


Hello, 

SeSAC 용산 1기 웹 풀스택 과정

WITH 팀 리처드



JS





귀여운 — 형용사



미니언이 — 명사



춤춘다 — 동사



Javascript

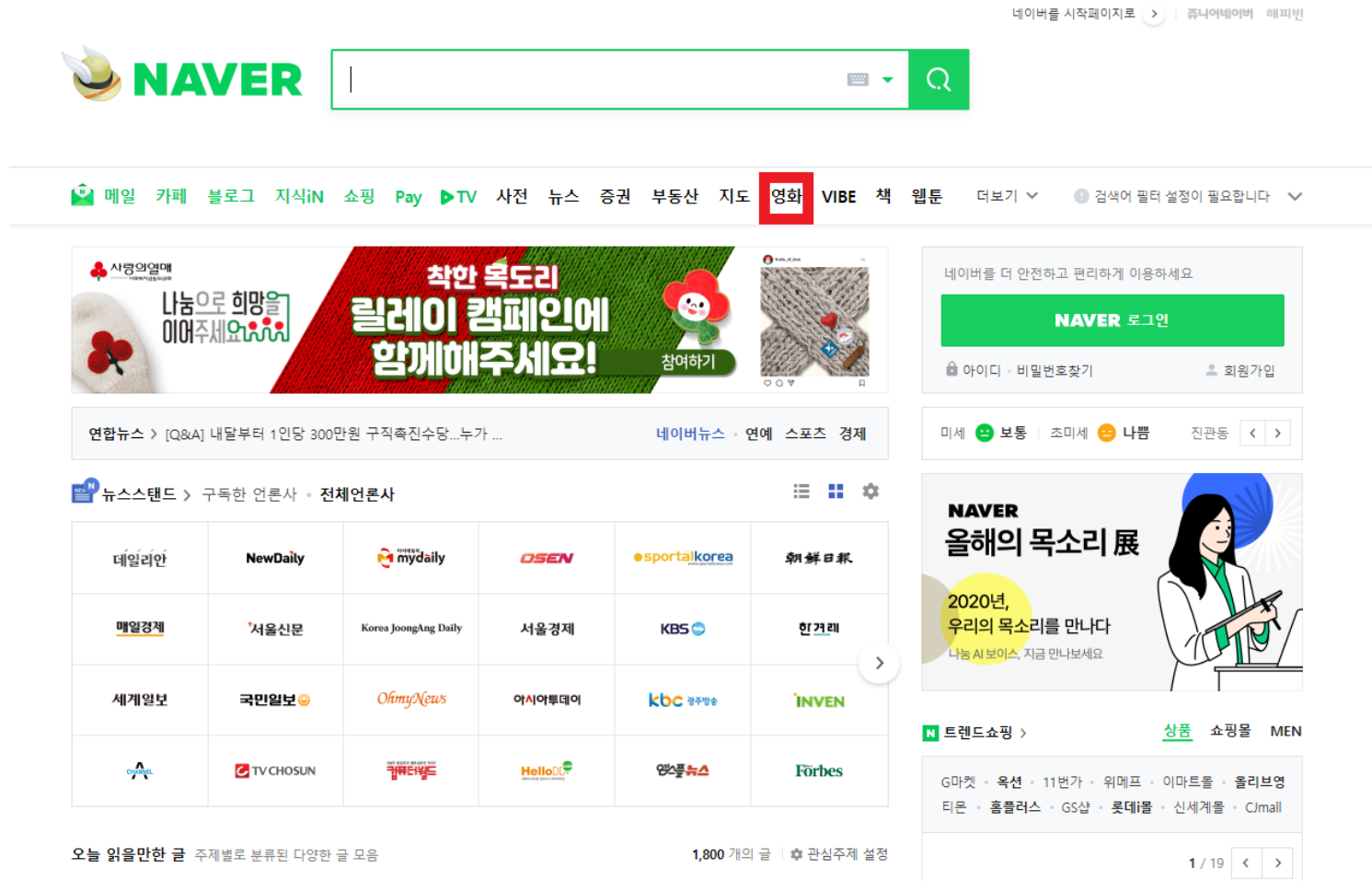
생동감



JavaScript

웹 페이지에서 복잡한 기능을 구현할 수 있도록 하는
스크립팅 언어 또는 프로그래밍 언어

Javascript 사용



동적기능

동적처리

이벤트 처리

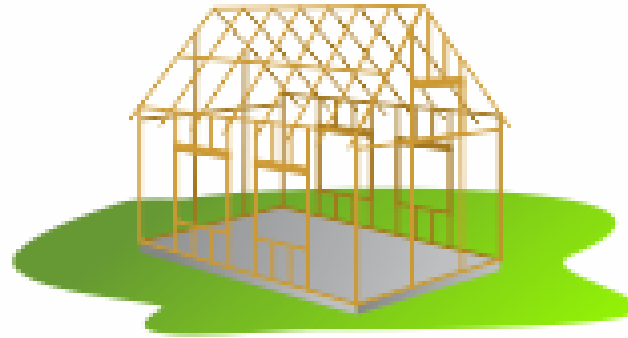
슬라이드 메뉴

...

HTML-CSS-JS



The land
"Web server"



The house frame
"HTML"



The bricks, tiles and paint
"CSS"

[출처] <https://www.deconetwork.com/blog/skinning-your-deconetwork-website/>

Javascript 사용법

```
<script>  
    // Javascript 코드 작성  
</script>
```

<head> </head> 안에서도 사용이 가능하며,

< body></body> 안에서도 사용이 가능하다.

Javascript 사용해보기



- **console.log()**
 - 브라우저의 개발자 도구 > 콘솔 에서 확인 가능
- **alert()**
 - 브라우저가 열렸을 때, 그 내용을 알림창으로 보여줌

Javascript 변수

Javascript 변수

Identifier

Memory

myNumber



Address	Value
0012CCGWH80	23

Javascript 변수

1) 선언



JavaScript ▾

```
let 변수이름;
```

```
var 변수이름;
```

2) 할당

```
const 변수이름 = 값;
```

```
let 변수이름 = 값;
```

```
var 변수이름 = 값;
```

var & let & const

```
var 변수이름;
```

- 1) 선언 단계와 초기화가 동시에 이루어지며 아무것도 할당하지 않으면 자동으로 **undefined** 가 할당
- 2) **중복 선언** 가능. **재선언** 가능

var & let & const

```
let 변수이름;
```

- 1) 변수 **중복 선언이 불가능**하지만, **재할당 가능**
- 2) var 과 마찬가지로 선언을 하지 않으면 자동으로 undefined가 들어간다.

var & let & const

```
const 변수이름 = 값;
```

- 1) 초반에 선언할 때 반드시 초기화를 동시에 진행해야 한다.
- 2) 재선언 불가능, 재할당 불가능

Javascript 자료형

언어 타입

강한 타입 언어

타입 검사를 통과하지 못한다면 실행 자체가 안 된다.

String, int, double 등처럼 타입을 1종류로 명확히 지정

약한 타입 언어

런타임에서 타입 오류를 만나더라도 실행을 막지 않는다.

타입이 여러 종류인 값들이 상관없이 지정된다.

언어 타입

강한 타입 언어 (Strong)	약한 타입 언어 (Weak)
Java, C, C++, C#	Javascript, Python

Javascript는 느슨한 언어!

- Javascript는 데이터 종류와 관계 없이 var, let, const 키워드로 변수를 선언하고 사용함! => 약한 타입 언어
- 강한 타입 언어들은 변수를 선언할 때 명확하게 타입을 1종류만 지정함!
 - ex. JAVA, C, C++

Javascript 자료형

Primitive 자료형

Object 자료형

- 기본형 (Primitive)

- string
- number
- boolean
- null
- undefined

- 객체 (Object)

- 기본형이 아닌 것은 모두 객체

Primitive 자료형



Boolean

참(true), 거짓(false) 둘 중 하나의 값을 갖는 요소

Number

숫자형으로 정수와 부동 소수점, 무한대 및 NaN(숫자 아님)

String

문자 데이터를 나타낼 때 사용

Primitive 자료형



Null

빈 값을 뜻하는 null 타입
타입은 존재하지만 값 존재 X

Undefined

값 X 타입 X

자료형 확인하기

- typeof()

```
typeof('문자')  
typeof(245)  
typeof(function() { } );  
typeof(true)  
typeof({})  
typeof([])  
typeof(NaN)|  
typeof(null)
```

형 변환 - 문자열로

1. String()

```
String(true);
```

2. toString()

```
a.toString();  
(false).toString();
```


형 변환 - 정수로

1. Number()

```
Number(true)    // 1  
Number('10')
```

2. parseInt()

```
parseInt('10');
```

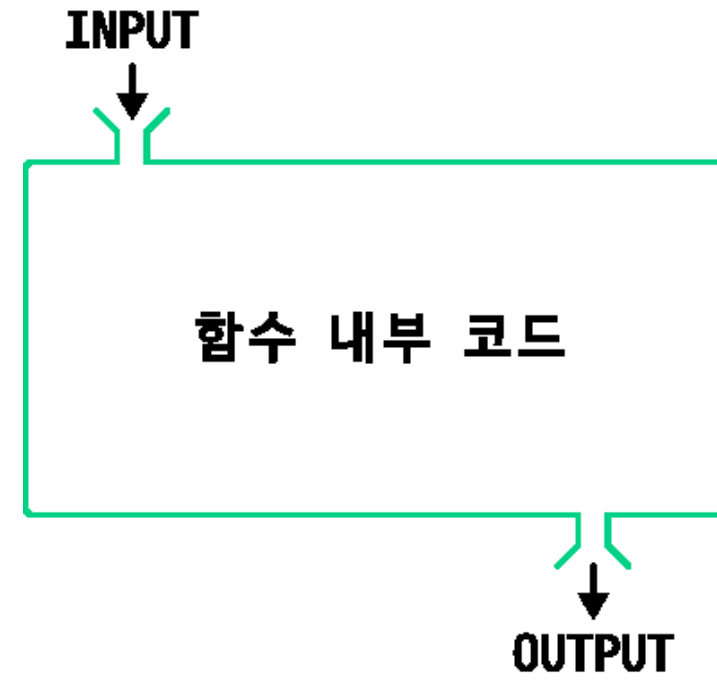
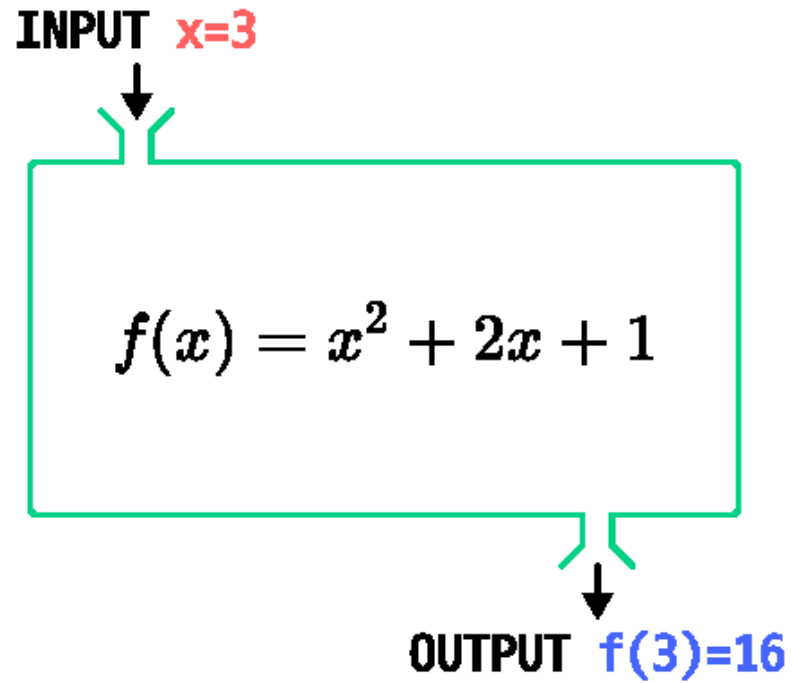
Javascript 연산자

연산자

- 대입 연산자: =
- 비교 연산자: ==, !=, ===, !==, >, >=, <, <=
- 산술 연산자: +, -, *, /
- 논리 연산자: !, &&, ||

Javascript 함수

함수



Javascript 함수

```
function hello() { }
```

특정 작업을 수행하기 위해 독립적으로 설계된 **코드 집합!**

Javascript 함수

```
function hello() { }
```

→ Keyword

Javascript 함수

```
function hello() { }
```

Name. 함수 이름

일반적으로 카멜 표기법 이용 (helloWorld)

Javascript 함수

```
function hello() { }
```

Parameter

함수를 선언할 때 **매개변수(인자)**를 받을 것.

Javascript 함수

```
function hello() { }
```

Body

함수가 실행되는 Scope 라고도 한다.

함수 선언 방식

명시적 함수 선언

```
function hello() { }
```

함수 표현식

```
const hello = function() {}
```

실습 11. Javascript 실습

1. no_1, no_2, no_3 이라는 3개의 변수를 선언하고, 각각 숫자 할당하기
2. no_1, no_2, no_3의 합계를 alert로 출력하는 add 함수 만들기
3. no_1과 no_2의 차이를 alert로 출력하는 sub 함수 만들기
4. no_1에서 no_2를 나눈 값을 alert로 출력하는 divide 함수 만들기
5. no_1과 no_2의 곱을 alert로 출력하는 mul 함수 만들기

Javascript 조건문

Javascript 조건문

특정 조건 만족 시 (조건이 참인 경우) 실행하는 명령의 집합

특정한 조건 속에서 작업을 수행하고 싶을 때 사용

if

switch

Javascript if문

만약 ~~라면

```
if ( 조건1 ) {  
    // 조건1이 참이라면 실행  
}
```

Javascript if문

```
if ( 조건1 ) {  
    // 조건1이 참이라면 실행  
} else if ( 조건2 ) {  
    // 조건1이 참이 아니고 조건2가 참이라면 실행  
}
```


Javascript if문

```
if ( 조건1 ) {  
    // 조건1이 참이라면 실행  
} else if ( 조건2 ) {  
    // 조건1이 참이 아니고 조건2가 참이라면 실행  
} else {  
    // 조건 1과 2가 모두 참이 아닐 때 실행  
}
```

Javascript if문

비교연산자

$a == b$: a와 b가 동일하면 참

$a != b$: a와 b가 동일하지 않으면 참

$a < b$: a가 b보다 작으면 (b가 a보다 크면) 참

$a <= b$: a가 b보다 작거나 같으면 참

논리연산자

$a \&\& b$: a AND b. a 그리고 b

$a \parallel b$: a OR b. a 또는 b

Javascript if문 중첩

```
if ( 조건1 ) {  
    if ( 조건2 ) {  
        //실행  
    } else {  
        //실행2  
    }  
}
```

Javascript switch문

```
switch ( 변수 ){  
    case 값1:  
        // 변수와 값1이 일치하면 실행  
        break;  
    case 값2:  
        // 변수와 값2가 일치하면 실행  
        break;  
    default: //다 아닐 때  
        break;
```

Javascript 삼항 연산자

조건 ? 참일 때의 실행 : 거짓일 때의 실행

EX)

```
var n = 3;
```

```
n > 0 ? alert("큘") : alert("작음");
```

Javascript 반복문

Javascript 반복문

똑같은 명령을 일정 횟수만큼 반복해 수행하도록 하는 실행문

for

while

for / in

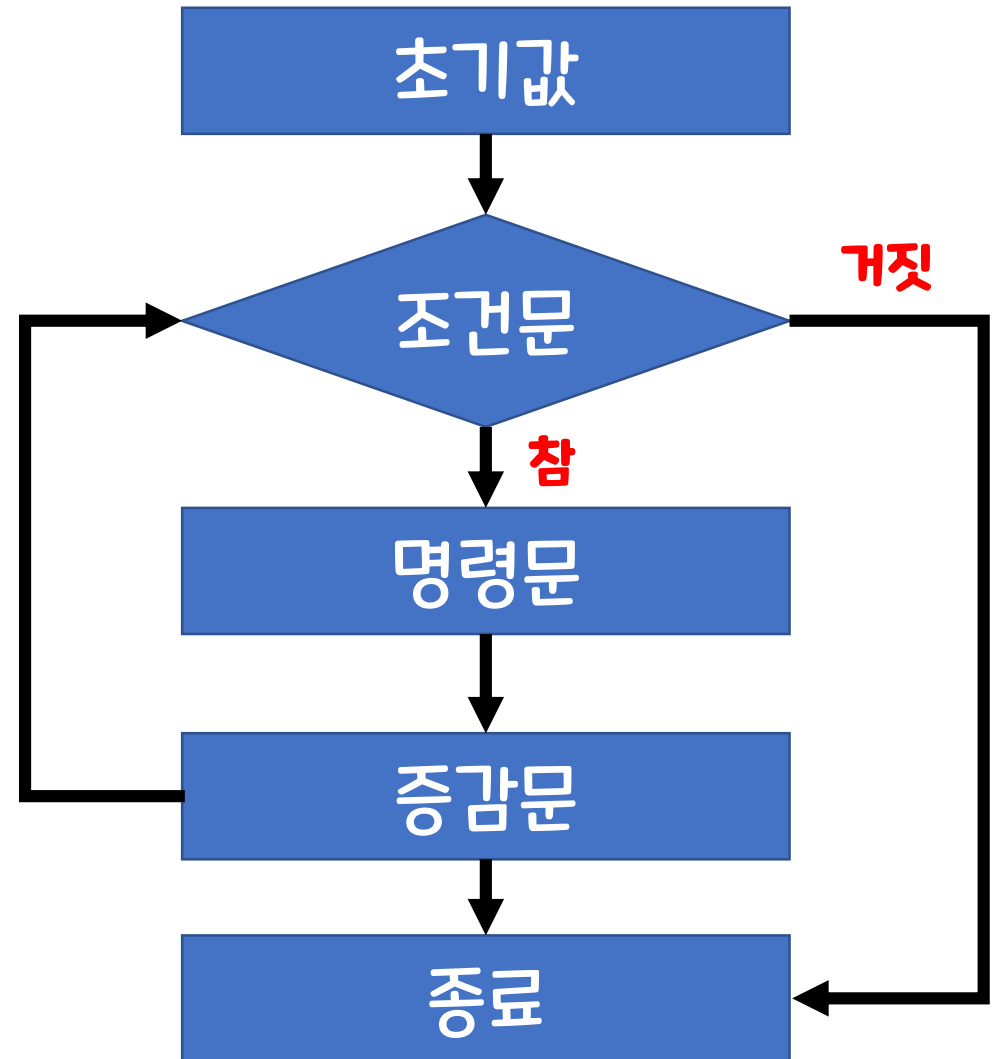
do / while

Javascript for문

```
for ( 초기문; 조건문; 증감문 ){  
    // 조건문의 결과가 참인 경우 반복적으로 실행  
    // 명령문  
}
```


Javascript for문

```
for ( 초기값; 조건문; 증감문 ){  
    // 조건문의 결과가 참인 경우 반복적으로 실행  
    // 명령문  
}
```



Javascript while문

```
while ( 조건문 ) {  
    // 조건문이 참인 경우 반복적으로 실행하는 명령문  
}
```

※ 주의사항

- while문의 경우 명령문에서 조건문의 결과를 바꾸지 않으면 무한 루프에 빠질 수 있다.

Javascript for & while

```
var i = 0;  
while ( i < 10 ){  
    console.log( i );  
    i++;  
}
```

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

실습 12. 구구단 만들기

2*1 = 2

2*2 = 4

2*3 = 6

2*4 = 8

2*5 = 10

2*6 = 12

2*7 = 14

2*8 = 16

2*9 = 18

2*10 = 20

3*1 = 3

3*2 = 6

3*3 = 9

3*4 = 12

3*5 = 15

3*6 = 18

3*7 = 21

3*8 = 24

3*9 = 27

3*10 = 30

4*1 = 4

4*2 = 8

4*3 = 12

4*4 = 16

4*5 = 20

콘솔 창에 구구단 띄우기

힌트 1. <script> 에서 콘솔에 글 띄우는 방법

`console.log(띄울 내용);`

힌트 2. 변수와 글자를 연결하는 방법

`var i = 0;`

`console.log(i + “입니다.”);`

실습 13. 합계 구하기

정수 n 이 있을 때 0부터 n 까지의 수 중
2또는 3의 배수의 총합을 구하시오.
(단, n 은 $0 < n < 100$)

Ex) n 이 10이라면 $2 + 3 + 4 + 6 + 8 + 9 + 10$