## Web

## **Contents**

PART 01
Web Architecture

01. Front-End

02. Web Architecture 이해

PART 02 HTML 기본

01. HTML 개요

02. Web Page 구성 요소 및 속성

02. Web Page 공간 분할

PART 03 HTML 활용

01. 데이터 정렬

02. 데이터 입력

03. Web Page 이동

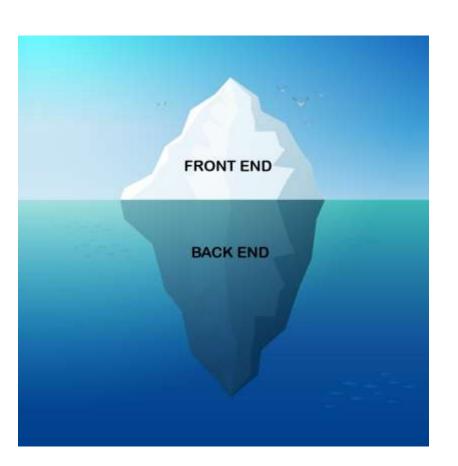
04. 멀티미디어 지원

## Part01. Web Architecture

## 01. Front-End

• Front-End란?

보이는 부분??



• Front-End란?

사용자에게 다양한 컨텐츠를 제공하고,

요청에 반응하여 동작하는 분야

• Front-End란?

UI? UX?

• Front-End란?

UI(User Interface) : 사용자 환경

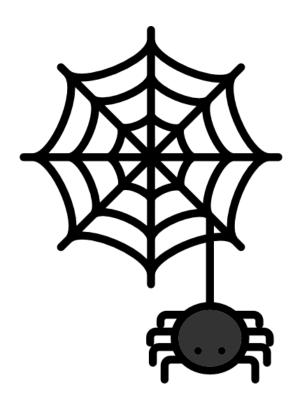
UX(User eXperience): 사용자 경험

#### • Front-End란?



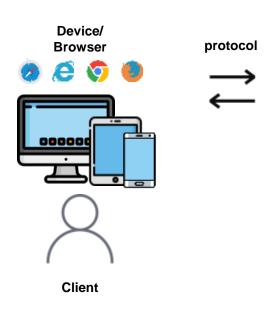
## 02. Web Architecture 이해

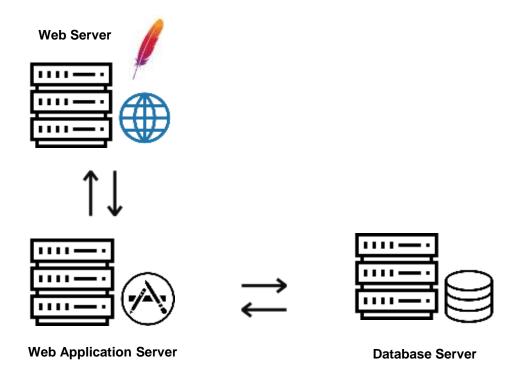
• Web 이란?



Web (World Wide Web) 이란? 000000

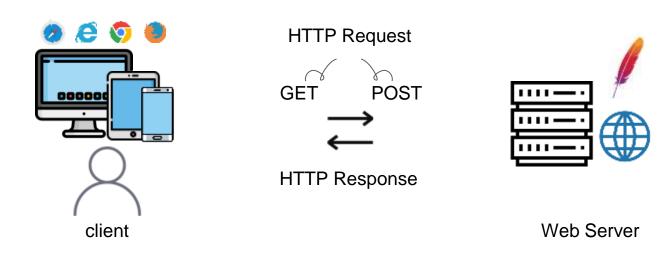
#### • Web Architecture 이해





#### HTTP

### HyperText Transfer Protocol (하이퍼본문 전송 규약)



# Part02. HTML 기본

## 01. HTML 개요

### ● HTML 이란?

#### **HyperText Markup Language**















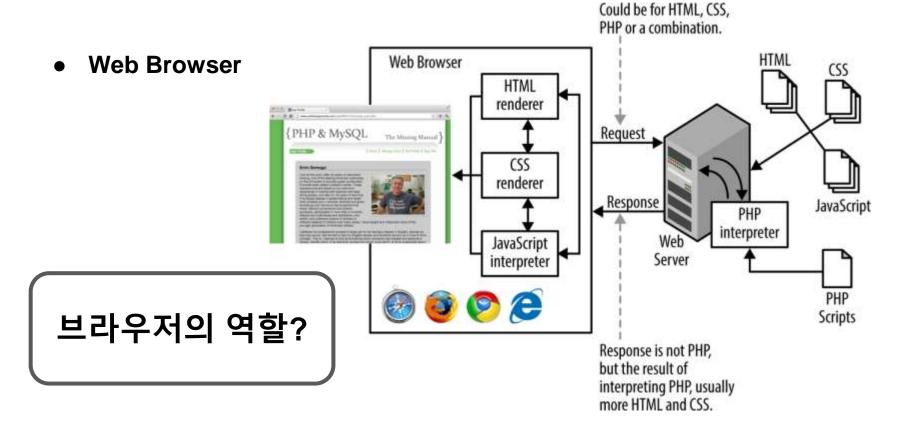
#### Web Browser



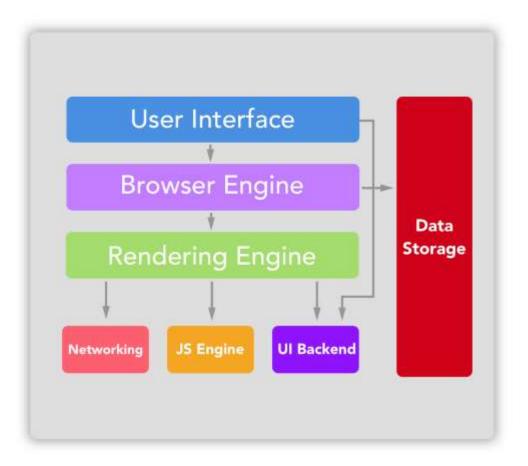




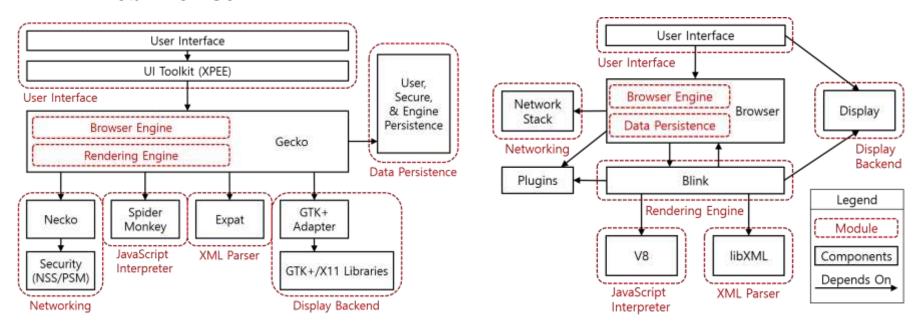




Web Browser

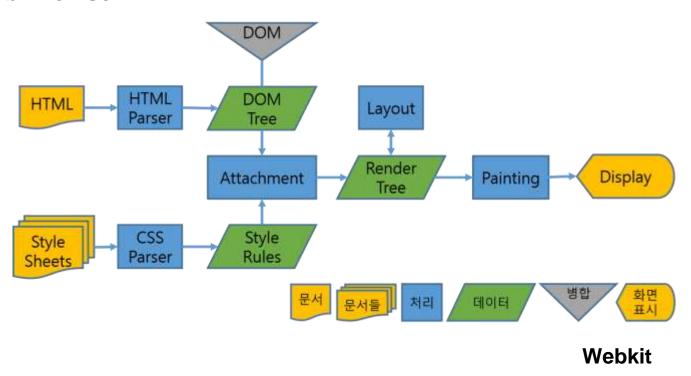


#### Web Browser

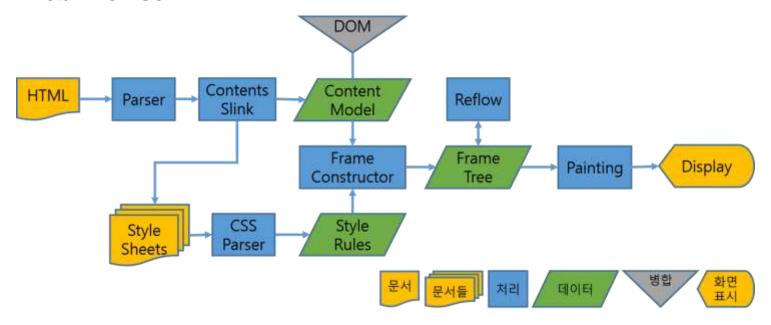


Firefox

#### Web Browser



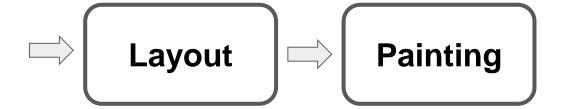
#### Web Browser



Gecko

• Web Browser





#### Web Editor

#### **Visual Studio Code**



## Let's make it





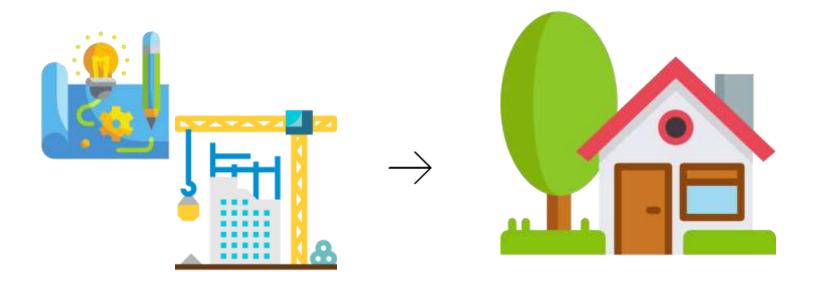


# 02. Web Page 구성 요소 및 속성



# HTML 기본 구조

● HTML 기본 구조 (Structure)



● HTML 기본 구조 (Structure)



#### ● HTML 기본 구조 (Structure)

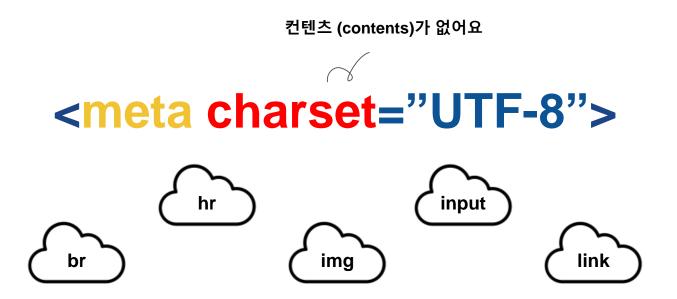
```
1 <!DOCTYPE html>
 <html lang="en"> 2
 <head> 3
     <meta charset="UTF-8">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
     <meta http-equiv="X-UA-Compatible" content="ie=edge">
     <title>Document</title>
 </head>
 <body> 4
     <h1>나만의 SNS 만들기</h1>
     수업에 오신것을 환영합니다
 </body>
 </html>
```

## HTML 기본 요소 및 문법

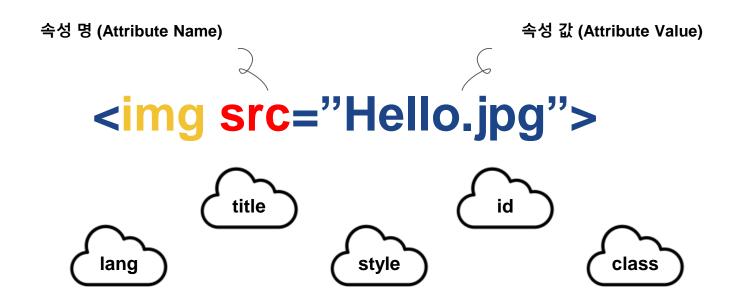
● HTML 요소 (Element)



● HTML 빈 요소 (Empty Element)



● 속성 (Attribute)



● 주석 (Comments)





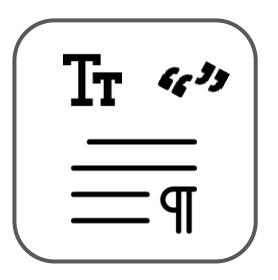




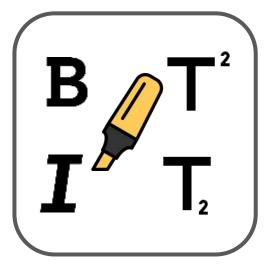




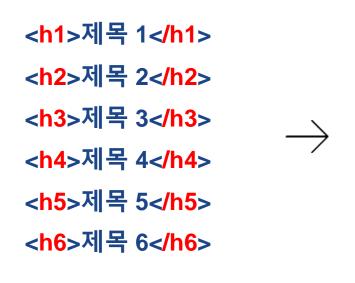
제목 (Heading)

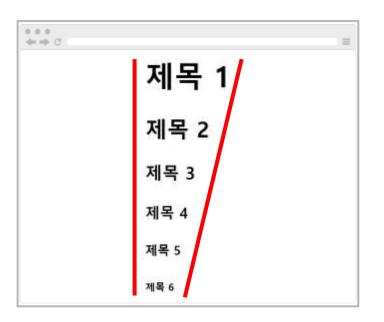


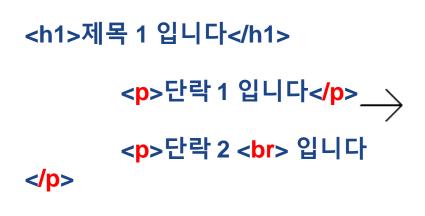
단락 (Paragraph)

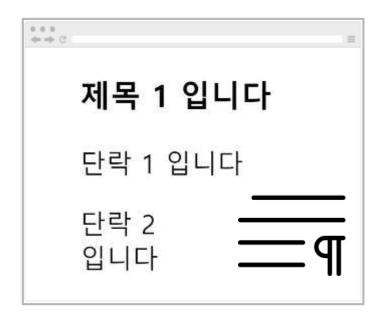


형태 (Formatting)









● HTML 텍스트 (Text)

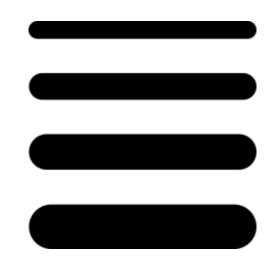
<h1>제목 1 입니다</h1>
단락 3 입니다 
<hr>
단락 4 입니다

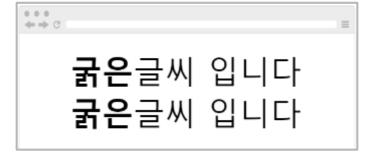


● HTML 텍스트 (Text)

<b>굵은</b>글씨 입니다

<strong>굵은</strong>글씨 입니다



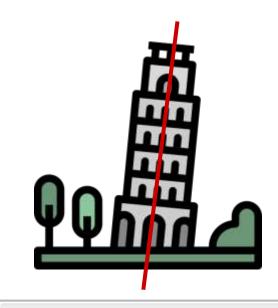


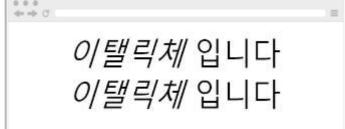
● HTML 텍스트 (Text)

<i>이탤릭체</i> 입니다

<em>이탤릭체</em> 입니다







● HTML 텍스트 (Text)

<mark>마크</mark>입니다





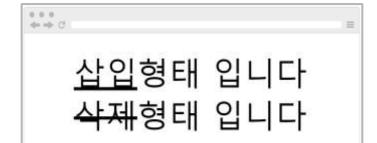
● HTML 텍스트 (Text)

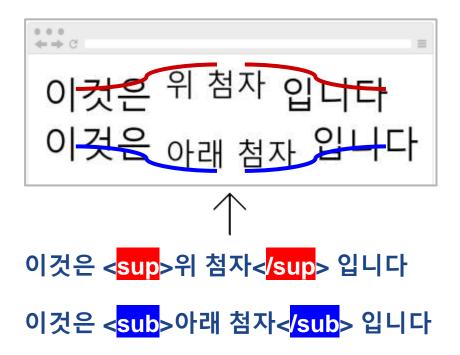
<ins>삽입</ins>형태입니다

<del>삭제</del>형태 입니다









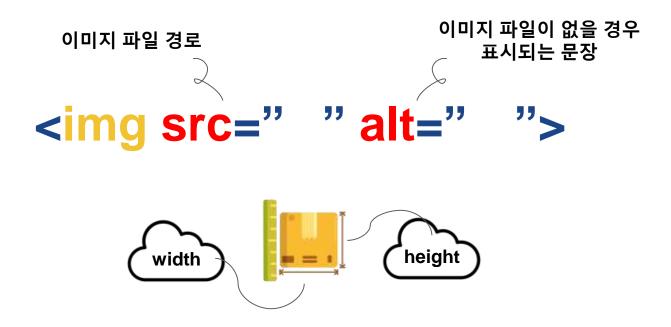


● HTML 스타일 (Style)





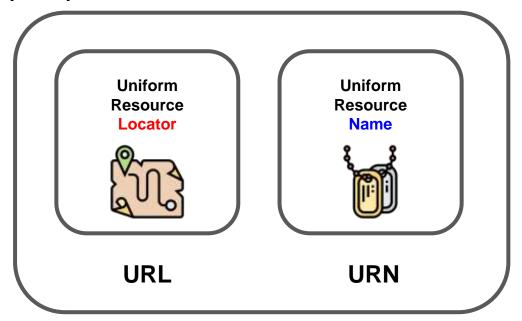
• HTML 이미지 (Image)



● HTML 경로 (Path)



● HTML 경로 (Path)

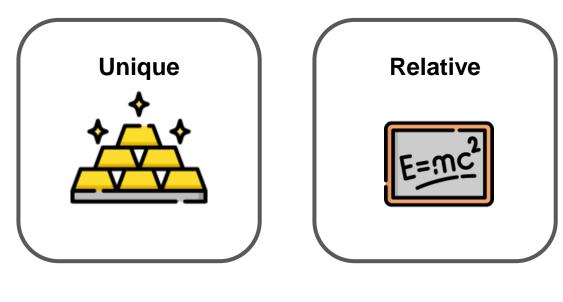


**URI (Uniform Resource Identifier)** 

HTML URL (Uniform Resource Locator)

네트워크 **Web Server** 요청 시 규약 연결기기에 부여되는 (주로 http) 고유 번호 <프로토콜://ip주소 :port번호/경로/파일명?쿼리> 웬서버 웹서버 접근 관문 웹서버 자원 경로 요청 파라미터 (http는 80) (key1=value1)

HTML URL (Uniform Resource Locator)



절대 (Absolute) URL

상대 (Relative) URL



● HTML 링크 (Link)



● HTML 링크 (Link)



# 02. Web Page 공간 분할

● 마음에 드는 web page 고르기?!



• 공간 분할 적용의 필요성

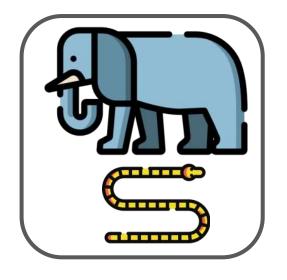


THE PART OF THE PA

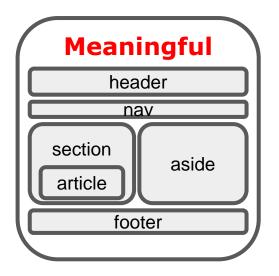
공간 분할 적용

공간 분할 미적용

#### • 공간 분할



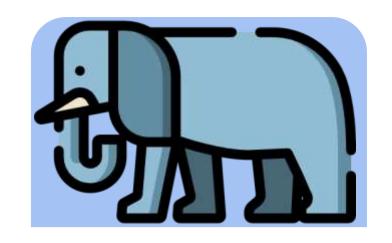


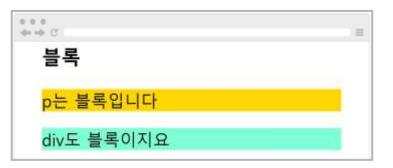


의미적 분할

• 공간 분할

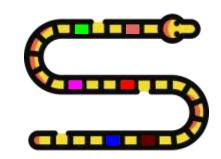
```
<h3>블록</h3>
p는 블록입니다
<div>div도 블록이지요</div> \
```

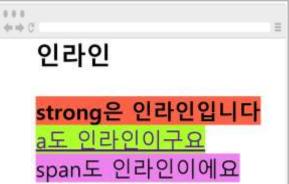




• 공간 분할

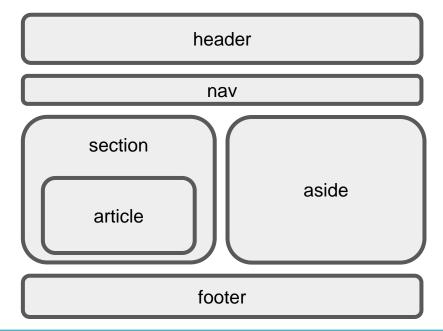






#### • 공간 분할

시멘틱 (semantic) 🏶



## Part03. HTML심화

#### PART 03 HTML 심화









### PART 03 HTML 심화



## 01. 데이터 정렬

#### PART 03 HTML 활용

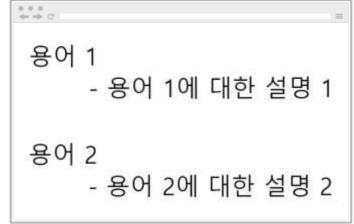
● 리스트 (list)

```
<dl>
    <dl><dt>용어 1</dt>
    <dd>+ 용어 1에 대한 설명 1</dd>
    <br><br><br><br><dt><br><dt><br><dt><br><br></dl>
    <dd>+ 용어 2에 대한 설명 2</dd>
    </dl>
    <dd>+ 용어 2에 대한 설명 2</dd>
    </dl>
    </dl>
    </dl>
    <dt>+ 용어 2에 대한 설명 2</dd>
    </di>
    </di>
```





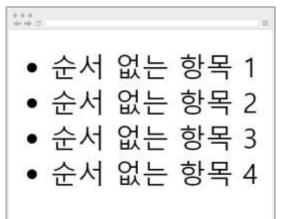




#### PART 03 HTML 활용

● 리스트 (list)





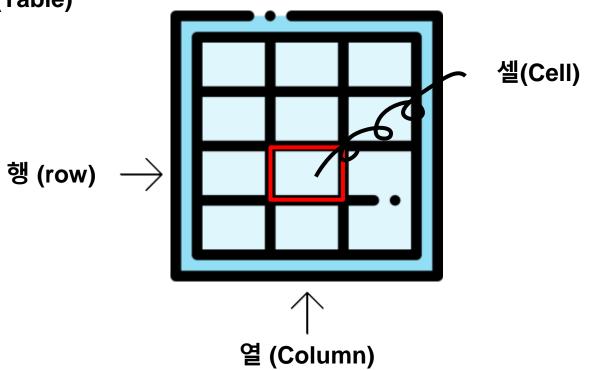
● 리스트 (list)

1 2 3

→ → C

1. 순서 있는 항목 1 2. 순서 있는 항목 2 3. 순서 있는 항목 3 4. 순서 있는 항목 4

● 테이블 (Table)



# ● 테이블 (Table)

```
<caption>테이블</caption>
제목 1
 제목 2
 제목 3
셀 1-1
 셀 1-2
 셀 1-3
```



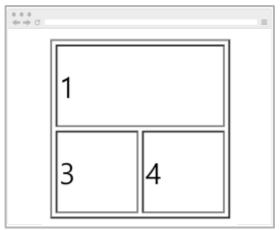


● 테이블 (Table)

```
            1

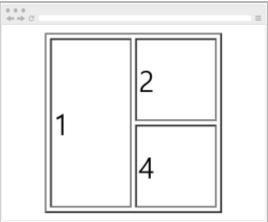
            >
```





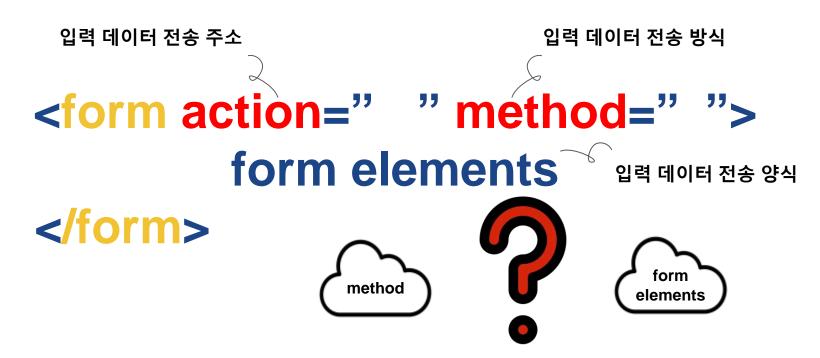
• 테이블 (Table)







# 02. 데이터 입력

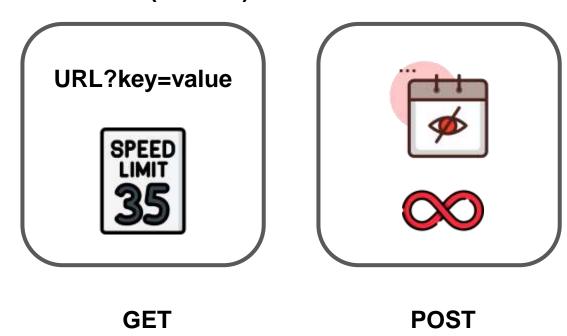


● 입력 데이터 전송 방식(Method)



81

• 입력 데이터 전송 방식 (Method)



● 입력 데이터 전송 양식 (Form Elements)



**Form Elements** 

```
Tr 🗀 a
```

```
<form action="">
텍스트를 입력하세요 : <br><input type="text">
</form>
```



```
<form action="">
ID : <br>
ID : <br>
<input type="text"
name="userId"><br>
PW : <br>
<input type="password"
name="password">
</form>
```





```
<form action="">
  <input type="radio"
    name="radio" checked>
      라디오 1 <br>
  <input type="radio"
    name="radio">
      라디오 2 <br>
  <input type="radio"
    name="radio">
      라디오 3 <br>
  <input type="radio"
    name="radio">
      라디오 4 <br>
</form>
```





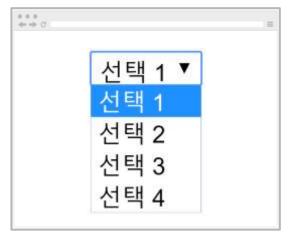
</form>

```
데이터 입력ction="">
      <input type="checkbox"
            name="check"
    checked>
              체크박스 1 <br>
      <input type="checkbox"</pre>
            name="check">
              체크박스 2 <br>
      <input type="checkbox"</pre>
        name="check">
              체크박스 3 <br>
      <input type="checkbox"
            name="check"
    disabled>
              체크박스 4
```



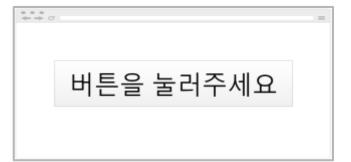








```
<br/>
<button type="button">
<br/>
버튼을 눌러주세요
</button>
```



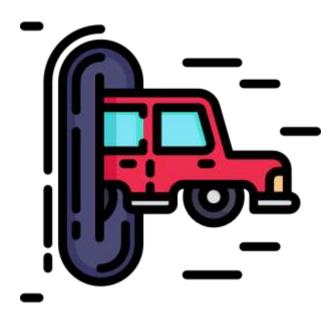
```
<form action="">
  텍스트를 입력하세요:
  <br
 <input type="text"
        name="text"
        placeholder="텍스트
를
                작성해 주세
요">
  <br/>br>
  <input type="submit"
        value="전송">
</form>
```





# 03. Web Page 이동

• 페이지 이동

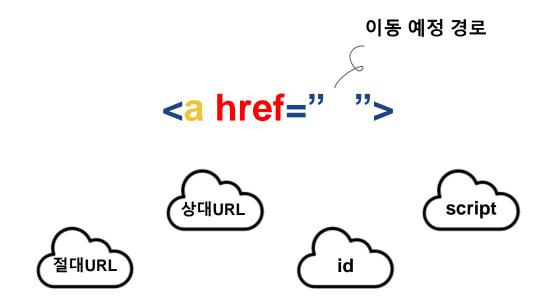


● HTML 링크 (Link)





● HTML 링크 (Link)





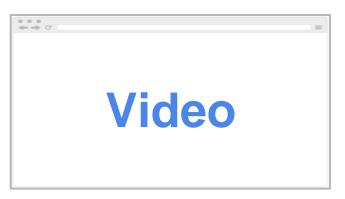
# 04. 멀티미디어 지원

• 멀티미디어 비디오 (Video)



```
<video>
<source src="삽입 예정
비디오 파일 경로"
type=video/파일 확장자

명>
</video>
```



• 멀티미디어 오디오 (Audio)

```
<audio>
<source src="삽입 예정
    오디오 파일 경로"
    type=audio/파일 확장자

명>
</audio>
```

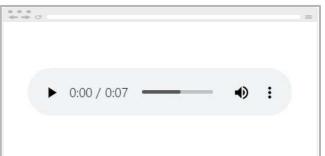




• 멀티미디어 오디오 (Audio)



```
<audio>
</audio>
```



# ● 멀티미디어 오디오 (Audio)



● 데이터 속성 (data-\*)

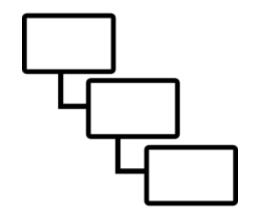
# ● 특수 기호 (Entity)

Character	Entity	Note
&	&	Interpreted as the beginning of an entity or character reference.
<	<	Interpreted as the beginning of a <u>tag</u>
>	>	Interpreted as the ending of a <u>tag</u>
"	"	Interpreted as the beginning and end of an attribute's value.

CSS



CSS (Cascading Style Sheets)



Cascading 구조화된 문서(ex. HTML)를 순차적, 종속적으로

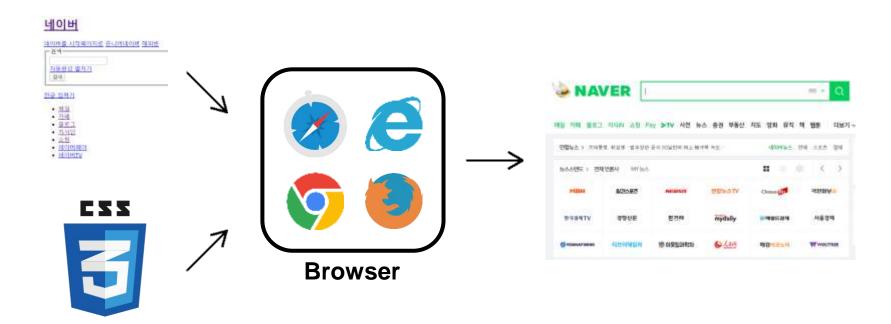


Style 어떻게 꾸밀 것인가에 대해 정의

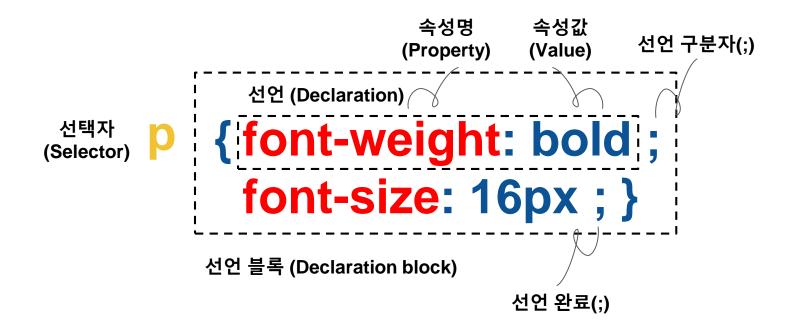


Sheet별도 목록 또는파일로 관리

CSS (Cascading Style Sheets)

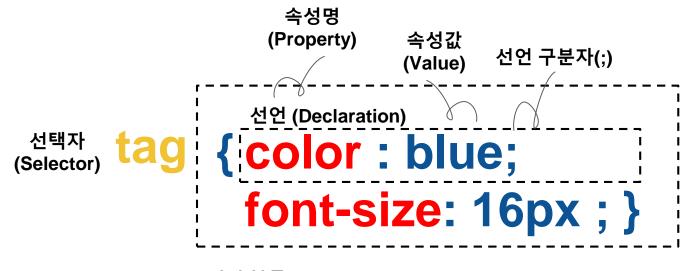


#### • CSS 문법



#### PART 01 CSS 기초

#### CSS 문법



선언 블록 (Declaration block)

## **JavaScript**

### **Contents**

#### PART 01 JavaScript 기초

- 01. JavaScript 개요
- 02. 타입과 연산자
- 03. 흐름 제어

#### PART 02 JavaScript 활용

- 01. 함수
- 02. 객체와 메소드
- 03. DOM

Part01. JavaScript 기초

01. JavaScript 개요

**HTML** 



CSS



#### **JavaScript**



출처 : dribbble / ohmyfiesta / gfycat

https://www.ecma-international.org/

https://ko.wikipedia.org/wiki/인터프리터

● JavaScript의 필요성

https://v8.dev/

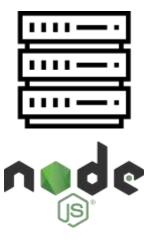
https://kangax.github.io/compat-table/es6/

https://gs.statcounter.com/browser-market-share#m

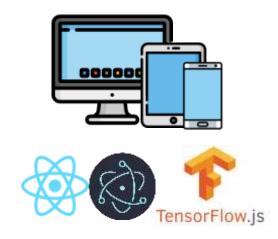




Front-End 웹 브라우저에서 작동하는 유일한 프로그래밍 언어

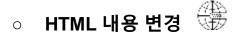


Back-End 서버 측 프로그래밍에서도 사용 되고 있음



Cross Platform 데스크톱, 웹, 앱, 머신러닝 등 플랫폼을 넘나들며 개발

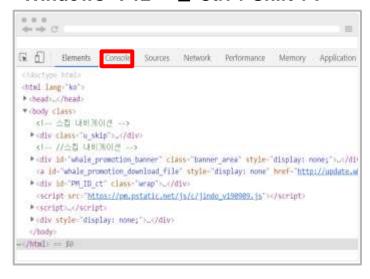
● JavaScript 기본 기능

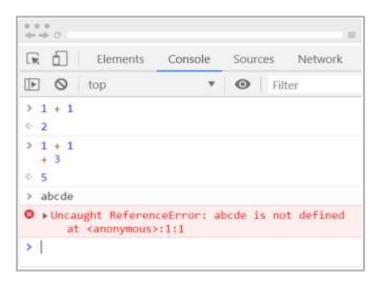


- HTML 속성 변경
- HTML 스타일 변경

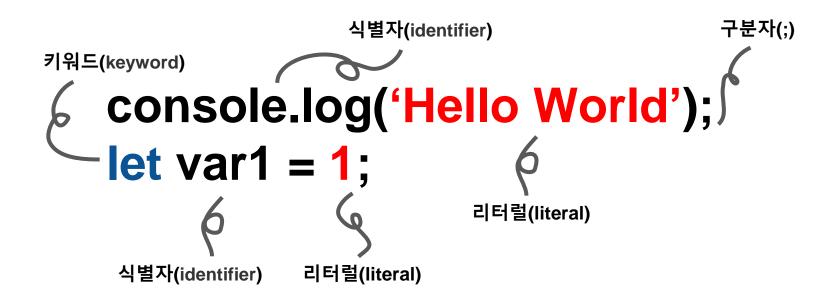
#### ● 크롬 개발자 도구

○ Windows F12 또는 Ctrl + Shift + I





● JavaScript 문법 ∰



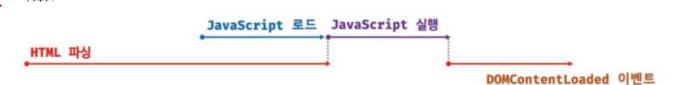
JavaScript 적용



○ <script> 태그

○ 외부 script 파일 (.js)

o defer vs async



● JavaScript 출력

- o console.log()
- o alert()
- o document.write()
  - innerHTML

# 02. 타입과 연산자

#### • 변수와 상수







const



#### • 변수와 상수

await	break	case	catch	class	const
continue	debugger	default	delete	do	else
enum	export	extends	false	finally	for
function	if	implements*	import	in	instanceof
interface*	let*	new	null	package*	private*
protected*	public*	return	super	static*	switch
this	throw	true	try	typeof	var
void	while	with	yield*		

## 이 고양이를 설명해 보세요

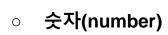




#### ● 원시 타입(primitive type)

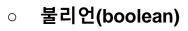
구분	데이터 타입	설명		
원시 타입	숫자 <sup>rumber</sup> 타입	숫자. 정수와 실수 구분 없이 하나의 숫자 타입만 존재		
	문자열 <sup>string</sup> 타입	문자열		
	불리언 <sup>boolean</sup> 타입	논리적 참(true)과 거짓(false)		
	undefined 타입	var 키워드로 선언된 변수에 암묵적으로 할당되는 값		
	null 타입	값이 없다는 것을 의도적으로 명시할 때 사용하는 값		
	심벌 <sup>symbol</sup> 타입	ES6에서 추가된 7번째 타입		
객체 타입		객체, 함수, 배열 등		

● 원시 타입(primitive type)





문자열(string)





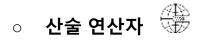
undefined







● 연산자 (Operator) 🥞



○ 대입 연산자 🌐

## 03. 흐름 제어



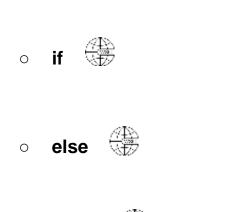


• 조건 관련 연산자

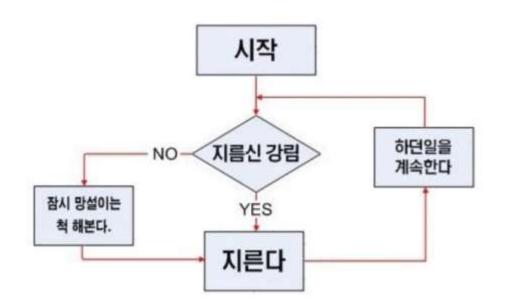
○ 비교 연산자

○ 논리 연산자

● 조건문 (condition) ③



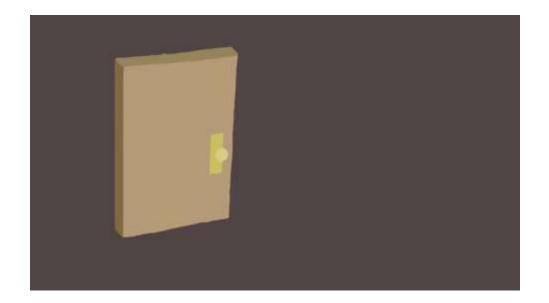
else if



### ● 반복문 (loop)



o for 🔀



Part02. JavaScript 심화

# 01. 함수

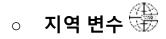


● 함수 문법 ∰

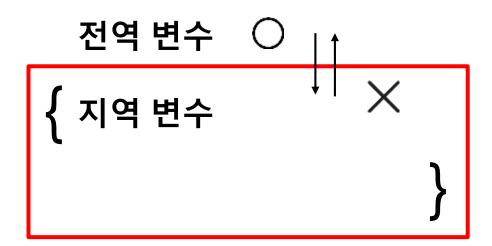
```
function 함수명(매개변수1, 매개변수2, ...) {
    함수가 호출되었을 때 실행하고자 하는 실행문

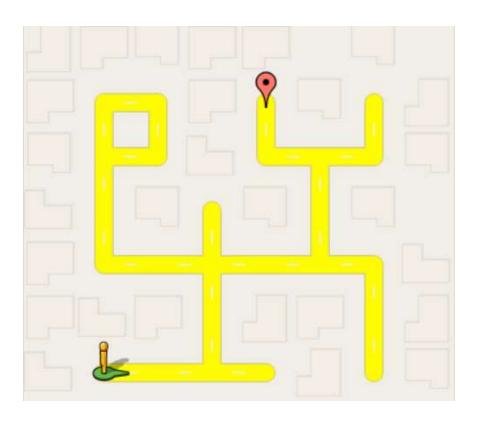
return 함수 밖으로 내보내고 싶은 값
}
```

• 변수의 범위



o 전역 변수





# 02. 객체와 메소드



• 객체 (Object) 😩



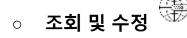
○ 프로퍼티

o 메소드와 this



● 배열 (Array) 🥮





· 추가

반복



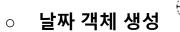
• Number

- o toString()
- o Number()
- o parseInt()

○ toFixed() 🦀



• 날짜 (Date)







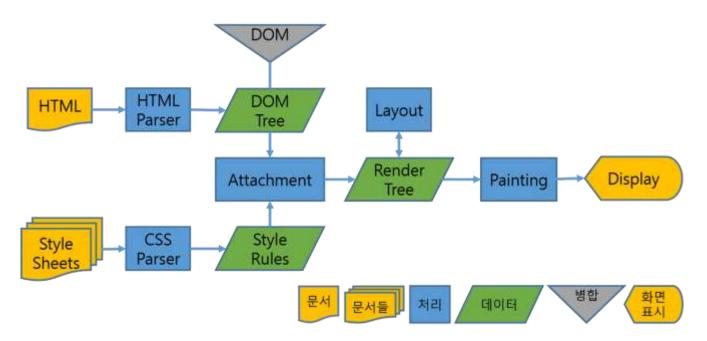


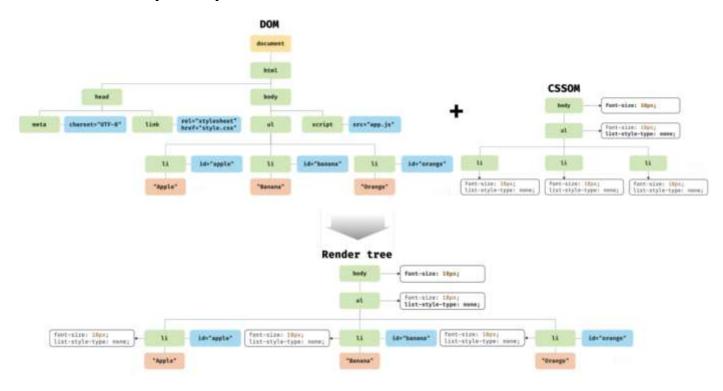
● 문자열 (String)

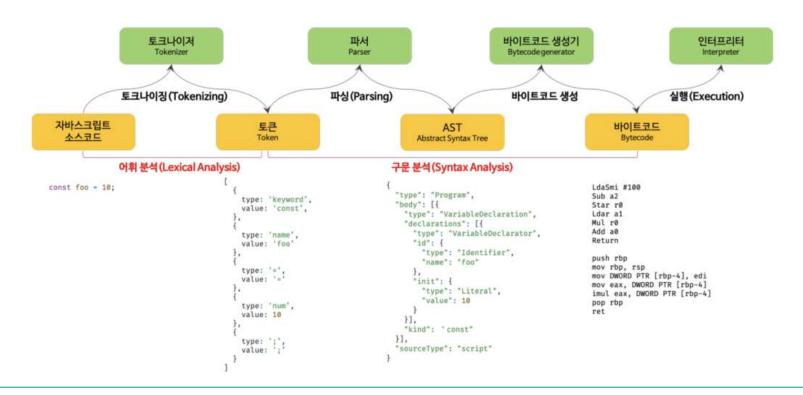


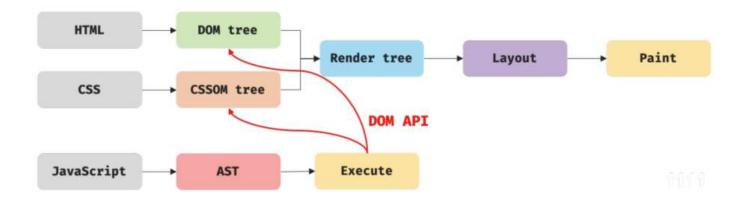
- o substring() // / replace()
- o split() / concat()

## 03. DOM









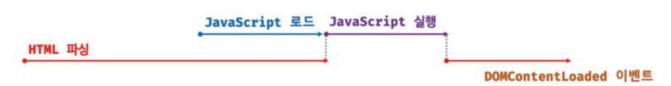
JavaScript 적용

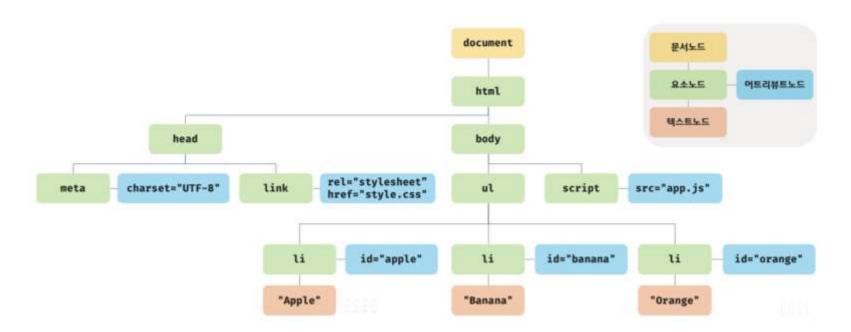


○ <script> 태그

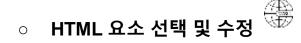
○ 외부 script 파일 (.js)

o defer vs async



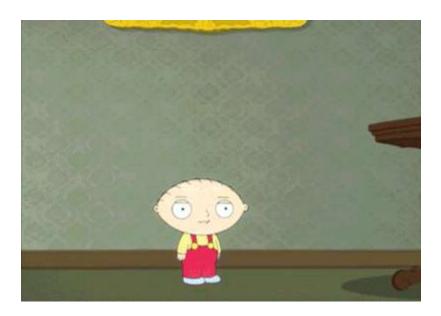


• Document 객체 🛞



○ HTML 요소 추가 🍪

○ 자식 노드 접근 🍪



• 비동기통신



。 동기 vs 비동기

o Axios

o jsonplaceholder

o unplash API

# **JavaScript**

**HTML** 



CSS



#### **JavaScript**



출처 : <u>dribbble</u> / <u>ohmyfiesta</u> / <u>gfycat</u>

https://www.ecma-international.org/

https://ko.wikipedia.org/wiki/인터프리터

● JavaScript의 필요성

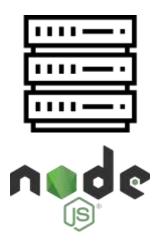
https://v8.dev/

https://kangax.github.io/compat-table/es6/

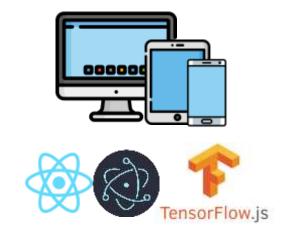
https://gs.statcounter.com/browser-market-share#m



Front-End 웹 브라우저에서 작동하는 유일한 프로그래밍 언어



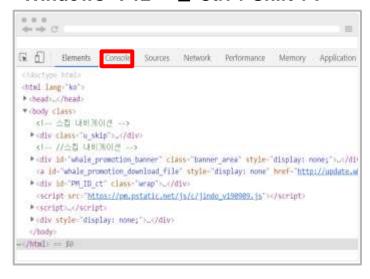
Back-End 서버 측 프로그래밍에서도 사용 되고 있음



Cross Platform 데스크톱, 웹, 앱, 머신러닝 등 플랫폼을 넘나들며 개발

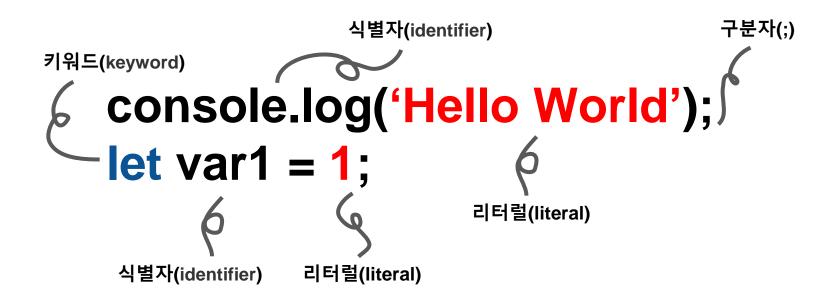
#### ● 크롬 개발자 도구

#### ○ Windows F12 또는 Ctrl + Shift + I





● JavaScript 문법 ∰



JavaScript 적용



○ <script> 태그

○ 외부 script 파일 (.js)

o defer vs async

