

<h1>

D.com

WEB - 스타디

</h1>

#front-end편

4강.Javascript

-웹에 활기를 불어넣어봅시다!



오늘의 학습 목표!

1. Javascript 기본 문법을 알아봅시다!

1. Javascript를 이용하여 웹을 제어해 봅시다!

강의 자료 개요

별표 친 페이지는 반드시 읽어주세요!



이론

이론 내용입니다. 가벼운 마음으로 읽어주세요!

실습


百聞이 不如一打!
직접 따라해보세요!

읽을거리

읽어두면 배운 것을 심층적으로 이해하는데 있어 도움이 되는
내용입니다! 여유가 있으면 읽어주세요!



Javascript – Introduction



JS 기초!



JavaScript



Html/Css를 이용하여
웹의 겉모습을 만들었다
면 Javascript를 이용
하여 웹에 기능을 넣
을 수 있습니다.

JS 기초!



**Javascript를 로드하는 방법은
크게 3가지 있습니다. (CSS와 비슷!)**

1. 인라인 스타일 (Inline Style)
2. `<Script> </Script>`
3. JS 파일 호출

JS 기초!

1. 인라인 스타일 (Inline Style)

HTML Tag내에 Javascript를 선언하는 방식입니다.
태그의 역할을 바로 알아볼 수 있다는 장점이 있지만,
코드가 쉽게 **복잡 해진다는 단점**이 있어
잘 사용되지 않는 방식입니다.

```
<input type="button" onclick="alert('Hello world');" value="button" />
```

JS 기초!

2.<Script> </Script>

CSS를 선언할 때 사용하는 <Style>처럼
HTML 문서내에 Javascript를 사용할 수 있습니다.

```
<script>  
alert('hello world');  
</script>
```


JS 기초!

3. JS 파일 호출

외부 JS 파일로 분리하여 HTML에 로드하는 방식입니다.

앞의 두가지보다

관리 유용, 속도(캐싱), 보안 향상 등의 장점이 있어

가장 선호되는 방식입니다!

```
<script src="hello.js"></script>
```

읽을 거리



JS 파일 호출하는 <Script>는 어디에 위치시키나요?

Front-End 개발자 인터뷰에서 자주 출제되는
질문 중 하나입니다.

<https://dailydev.tistory.com/15>

Javascript – Syntax



JS 기본 문법!



이번 강의에서는
“변수 선언, 출력, 데이터형, 조건문, 반복문, 함수”에
대해서 알아보니다.

다른 언어(Ex. C, Python)과 크게 다르지 않으니 금방 배울 수 있습니다!

Javascript ES6 버전 기반으로 설명합니다.

(후에 Back-end 스터디에서도 ES6 문법을 사용합니다.)

JS 기본 문법!

변수 선언

```
let a;  
a = 3  
let b = 'web';  
const pi = 3.14;
```

변수는 Let, Const로 선언할 수 있습니다.
Const는 한번 선언한 값에 대하여 변경할 수 없습니다.(상수)

읽을 거리

Scope

JS에서는 Scope 개념이 중요합니다!

예를 들어, Let은 Block-Scope로
{ }중괄호 안에서만 변수가 유효합니다!

<https://medium.com/@khwsc1/%EB%B2%88%EC%97%AD-%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8-%EC%8A%A4%EC%BD%94%ED%94%84%EC%99%80-%ED%81%B4%EB%A1%9C%EC%A0%80-javascript-scope-and-closures-8d402c976d19>

읽을 거리



다른 곳에서는 변수 선언으로
Var을 사용 하던대요?

Var은 Javascript ES5의 문법입니다.
ES6에서는 Var의 위험성 때문에 사용을
‘지양’합니다.

JS 기본 문법!

주석

```
// 한 줄 주석
```

```
/*  
여러 줄  
주석  
*/
```


JS 기본 문법!

자료형

```
let num = 1;  
let str1 = 'string';  
let str2 = "string";  
let bool = true;
```

대표적으로 Boolean, Number, String이 있습니다.
(Python과 유사)

JS 기본 문법!



출력

```
console.log(1);  
console.log(num);
```

콘솔창에 값을 출력할 수 있습니다.

JS 기본 문법!

산술 연산

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
%	나머지(나누기의)
++	증가
--	감소

타 언어와 유사한
산술 연산을 가지고 있습니다.

읽을 거리



**참고로 Javascript에서 세미콜론은
‘선택’입니다!**

<https://okky.kr/article/448378>

JS 기본 문법!

형 변환

```
let num1 = 1;  
let str1 = String(num1);  
  
let str2 = '1';  
let num2 = Number(str2);
```

내장 함수인 String, Number를 사용합니다.

JS 기본 문법!

형 확인

```
console.log(typeof "1")  
console.log(typeof 1)
```

typeof를 사용하여 자료형을 확인할 수 있습니다.

JS 기본 문법!

조건문

```
let grade = 85;
if(grade>=90){
  console.log('A');
}
else if(grade>=80){
  console.log('B')
}
else {
  console.log("C")
}
```

C언어와 유사한 If/else If/Else 문법을 사용합니다..

JS 기본 문법!

조건문

```
switch(num%2){  
  case 0:  
    console.log('짝수');  
    break;  
  case 1:  
    console.log('홀수');  
    break;  
  default:  
    console.log('숫자가 아닙니다.');
```

```
    break;  
}
```

Switch문 또한 사용 가능합니다.

JS 기본 문법!

반복문

```
for(let i = 0; i < 10; i++){  
  console.log(i);  
}
```

For문도 C언어와 같습니다.

JS 기본 문법!

반복문

```
while(i<10){  
  console.log(i);  
  i++;  
}
```

While문도 C언어와 같습니다.

JS 기본 문법!

함수

```
// 함수 선언식(function declaration)
function sum(a,b){
  return a+b;
}

// 함수 표현식(function expression)
const sum2=function(a,b){
  return a+b;
}
```

함수는 다음과 같이 선언합니다.

읽을 거리



함수 선언식 VS 함수 표현식

<https://joshua1988.github.io/web-development/javascript/function-expressions-vs-declarations/>

JS 기본 문법!

배열

```
let arr = [1,2,'HTML',true]; // 배열 선언
console.log(arr.length); // 배열 길이 출력
console.log(arr[0]); // 배열 0번째 요소 출력
arr[1] = '3' // 배열 2번째 요소 값 수정
arr.push(4); // 배열 마지막 요소 뒤에 값 추가1
arr[arr.length] = 5; // 배열 마지막 요소 뒤에 값
// 추가2
console.log(arr[20]); // undefined 값 출력
```

JS 배열 문법입니다. (Python과 C를 합친 스타일입니다.)

JS 기본 문법!



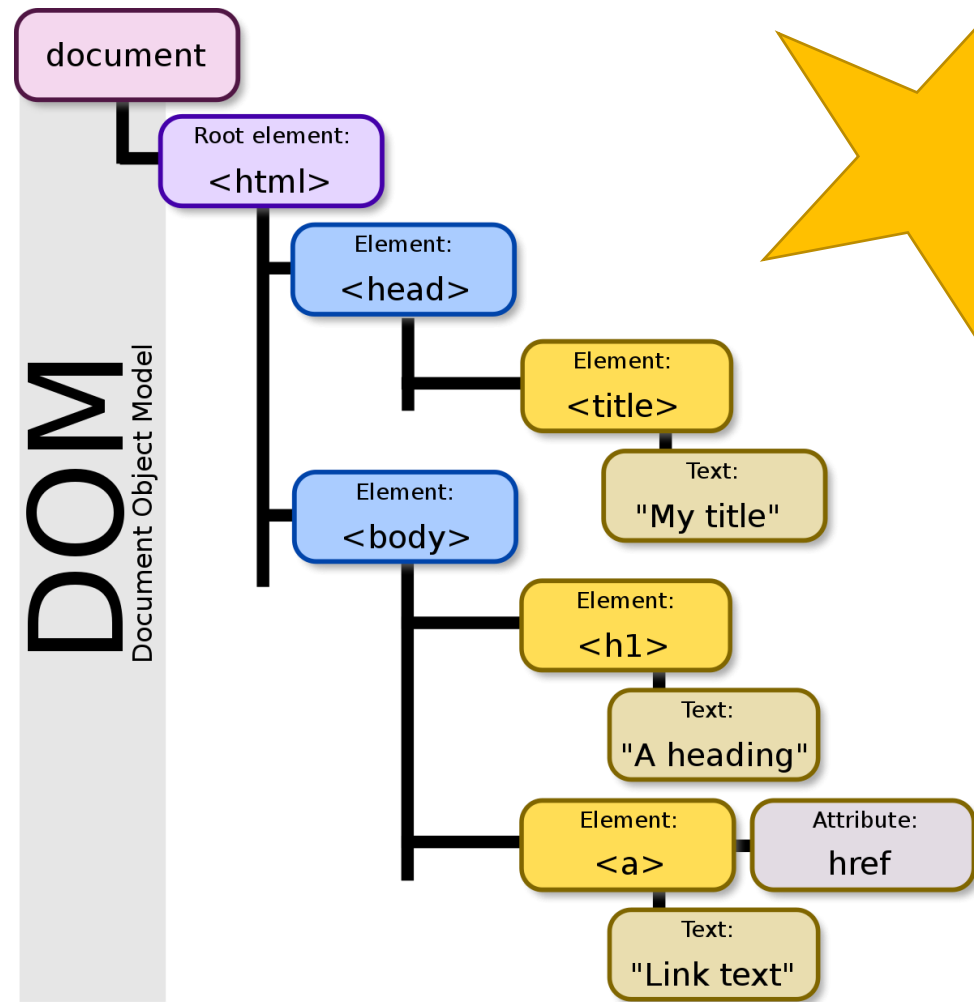
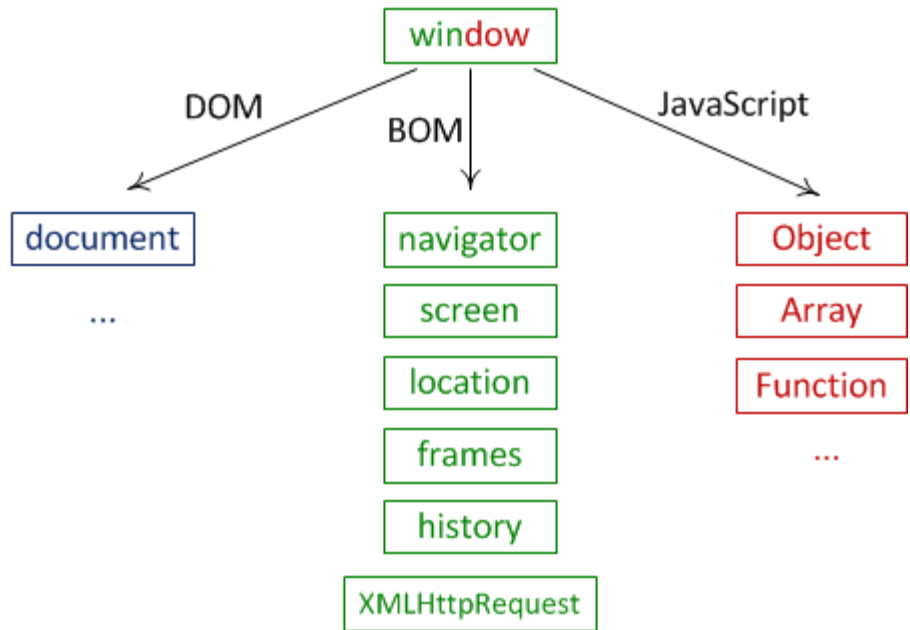
이외에도 JS에는 Object, Class, Module,
Async 등등의 문법이 존재합니다!

이들 문법은 추후 JS 심화 강의에서 다룹니다!

Javascript – Object Model



JS Object Model



웹 브라우저의 구성 요소는 하나하나가 객체화 되어있습니다.

읽을 거리



DOM, BOM이 무엇인가요?

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

각각 문서 객체 모델, 브라우저 객체 모델을 의미하며
Javascript를 이용하여 이들을 제어할 수 있습니다!
즉, 웹사이트에 동적인 기능을 추가할 수 있습니다.

JS DOM 제어



DOM을 제어한다. = HTML을 제어한다.

그렇다면 어떻게 JS로 DOM을 제어할 수 있을까요?

CSS가 꾸미기 전 선택자로 원하는 Element를 지정하듯이,
JS도 우선 원하는 Object를 **지정(Select)**하여야 합니다.
그리고 지정한 Object의 **속성(Property)**를 **변경**하거나
Event를 줄 수 있습니다!

JS DOM 제어

원하는 Object를 지정하는 방법에는 여러가지가 있지만,
대표적인 1가지의 방법을 알아보시다!

`document.getElementById`

JS DOM 제어

document.getElementById

```
<ul>
<li>HTML</li>
<li id="active">CSS</li>
<li>JavaScript</li>
</ul>
<script>
let li = document.getElementById('active');
li.style.color='red';
</script>
```

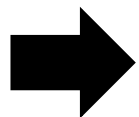
‘Active’ Id를 가진 요소를 찾아서 변수 li에 담는다.
그리고 그 요소를 폰트 색상을 Red로 한다.

JS DOM 제어



좀더 자세히 알아봅시다!

```
let li = document.getElementById('active');  
  
li.style.color='red';
```



Document Object 안에서 active라는 id를 가진 Element를 가져와서 li 변수에 넣는다!
그리고 li변수에 담긴 Element의 Style Object의 color Property를 red로 바꾼다!

#웹을 이루는 구성요소 하나하나는 객체이다!

JS DOM 제어

```
let li = document.getElementById('active');  
li.innerText = 'CSS3'
```

innerText Property를 통해 Element의 Text를 변경할 수도 있습니다!

Javascript – Event



JS Event

우리는 웹을 이용하면서 마우스 클릭 등 수많은 Event들을 발생시킵니다!
JS를 이용하여 Event를 처리하는 방법을 알아보니다!

JS Event



Event를 처리하려면
이벤트가 일어날 객체(**Event target**),
이벤트의 종류(**Event type**),
이벤트가 발생했을 때 동작할 코드(**Event handler**)
가 필요합니다!

읽을 거리



수많은 Event Type을 정리한 글입니다.

<https://developer.mozilla.org/en-US/docs/Web/Events>

JS Event



Event를 등록하는 방법에는 3가지가 있습니다.

1. Inline

2. Property Listener

3. `addEventListener()`

본 자료에서는 2, 3번을 알아보겠습니다!

JS Event

```
<input type="button" id="target" value="button"/>
<script>
let t = document.getElementById('target');
t.onclick = function(){
  alert('Hello world');
}
</script>
```

Property Listener 방법입니다.
Event Target의 Property로 이벤트를 등록합니다!

JS Event

```
let t = document.getElementById('target');
t.addEventListener('click', alert_one);
function alert_one() {
  alert(1);
};
t.addEventListener('click', alert_two)
function alert_two() {
  alert(2);
};
```

addEventListener() 방법입니다.

첫번째 파라미터로 Event Type, 두번째 파라미터로 Event Handler를 지정합니다.

이는 여러 개의 Event Handler를 등록할 수 있다는 장점이 있습니다.

가장 선호되는 방법입니다.

읽을 거리



Event 객체

이벤트가 실행된 맥락의 정보가 필요할 때 유용합니다!

Ex) 현재 이벤트를 처리 중인 Element 조회

<https://blog.sonim1.com/152>

오늘의 과제!



```
<!DOCTYPE html>
<html>

<head>
<title>webmaster</title>
</head>

<body>
<h1>I was a Web beginner.</h1>
<input type="button" id="target" value="Study!" />
</body>

</html>
```

I was a Web beginner.

Study!

다음과 같은 HTML 페이지가 주어집니다.

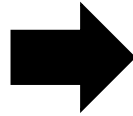
Study! 버튼을 누르면 <h1>의 텍스트가 변경되도록 코드를 작성해보세요! (문구는 자유!)

과제 Example!



I was a Web beginner.

Study!



Now, I am a Web Master!

Study!

과제 제출!



개인 Git의
“2019_Summer_Web_Study”에

작업한 파일들을 “4.Master” 폴더로 모아
Front-End-Assignment 폴더 안에 Push

다음 시간!

Nodejs를 이용한 Back-End 개발을 배워봅시다! 와~!



**Thank
You!**