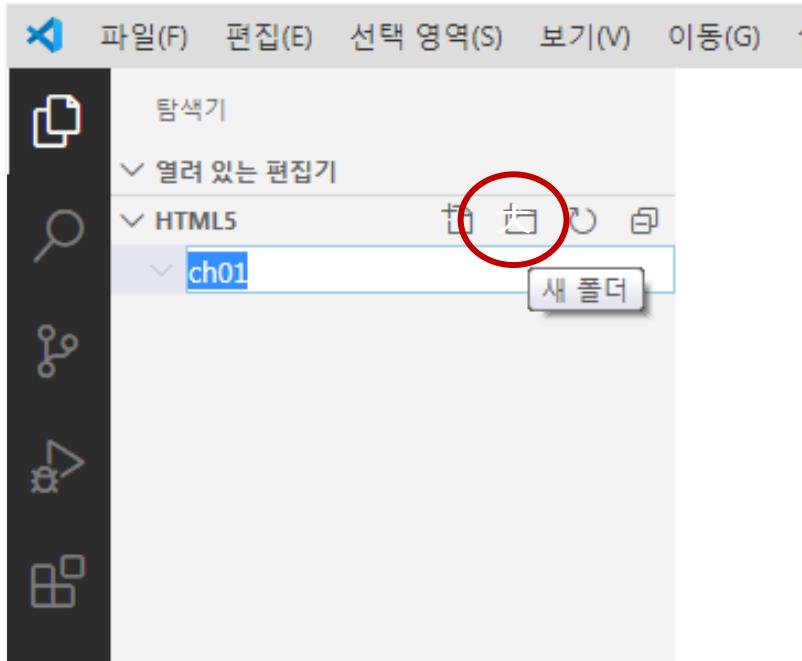


작업영역 폴더 만들기(탐색기)

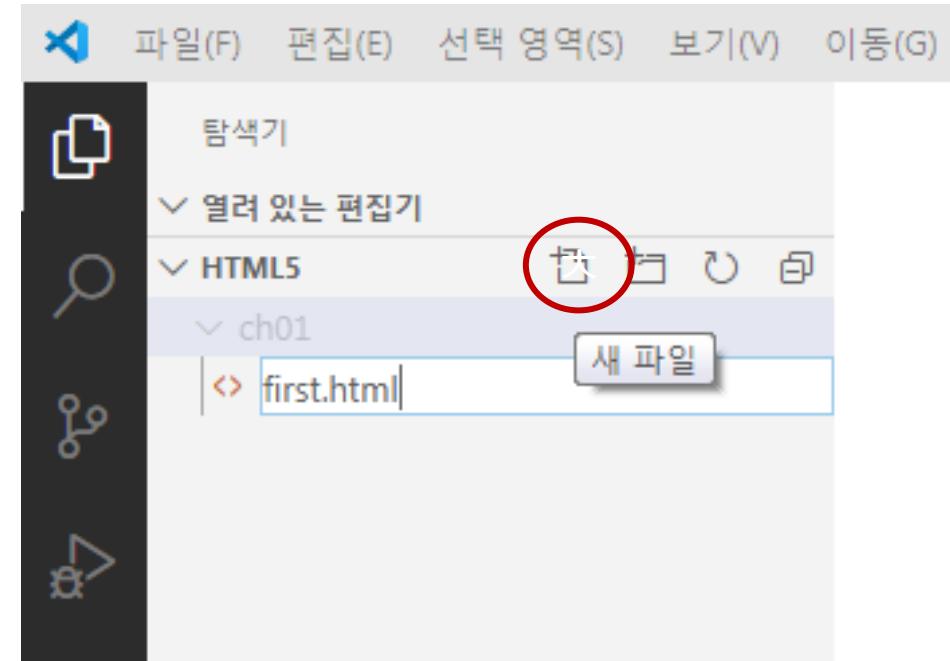


작업영역 폴더로 설정

비주얼 스튜디오 코드



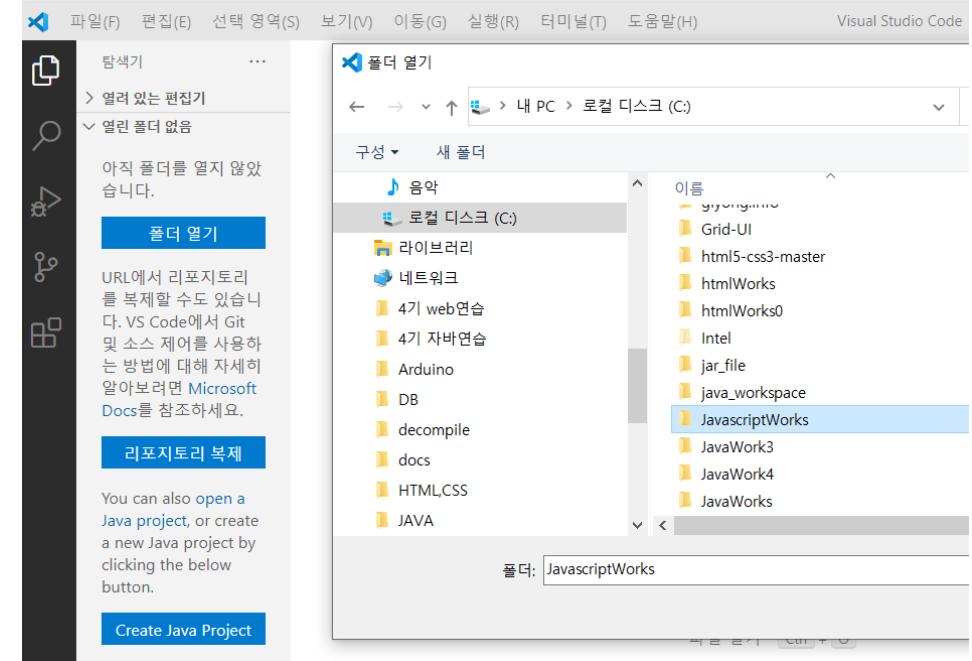
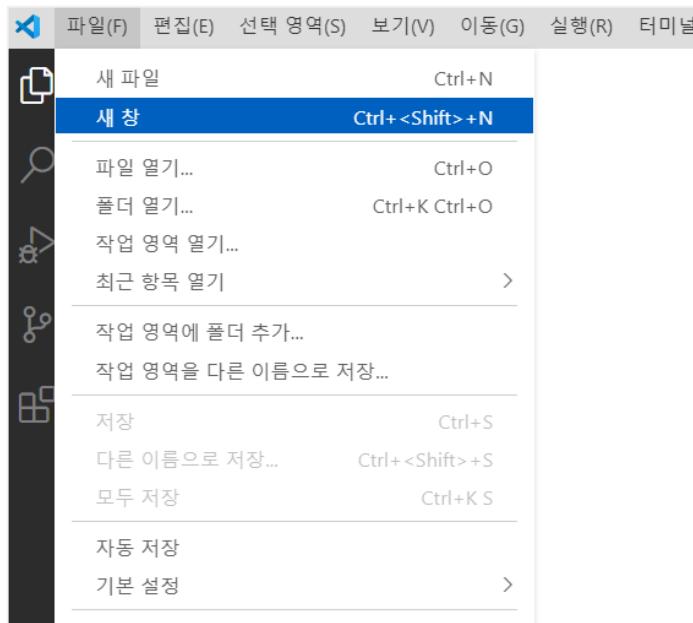
1. ch01 폴더 만들기



2. first.html 파일 만들기

비주얼 스튜디오 코드

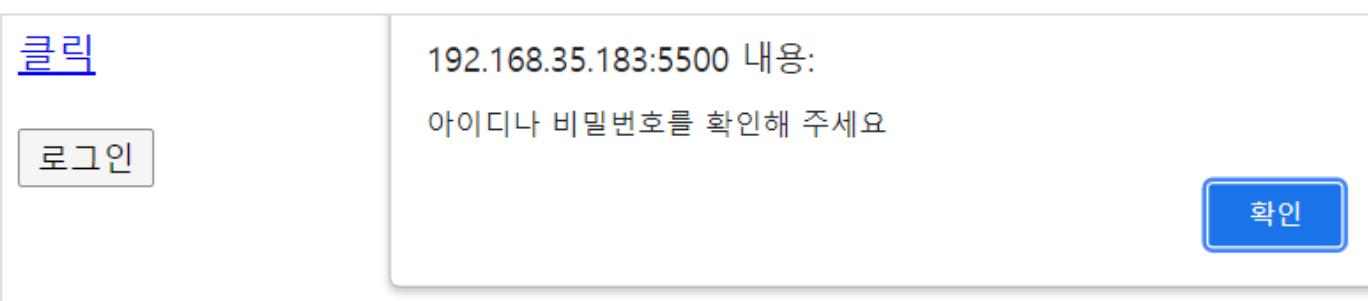
기존에 사용중인 경우 : 새창 열기 > 폴더 열기



JavaScript 작성하기

● 자바스크립트 함수 호출하기

태그의 **onclick=자바스크립트함수()**



<!-- 태그 속성인 onclick에 자바스크립트 함수를 호출함 -->

```
<a href="#" onclick="window.alert('안녕하세요~')">클릭</a><br><br>
```

```
<button onclick="alert('아이디나 비밀번호를 확인해 주세요')">로그인</button>
```

자바스크립트 Reference

자바스크립트 Reference

- <http://w3schools.com>
- Reference > JavaScript(HTML DOM) > DOM Window > Window Methods() >
alert() - 출력

alert()
atob()
blur()
btoa()
clearInterval()
clearTimeout()
close()
confirm()
focus()
getComputedStyle()
matchMedia()
moveBy()
moveTo()
open()
print()
prompt()

Window alert() Method

« Window Object

Example

Display an alert box:

```
alert("Hello! I am an alert box!!");
```

[Try it Yourself »](#)

JavaScript 작성하기

- 자바스크립트 태그 사용 – 내부 스크립트

- <script>~</script> 태그 안에 작성
- document(내장 객체)의 write() 메서드로 출력

자바 스크립트 예제

Hello~ JavaScript!

안녕~ 자바스크립트!

```
<h1>자바 스크립트 예제</h1>
<script>
    document.write("Hello~ JavaScript!<br>");
    document.write("<h3>안녕~ 자바스크립트!</h3>")
</script>
```



자바스크립트 Reference

자바스크립트 Reference

- <http://w3schools.com>
- Reference > JavaScript(HTML DOM) > DOM Document > write()

HTML DOM Reference

The references describe the properties and methods of each DOM object, along with examples.

| | | |
|---------------|----------------|----------|
| Attributes | Document | Element |
| Event Objects | HTMLCollection | Location |
| Screen | Style | Window |

HTML DOM write()

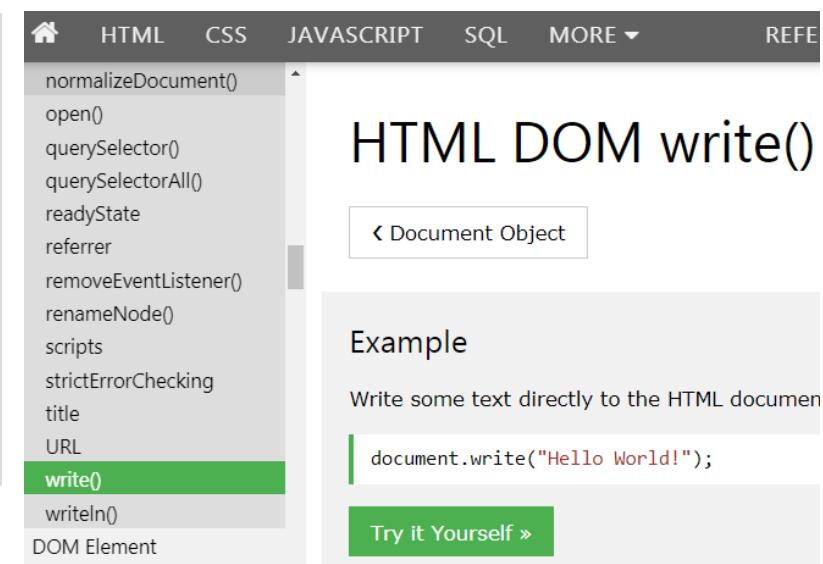
◀ Document Object

Example

Write some text directly to the HTML document

```
document.write("Hello World!");
```

Try it Yourself »



JavaScript 작성하기

자바스크립트 출력 – 외부 스크립트

<script src="js파일경로"></script>

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>Javascript</title>
    <script src="js/hello.js"></script>
</head>
<body>
</body>
</html>
```

```
// 자바스크립트 코드 작성
document.write("Hello~ Javascript!!<br>");

document.write("<h2>행복하세요~!!<h2>");
```

세미콜론 / 주석

문장 마침

- 문장 끝에 세미콜론(;) 사용

주석

- 한 줄 주석 : //~~
- 여러줄 주석 : /* ~~ */

```
/*
 * 자바스크립트 코드 작성
 * document.write()는 출력함수이다.
 */
document.write("Hello~ JavaScript!<br>");  
document.writ("<h3>안녕~ 자바스크립트!</h3>")
```

데이터 출력

콘솔창을 이용한 출력

- 크롬 브라우저 > F12(개발자도구) > Console
- console.log() 메서드**

```
<h2>콘솔(Console)로 보기</h2>
<script>
    //출력 -> console() 사용
    console.log("Hello~ Javascript!!");
    console.log(2000);
    console.log(2.5415);
    console.log(3 + 5);
    console.log("face");
</script>
```

Hello~ Javascript!!

2000

2.5415

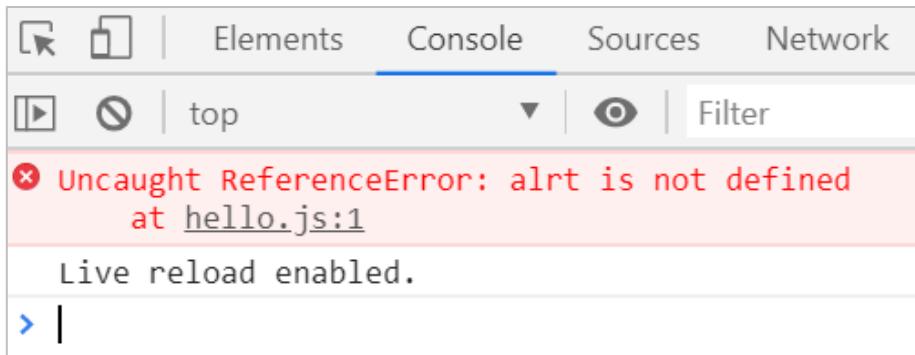
8

face

디버깅(Debugging)

디버깅(Debugging) - 오류(에러) 찾기

- 크롬 웹 브라우저의 검사 메뉴
- console 창에서 확인



```
<body>
  <script>
    document.write("Hello~ Javascript");
    document.writ("<h1>Hello~ Javascript<h1>"); ×
  </script>
```

자바스크립트 해석 엔진

웹 브라우저에는 **HTML 분석기(Parser)**, **CSS 분석기**,
자바스크립트 해석기(Interpreter)가 포함되어 있음



```
<!DOCTYPE html>          ①
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>Javascript</title>
    <style>
        body{
            text-align: center;  ②
        }
    </style>
</head>
<body>
    <script>           ③
        document.write("Hello~ Javascript");
    </script>
</body>
</html>
```

- ① HTML분석기는 HTML의 순서와 포함관계를 확인한다.
- ② CSS 분석기는 HTML 분석기가 태그 분석을 끝낸 다음 <style> 태그 사이의 스타일 정보를 분석한다.
- ③ 마지막으로 자바스트립트 해석기(기계어로 번역-**인터프리터**)가 <script> 태그 사이의 소스 코드를 해석한다.

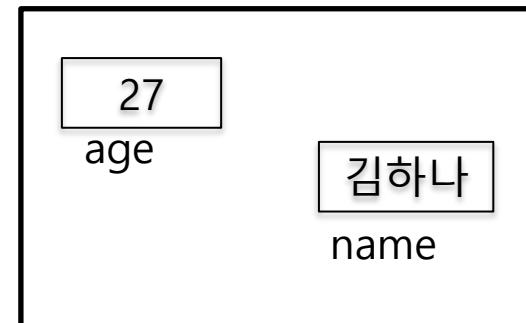
변수(Variable)

■ 변수란?

- 프로그램에서 사용되는 자료를 저장하기 위한 메모리 공간(영역)
- 할당받은 메모리의 주소 대신 부르는 이름 (메모리 주소: ac13bf00)
- 프로그램 실행 중에 값 변경 가능, variable 이라 함

■ 변수의 선언 및 초기화

- 변수 선언은 어떤 타입의 데이터를 저장할 것인지 그리고 변수이름은 무엇인지를 결정한다. **var, let** 키워드를 사용한다.
- **(자료형) 변수이름;** var age; let name;
- **(자료형) 변수이름 = 초기값;**
var age = 27, let name = "김하나"



변수(Variable)

● 변수의 선언과 초기화

var 변수명 = 값

let 변수명 = 값

● 변수명 작성시 주의

- 변수 이름은 숫자로 시작할 수 없고, 공백이 있으면 안됨.

틀린 예) var 3box; let sky 1234

- 변수 이름은 영문, 한글, 특수문자, 숫자의 결합으로 만든다.
- 영어 대소문자를 구별하며 **예약어**는 쓸 수 없다.

예약어) if, for, while 등

변수(Variable)

```
var name;  
name = "손흥민";  
  
var grade;  
grade = 2;  
  
//var class; 예약어는 사용 불가  
var schoolClass = 3;  
  
document.write(grade + "학년 ");  
document.write(schoolClass + "반" + "<br>");  
  
document.write(name + " 학생은 " + grade + "학년 " + schoolClass + "반입니다.");
```

variable.html

자료형(Type)

● 자료형(type)

프로그램에서 처리할 데이터의 형태로써,
사용할 데이터의 종류에 따라 메모리 공간을 적절하게 설정해 주는 것.

| 구 분(입력값) | 자료형 | 예시 |
|-----------------|---------|--|
| 숫자(정수, 실수) | number | <code>var age = 25</code> |
| 문자 | string | <code>var season = "winter"</code> |
| 불리언(true/false) | boolean | <code>let isMerried = true</code> |
| 배열 | object | <code>let seasons =["봄", "여름","가을", "겨울"]</code> |
| 객체 | object | <code>let date = new Date()</code> |

자료형(type)

- 크롬 개발자 도구(F12) > console

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. A red box highlights the clear button icon (a trash can) in the toolbar, with the text '화면 지우기 (clear)' (Clear screen) written below it. The main area displays a list of expressions and their results:

- > 5
< 5
- > 1.609
< 1.609
- > 5+3
< 8
- > 5-3
< 2
- > 'a'
< 'a'
- > "banana"
< 'banana'
- > '3'+5'
< '35'
- > 3+'5'
< '35'
- > typeof(5)
< 'number'
- > typeof('5')
< 'string'
- > typeof(1.60+2)
< 'number'
- > typeof('3'+5)
< 'string'
- 5 < 3
false
- var run=true
undefined
- run
true
- typeof(run)
'boolean'

자료형(type)

- 크롬 개발자 도구(F12) > console

```
var season = "여름"  
undefined  
  
season  
'여름'  
  
season = "가을"  
'가을'  
  
season  
'가을'
```

```
typeof(season)  
'string'  
  
typeof(seasons)  
'object'
```

```
var seasons = ["spring", "summer", "fall", "winter"]  
undefined  
  
seasons[0]  
'spring'  
  
season[2] = "autumn"  
'autumn'  
  
seasons  
▼ (4) [ 'spring', 'summer', 'fall', 'winter' ] ⓘ  
  0: "spring"  
  1: "summer"  
  2: "fall"  
  3: "winter"  
  length: 4  
▶ [[Prototype]]: Array(0)
```

자료형(type)

▪ undefined

자료형이 정의되지 않았을 때의 데이터 상태 : 변수를 선언할 때
아직 값이 할당되지 않은 경우

▪ null

데이터 값이 유효하지 않은 경우 : 변수에 값이 할당되었는데 잘못된
(유효하지 않은) 값인 경우

```
> var name
< undefined
> name
< ""
> name = "Elsa"
< "Elsa"
> name
< "Elsa"
```

```
> var inputName = prompt("이름을 입력해 주세요:")
< undefined
> inputName
< "kim"
> var inputName = prompt("이름을 입력해 주세요:");
< undefined
> inputName
< null
```

컴퓨터에서 데이터 표현하기

- **비트(binary digit)**

컴퓨터가 표현하는 데이터의 최소 단위로 2진수 하나의 값을 저장할 수 있는 메모리의 크기

컴퓨터는 0과 1로만 데이터를 저장함(0-> 신호꺼짐, 1-> 신호켜짐)

- 비트로 표현할 수 있는 수의 범위

| 비트수 | 표현할 수 있는 범위(십진수) | |
|------|---|-------|
| 1bit | 0, 1(0~1) | 2^1 |
| 2bit | 00, 01, 10, 11(0~3) | 2^2 |
| 3bit | 000, 001, 010, 011, 100, 101, 110, 111(0~7) | 2^3 |

10진수를 2진수로 바꾸기

▪ 진수 표현

| 10진수 | 2진수 | 16진수 | 10진수 | 2진수 | 16진수 |
|------|----------|------|------|-----------|------|
| 1 | 00000001 | 1 | 9 | 00001001 | 9 |
| 2 | 00000010 | 2 | 10 | 00001010 | A |
| 3 | 00000011 | 3 | 11 | 00001011 | B |
| 4 | 00000100 | 4 | 12 | 00001100 | C |
| 5 | 00000101 | 5 | 13 | 00001101 | D |
| 6 | 00000110 | 6 | 14 | 00001110 | E |
| 7 | 00000111 | 7 | 15 | 00001111 | F |
| 8 | 00010000 | 8 | 16 | 000010000 | 10 |

자리 올림 발생

컴퓨터에서 데이터 표현하기

▪ 진수 표현

```
<script>
    let num = 10;          //10진수 10
    let bNum = 0b1010      //2진수 10
    let hNum = 0xA;        //16진수 10

    document.write("10진수 : " + num + "<br>");
    document.write("2진수 : " + bNum + "<br>");
    document.write("16진수 : " + hNum + "<br>");
</script>
```

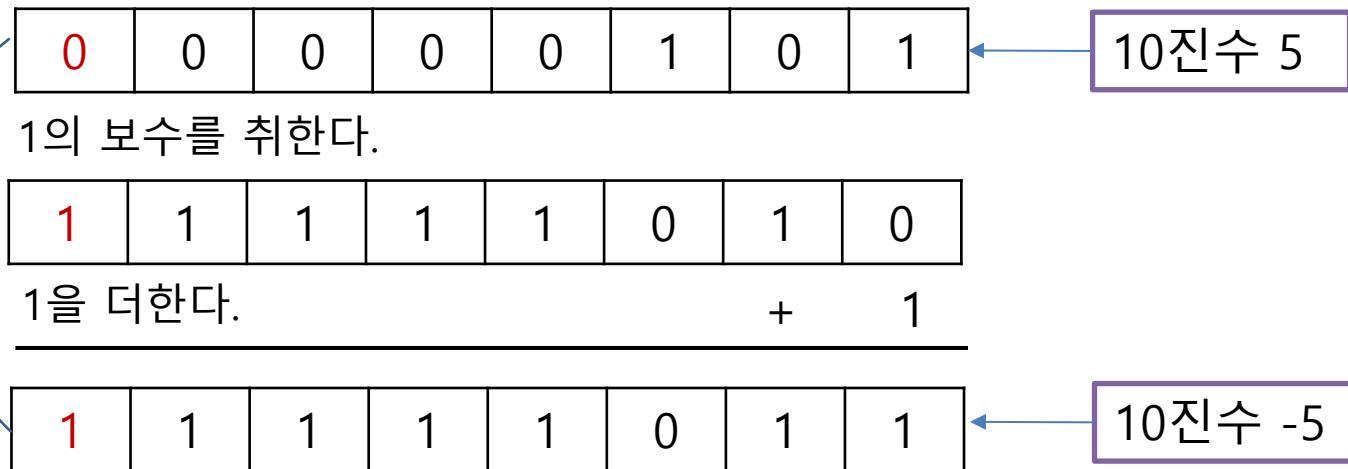
digit.html

부호 있는 수를 표현하는 방법

- 음의 정수는 어떻게 표현할까?

- 정수의 가장 왼쪽에 존재하는 비트는 부호비트입니다.
(양의 정수는 0, 음의 정수는 1을 붙인다.)
- 음수를 만드는 방법은 2의 보수를 취한다.(1의 보수는 0과 1을 반대로 바꿈)

두 수를 더
하면 0이됨
100000000
~~맨앞 1은 제거됨~~



- 1은 11111111 (0은 00000000)
- 2는 11111110(1을 뺀다)
- 3은 11111101(1을 뺀다)
- 4는 11111100(1을 뺀다)
- 5는 11111011(1을 뺀다)

아스키 코드

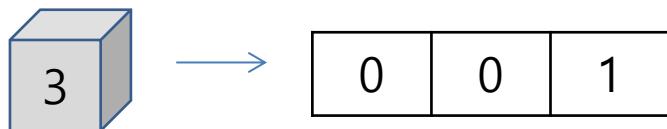
● 아스키 코드(ASCII Code)

아스키 코드는 미국 [ANSI](#)에서 표준화한 정보교환용 7비트 부호체계이다.

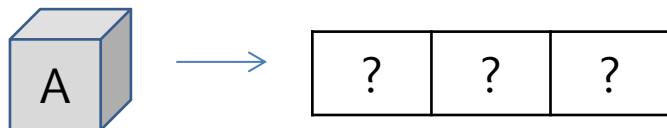
000(0x00)부터 127(0x7F)까지 총 128개의 부호가 사용된다.

영문 키보드로 입력할 수 있는 모든 기호들이 할당되어 있는 부호 체계이다.

10진수를 2진수로 변환



문자를 2진수로 변환



저장하는 문자에 해당하는 숫자를 지정하고 메모리에 저장할때는 그 숫자를 비트 단위로 바꾸어 저장

아스키 코드

● 아스키 코드(ASCII Code)

| Dec | Hx | Oct | Char | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|-----|--------|-----|----------------------------|-----|--------|-------|-------|-----|-----|--------|-------|------|-----|--------|--------|-----|------|-----|
| 0 | 0 000 | NUL | (null) | 32 | 20 040 | | Space | | 64 | 40 100 | @ | Ø | 96 | 60 140 | ` | ` | | |
| 1 | 1 001 | SOH | (start of heading) | 33 | 21 041 | ! | ! | ! | 65 | 41 101 | A | A | 97 | 61 141 | a | a | | |
| 2 | 2 002 | STX | (start of text) | 34 | 22 042 | " | " | " | 66 | 42 102 | B | B | 98 | 62 142 | b | b | | |
| 3 | 3 003 | ETX | (end of text) | 35 | 23 043 | # | # | # | 67 | 43 103 | C | C | 99 | 63 143 | c | c | | |
| 4 | 4 004 | EOT | (end of transmission) | 36 | 24 044 | $ | \$ | \$ | 68 | 44 104 | D | D | 100 | 64 144 | d | d | | |
| 5 | 5 005 | ENQ | (enquiry) | 37 | 25 045 | % | % | % | 69 | 45 105 | E | E | 101 | 65 145 | e | e | | |
| 6 | 6 006 | ACK | (acknowledge) | 38 | 26 046 | & | & | & | 70 | 46 106 | F | F | 102 | 66 146 | f | f | | |
| 7 | 7 007 | BEL | (bell) | 39 | 27 047 | ' | ' | ' | 71 | 47 107 | G | G | 103 | 67 147 | g | g | | |
| 8 | 8 010 | BS | (backspace) | 40 | 28 050 | (| (| (| 72 | 48 110 | H | H | 104 | 68 150 | h | h | | |
| 9 | 9 011 | TAB | (horizontal tab) | 41 | 29 051 |) |) |) | 73 | 49 111 | I | I | 105 | 69 151 | i | i | | |
| 10 | A 012 | LF | (NL line feed, new line) | 42 | 2A 052 | * | * | * | 74 | 4A 112 | J | J | 106 | 6A 152 | j | j | | |
| 11 | B 013 | VT | (vertical tab) | 43 | 2B 053 | + | + | + | 75 | 4B 113 | K | K | 107 | 6B 153 | k | k | | |
| 12 | C 014 | FF | (NP form feed, new page) | 44 | 2C 054 | , | , | , | 76 | 4C 114 | L | L | 108 | 6C 154 | l | l | | |
| 13 | D 015 | CR | (carriage return) | 45 | 2D 055 | - | - | - | 77 | 4D 115 | M | M | 109 | 6D 155 | m | m | | |
| 14 | E 016 | SO | (shift out) | 46 | 2E 056 | . | . | . | 78 | 4E 116 | N | N | 110 | 6E 156 | n | n | | |
| 15 | F 017 | SI | (shift in) | 47 | 2F 057 | / | / | / | 79 | 4F 117 | O | O | 111 | 6F 157 | o | o | | |
| 16 | 10 020 | DLE | (data link escape) | 48 | 30 060 | 0 | Ø | Ø | 80 | 50 120 | P | P | 112 | 70 160 | p | p | | |
| 17 | 11 021 | DC1 | (device control 1) | 49 | 31 061 | 1 | 1 | 1 | 81 | 51 121 | Q | Q | 113 | 71 161 | q | q | | |
| 18 | 12 022 | DC2 | (device control 2) | 50 | 32 062 | 2 | 2 | 2 | 82 | 52 122 | R | R | 114 | 72 162 | r | r | | |
| 19 | 13 023 | DC3 | (device control 3) | 51 | 33 063 | 3 | 3 | 3 | 83 | 53 123 | S | S | 115 | 73 163 | s | s | | |
| 20 | 14 024 | DC4 | (device control 4) | 52 | 34 064 | 4 | 4 | 4 | 84 | 54 124 | T | T | 116 | 74 164 | t | t | | |
| 21 | 15 025 | NAK | (negative acknowledgement) | 53 | 35 065 | 5 | 5 | 5 | 85 | 55 125 | U | U | 117 | 75 165 | u | u | | |

유니 코드(Unicode)

● 유니 코드(Unicode)

유니코드는 전 세계의 모든 문자를 컴퓨터에서 일관되게 표현하고 다룰 수 있도록 설계된 산업 표준이다.

인코딩 포맷: UTF-8, UTF-16 등

//문자 셋 - 아스키 코드, 유니코드

```
var ch1 = 65;
console.log(ch1);
console.log(String.fromCharCode(ch1));
//문자로 출력 'A'
```

charset.html

//아스키 코드

```
var ch2 = 'B';
console.log(ch2);
console.log(ch2.charCodeAt(0));
//코드값으로 출력 - 66, 0번 인덱스의 자료를 변환
```

//유니코드

```
var ch3 = '나';
console.log(ch3);
console.log(ch3.charCodeAt(0)); //45208
```

