

JavaScript

◆ 자료형과 변수

정수아

Contents

01 자료형

02 변수

03 증가/감소 연산자

04 불리언

05 비교 연산

06 조건 연산



01

자료형

자료형

❖ 데이터

- 컴퓨터 프로그램에 저장된 정보
- 예) 이름, 나이, 주소, 성별 등 모든 것이 데이터

❖ 데이터 타입

- 자료형
- 자바스크립트 언어로 다룰 수 있는 데이터의 종류

자료형

❖ 자바스크립트의 기본 자료형

- 숫자 타입

```
5;
```

- 문자열 타입

```
"안녕하세요. 저는 문자열입니다.";
```

- 논리 타입

```
// 참(true) 또는 거짓(false)이 될 수 있는 값  
true;
```

숫자와 연산자

❖ 덧셈, 뺄셈, 곱셈, 나눗셈 같은 기본 사칙 연산 수행

- 연산자(operator)라는 기호 +, -, *, /가 사용

```
> 12345 + 56789;
```

덧셈

```
< 69134
```

```
> 22 + 33 + 44;
```

```
< 99
```

```
> 1000 - 17;
```

뺄셈

```
< 983
```

```
> 123 * 456;
```

곱셈

```
< 56088
```

```
> 12345 / 250;
```

나눗셈

```
< 49.38
```

```
> 1234 + 57 * 3 - 31 / 4;
```

복잡한 계산

```
< 1397.25
```

숫자와 연산자

❖ 연산자 우선순위

- 곱셈, 나눗셈 > 덧셈, 뺄셈

$$\begin{array}{c} 1234 + 57 * 3 - 31 / 4 \\ \downarrow \\ 1234 + 171 - 31 / 4 \\ \downarrow \\ 1234 + 171 - 7.75 \\ \downarrow \\ 1405 - 7.75 \\ \downarrow \\ 1397.25 \end{array}$$



02

변수

변수

❖ 변수

- 데이터를 저장하는 공간의 이름
- C나 Java와 달리, 변수에 데이터 타입을 정하지 않음

❖ 변수 선언 방법

- var 키워드
- let 키워드
- const 키워드

변수

❖ var 키워드로 변수 선언

```
var 변수이름;
```

```
var score;  
var year, month, day;
```

❖ var 키워드 없이 변수 선언 가능

```
age = 21;      // var 없이 변수 age 선언
```

- 만약 age가 이미 선언된 변수라면, 새로운 변수를 생성하지 않고 이미 존재하는 age에 21 저장

변수

❖ **let, const** 키워드로 변수 선언

```
let 변수이름;  
const 변수이름;
```

```
let score;  
const year, month, day;
```

변수

❖ var, let, const 비교

키워드	설명
var	<ul style="list-style-type: none">• 이미 존재하는 변수와 같은 이름으로 변수 선언 시, 에러 발생하지 않음• 변수 선언 전에 변수 사용 시, 에러 발생하지 않음• 변수 생성 시, 초기값을 설정하지 않아도 됨• 변수 값 재할당 가능
let	<ul style="list-style-type: none">• 이미 존재하는 변수와 같은 이름의 변수 선언 시, 에러 발생• 변수 선언 전에 변수 사용 시, 에러 발생• 변수 생성 시, 초기값을 설정하지 않아도 됨• 변수 값 재할당 가능
const	<ul style="list-style-type: none">• 이미 존재하는 변수와 같은 이름의 변수 선언 시, 에러 발생• 변수 선언 전에 변수 사용 시, 에러 발생• 변수 생성 시, 반드시 초기값을 설정해야 함• 변수 값 재할당 불가능

변수 호이스팅(Hoisting)

❖ var 키워드를 사용할 경우

- 변수 호이스팅 가능

```
age = 5;  
var age;  
console.log(age);
```

- 실행 결과

```
5
```

변수 호이스팅(Hoisting)

❖ let 키워드를 사용할 경우

```
let age = 5;  
let phone;  
  
phone = 1234;  
  
console.log(age);  
console.log(phone);
```

• 실행 결과

```
5  
1234
```

변수 호이스팅(Hoisting)

❖ let 키워드를 사용할 경우

- 변수 호이스팅 불가능

```
age = 5;  
let age;  
  
console.log(age);
```

- 실행 결과

```
Uncaught ReferenceError: Cannot access 'age' before  
initialization
```

변수 스코프(Scope)

❖ var - 함수 스코프

- if문 내에서 var로 선언한 변수는 접근 가능

```
if(true) {  
  var age = 5;  
}  
  
console.log(age);
```

- 실행 결과

5

변수 스코프(Scope)

❖ var - 함수 스코프

- 함수 내에서 var로 선언한 변수는 해당 함수 내에서만 접근 가능

```
function person(){  
  var age = 5;  
}  
  
console.log(age);
```

- 실행 결과

Uncaught ReferenceError: age is not defined

변수 스코프(Scope)

❖ let, const - 블록 스코프

- let, const로 선언한 변수

```
if(true) {  
  let age = 5;  
}  
  
console.log(age);
```

- 실행 결과

Uncaught ReferenceError: age is not defined

초기화

❖ 변수 값 초기화

- 변수에 값을 할당하는 것
- 변수 나이의 값을 12로 초기화

```
>> var 나이 = 12;
```

- 변수 값을 바꾸고 싶으면 다시 등호를 사용
 - 이미 '나이'라는 변수가 존재하기 때문에 var 키워드를 사용하지 않음

```
>> 나이 = 13;  
<< 13
```

초기화

❖ 변수 타입과 값

- 자바스크립트에는 변수의 타입이 없음

```
var score      // 정상적인 변수 선언  
int score      // 에러. 변수 타입을 사용하지 않음
```

- 따라서, 아무 값이나 저장할 수 있음

```
score = 66;     // 정수 저장 가능  
score = "high"  // 문자열 저장 가능
```

[실습] 변수 응용

- ❖ ‘사람수’라는 변수를 만들고, 1+3을 할당
- ❖ ‘사탕개수’라는 변수를 만들고, 8을 할당
- ❖ 사탕개수 / 사람수 $\rightarrow 8 / 4$

```
>> var 사람수 = 1 + 3;  
    var 사탕개수 = 8;  
    사탕개수 / 사람수;  
<< 2
```

[실습] 계산하며 변수 만들기

❖ 한 시간은 몇 초일까요?

```
>> var 일분은몇초 = 60;  
    var 한시간은몇분 = 60;  
    var 한시간은몇초 = 일분은몇초 * 한시간은몇분;  
    한시간은몇초;  
<< 3600
```

❖ 하루는 몇 초일까요?

```
>> var 하루는몇시간 = 24;  
    var 하루는몇초 = 한시간은몇초 * 하루는몇시간;  
    하루는몇초;  
<< 86400
```

[실습] 계산하며 변수 만들기

❖ 일 년은 몇 초일까요?

```
>> var 일년은며칠 = 365;  
    var 일년은몇초 = 하루는몇초 * 일년은며칠;  
    일년은몇초;  
<< 31536000
```

❖ 나이는 몇 초일까요?

```
>> var 나이 = 29;  
    나이 * 일년은몇초;  
<< 914544000
```



03

증가/감소 연산자

증가와 감소

❖ 1씩 늘어나는 것은 증가, 1씩 줄어드는 것은 감소

❖ 증가 연산자(++), 감소 연산자(--)

❖ 전위 연산자

- 연산자를 변수 앞에 작성
- 증가 또는 감소한 후의 값이 반환

```
> var 하이파이브 = 0;  
   ++하이파이브;
```

```
< 1
```

```
> ++하이파이브;
```

```
< 2
```

```
> --하이파이브;
```

```
< 1
```

증가와 감소

❖ 후위 연산자

- 연산자를 변수 뒤에 작성
- 증가 또는 감소하기 전의 값이 반환

```
> 하이파이브 = 0;  
   하이파이브++;
```

```
< 0
```

```
> 하이파이브++;
```

```
< 1
```

```
> 하이파이브;
```

```
< 2
```

증가와 감소

❖ 증감 연산자

연산자	내용
++a	a를 1 증가한 후, a 값 사용
--a	a를 1 감소한 후, a 값 사용
a++	a를 사용한 후, a 값 1 증가
a--	a를 사용한 후, a 값 1 감소

대입 연산자

❖ 오른쪽 식의 결과를 왼쪽에 있는 변수에 대입

연산자	내용
$a = b$	b 값을 a에 대입
$a += b$	$a = a + b$ 와 동일
$a -= b$	$a = a - b$ 와 동일
$a *= b$	$a = a * b$ 와 동일
$a /= b$	$a = a / b$ 와 동일
$a \% = b$	$a = a \% b$ 와 동일

대입 연산자

❖ $x += 5$ 는 $x = x + 5$ 와 같은 의미

❖ $x -= 9$ 는 $x = x - 9$ 와 같은 의미

```
>> var 점수 = 10;  
    점수 += 7;  
<< 17
```

```
>> 점수 -= 3;  
<< 14
```

[실습] *=와 /= 연산자

```
var 풍선 = 100;  
풍선 *= 2;  
???
```

```
var 풍선 = 100;  
풍선 /= 4;  
???
```



04

불리언

논리 연산자

- ❖ 참이나 거짓, 둘 중 하나의 값
- ❖ 참(TRUE), 거짓(FALSE)

논리 연산자

❖ 불리언 연산자

- 숫자를 산술 연산자(+, -, *, / 등)와 함께 사용하듯이 불리언 값(TRUE, FALSE)은 불리언 연산자와 함께 사용

❖ 불리언 연산자 종류

&&	AND
	OR
!	NOT

&& (AND)

❖ 두 개의 불리언 값이 모두 참인지 확인

A	B	A && B
참	참	참
참	거짓	거짓
거짓	거짓	거짓
거짓	참	거짓

&& (AND)

❖ 하나는 true, 다른 하나는 false인 경우

```
>> var 샤워하기 = true;  
    var 가방챙기기 = false;  
    샤워하기 && 가방챙기기;  
<< false
```

❖ 둘 다 true인 경우

```
>> var 샤워하기 = true;  
    var 가방챙기기 = true;  
    샤워하기 && 가방챙기기;  
<< true
```

|| (OR)

❖ 두 개의 불리언 값이 하나라도 true인지 확인

A	B	A B
참	참	참
참	거짓	참
거짓	거짓	거짓
거짓	참	참

|| (OR)

❖ 두 개의 불리언 값이 하나라도 true인지 확인

```
>> var 사과챙기기 = true;  
    var 오렌지챙기기 = false;  
    사과챙기기 || 오렌지챙기기;  
<< true
```

! (NOT)

❖ false는 true로, true는 false로 바꿈

A	!A
참	거짓
거짓	참

! (NOT)

❖ false는 true로, true는 false로 바꿈

```
>> var 주말인가요 = true;  
    var 샤워해야하나요 = !주말인가요;  
    샤워해야하나요;  
<< false
```

논리 연산자 결합

❖ 논리 연산자 우선 순위

- $\&\&$ > \parallel

```
>> var 주말인가요 = false;
    var 샤워하기 = true;
    var 사과챙기기 = false;
    var 오렌지챙기기 = true;
    var 나가야하나요 = !주말인가요 && 샤워하기 && (사과챙기기 || 오렌지챙기기);
    나가야하나요
<< true
```


불리언으로 숫자 비교

❖ 불리언 값은 간단한 숫자 문제에 활용할 수 있음

- 초과/이상 연산자
- 이하/미만 연산자
- 삼중 등호
- 이중 등호



05

비교 연산

비교 연산자

❖ 비교 연산

- 두 값을 비교한 결과가 true 또는 false인 연산

연산자	내용
$a < b$	a가 b보다 작으면 true
$a > b$	a가 b보다 크면 true
$a \leq b$	a가 b보다 작거나 같으면 true
$a \geq b$	a가 b보다 크거나 같으면 true
$a == b$	a가 b와 같으면 true
$a != b$	a가 b와 같지 않으면 true

비교 연산자

❖ 초과 연산자

```
>> var 키 = 165;  
    var 키제한 = 150;  
    키 > 키제한;  
<< true
```

```
>> var 키 = 150;  
    var 키제한 = 150;  
    키 > 키제한;  
<< false
```

비교 연산자

❖ 이상 연산자

```
>> var 키 = 150;  
    var 키제한 = 150;  
    키 >= 키제한;  
<< true
```

비교 연산자

❖ 이하/미만 연산자

```
>> var 키 = 150;  
    var 키제한 = 120;  
    키 < 키제한;  
<< false
```

```
>> var 키 = 120;  
    var 키제한 = 120;  
    키 <= 키제한;  
<< true
```

삼중 등호(일치 연산자)

❖ 두 숫자가 같은지 확인

- 삼중 등호(===) 사용
 - 두 값이 완전히 같습니까? 라는 뜻
 - 등호 한 개(=)는 변수 오른쪽에 있는 값을 왼쪽에 저장하라는 뜻

```
>> var 비밀숫자 = 5;  
    var 아이언맨추측 = 3;  
    비밀숫자 === 아이언맨추측;  
<< false
```

```
>> var 헐크추측 = 7;  
    비밀숫자 === 헐크추측;  
<< false
```

```
>> var 토르추측 = 5;  
    비밀숫자 === 토르추측;  
<< true
```

이중 등호(동등 연산자)

❖ 자료형의 유형이 다른 두 개의 값이 같은지 비교

- 이중 등호(==) 사용

```
>> var 문자열숫자 = "5";  
    var 진짜숫자 = 5;  
    문자열숫자 === 진짜숫자;  
<< false
```

```
>> 문자열숫자 == 진짜숫자;  
<< true
```




06

조건 연산

조건 연산

❖ 조건의 결과에 해당하는 값을 반환

조건 ? 참일_때의_결과 : 거짓일_때의_결과

```
var x = 5;  
var y = 3;  
var big = (x > y) ? x : y;
```



07

문자열

문자열

❖ 자바스크립트에서 문자열은 문자를 의미

- 문자에는 글자, 숫자, 구두점, 공백이 포함

❖ 문자열 표현 방법

- 문자열 앞뒤로 큰따옴표(“)를 붙임

“안녕하세요. 반갑습니다!”

문자열

❖ 큰따옴표 안에 숫자를 넣은 경우 문자열로 취급

```
var 숫자아홉 = 9;  
var 문자열아홉 = "9";
```

❖ 변수에 문자열 저장 가능

```
var 나이 = 5;           // 숫자 저장  
나이 = "다섯살";       // 문자열 저장
```

문자열 비교

❖ 비교 연산자(!=, ==, >, <, <=, >=) 사용 가능

- 사전에서 뒤에 나오는 문자열이 크다고 판단

```
var name = "kitae";  
var res1 = (name == "kitae"); // res1 = true  
var res2 = (name > "park");   // res2 = false
```

문자열 연결하기

❖ + 연산자

```
>> var 인사 = "안녕하세요.";
    var 내이름 = "정수아";
    인사 + 내이름;
<< "안녕하세요.정수아 "
```

❖ concat() 함수

```
>> var 인사 = "안녕하세요.";
    인사.concat("정수아입니다.");
<< "안녕하세요.정수아입니다."
```

문자열 연결하기

❖ 따옴표 안 공백도 문자로 취급

```
>> var 인사 = "안녕하세요. ";  
    var 내이름 = "정수아";  
    인사 + 내이름;  
<< "안녕하세요. 정수아 "
```


문자열 앞 뒤 공백 제거

❖ trim() 함수

- 문자열의 앞 뒤 공백을 제거
- 예시

```
>> var name = " sooa ";  
    name = name.trim();  
<< 'sooa'
```

❖ trim()은 어디에 사용될까?

문자열 길이 구하기

❖ 문자열 길이

- .length 속성
 - 문자열의 길이 값을 숫자로 반환
- 문자열 뒤에 붙이는 경우

```
>> "동해물과백두산이마르고닳도록".length;  
<< 14
```

- 문자열이 포함된 변수 뒤에 붙이는 경우

```
>> var 자바 = "자바";  
    자바.length;  
<< 2
```

```
>> var 스크립트 = "스크립트";  
    스크립트.length;  
<< 4
```

```
>> var 자바스크립트 = 자바 + 스크립트;  
    자바스크립트.length;  
<< 6
```

문자열에서 한 글자만 가져오기

❖ 문자열의 특정 위치에 있는 글자 가져오기

- 대괄호([])를 사용
- 문자열이나 문자열이 포함된 변수 뒤에 대괄호를 쓰고, 그 안에 원하는 글자의 번호(인덱스)를 입력
 - 글자의 번호는 0부터 시작

• 예시) 변수 인사의 첫 글자 가져오기

```
>> var 인사 = "안녕하세요";  
      인사[0];  
<< "안"
```

```
> var 인사 = "안녕하세요";  
  인사[0]  
← "안"  
-----  
> 인사[1]  
← "녕"  
-----  
> 인사[2]  
← "하"  
-----  
> 인사[3]  
← "세"  
-----  
> 인사[4]  
← "요"
```

[실습] 문자열에서 한 글자만 가져오기

```
>> var 암호1 = "지도 속";  
    var 암호2 = "희망을";  
    var 암호3 = "훔쳐본다";  
    var 암호4 = "?!";  
    암호1[1] + 암호2[1] + 암호3[1] + 암호4[1];  
<< "도망쳐!"
```

문자열에서 한 글자만 가져오기

❖ 문자열의 특정 위치에 있는 글자 가져오기

- charAt(n) 함수

```
var 인사 = "안녕하세요.";
인사.charAt(0);    // 안
인사.charAt(1);    // 녕
인사.charAt(2);    // 하
인사.charAt(3);    // 세
인사.charAt(4);    // 요
인사.charAt(5);    // .
```

문자열 잘라내기

❖ 문자열의 일부를 잘라내기

- slice(a, b) 함수

매개변수	설명
첫 번째 매개변수(a)	잘라내기가 시작되는 부분의 인덱스
두 번째 매개변수(b)	잘라내기가 끝나는 글자의 바로 다음 인덱스

- slice(a)
 - a번부터 문자열의 끝까지 잘라내기
- slice(a, b)
 - a번부터 문자열의 b-1번까지 잘라내기

문자열 잘라내기

❖ 문자열의 일부를 잘라내기

- slice(2, 5)
- 문자열에서 2번 글자부터 5번 글자 전까지 잘라내기

M	y		l	o	n	g		s	t	r	i
0	1	2	3	4	5	6	7	8	9	10	11

[실습] 문자열 잘라내기

❖ slice(a)

```
>> var 긴문자열 = "이 문자열은 참 깎니다";  
    긴문자열.slice(2);  
<< "문자열은 참 깎니다"
```

❖ slice(a, b)

```
>> var 긴문자열 = "이 문자열은 참 깎니다";  
    긴문자열.slice(2, 5);  
<< "문자열"
```


문자열 변경

❖ 일부 문자열을 다른 문자열로 변경

- replace() 함수

```
replace(찾을_문자열, 변경할_문자열)
```

- 예시

```
>> var every = "Boys and Girls";  
    var replace = every.replace("and", "or");  
    replace;  
<< "Boys or Girls"
```

문자열 분할

❖ 문자열 분할

- split("구분자") 함수
 - 하나의 문자열을 구분자를 기준으로 여러 개의 문자열로 분할한 배열 리턴

- 예시

- 빈칸을 기준으로 every 문자열을 "Boys", "and", "Girls"로 분할

```
var every = "Boys and Girls";  
var sub = every.split(" ");
```

- 결과

```
>> sub[0]  
<< "Boys"
```

```
>> sub[1]  
<< "and"
```

```
>> sub[2]  
<< "Girls"
```

문자열 전체를 대소문자로 바꾸기

❖ 대문자로 바꾸기

- toUpperCase() 함수
 - 영어로 된 모든 글자를 대문자로 변경
 - 숫자나 한글에는 적용 안됨
 - 예시

```
>> "Hello there, how are you doing?".toUpperCase();  
<< "HELLO THERE, HOW ARE YOU DOING?"
```

❖ 소문자로 바꾸기

- toLowerCase() 함수
 - 영어로 된 모든 글자를 소문자로 변경
 - 예시

```
>> "hELLo THERE, How ARE yOu doINg?".toLowerCase();  
<< "hello there, how are you doing?"
```

[실습] 첫 글자는 대문자, 나머지는 소문자로 변경

```
>> var 고칠문자열 = "hELlo THERE, How ARE yOu doINg?";  
    var 소문자 = 고칠문자열.toLowerCase();  
    var 첫글자 = 소문자[0];  
    var 첫글자대문자 = 첫글자.toUpperCase();  
    var 나머지 = 소문자.slice(1);  
    첫글자대문자 + 나머지;  
<< Hello there, how are you doing?
```

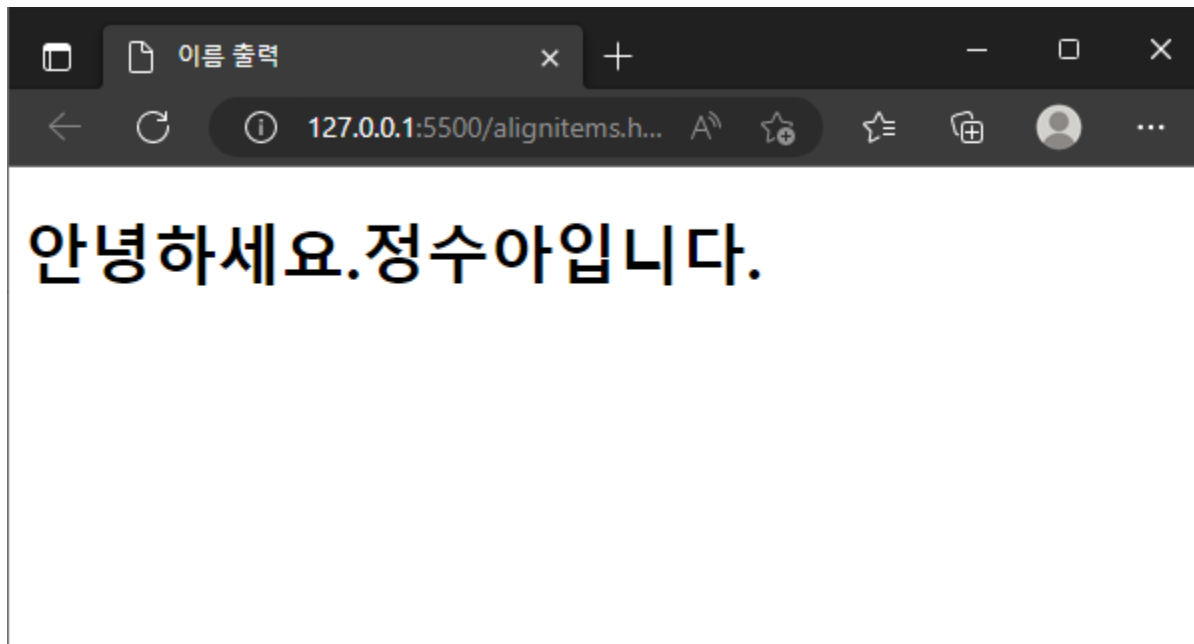
[실습] HTML에 자바스크립트 적용

❖ 웹 브라우저에 이름 출력해보기

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>이름 출력</title>
  </head>
  <body>
    <script>
      var name = "정수아";
      document.write("<h1>안녕하세요.");
      document.write(name);
      document.write("입니다.</h1>");
    </script>
  </body>
</html>
```

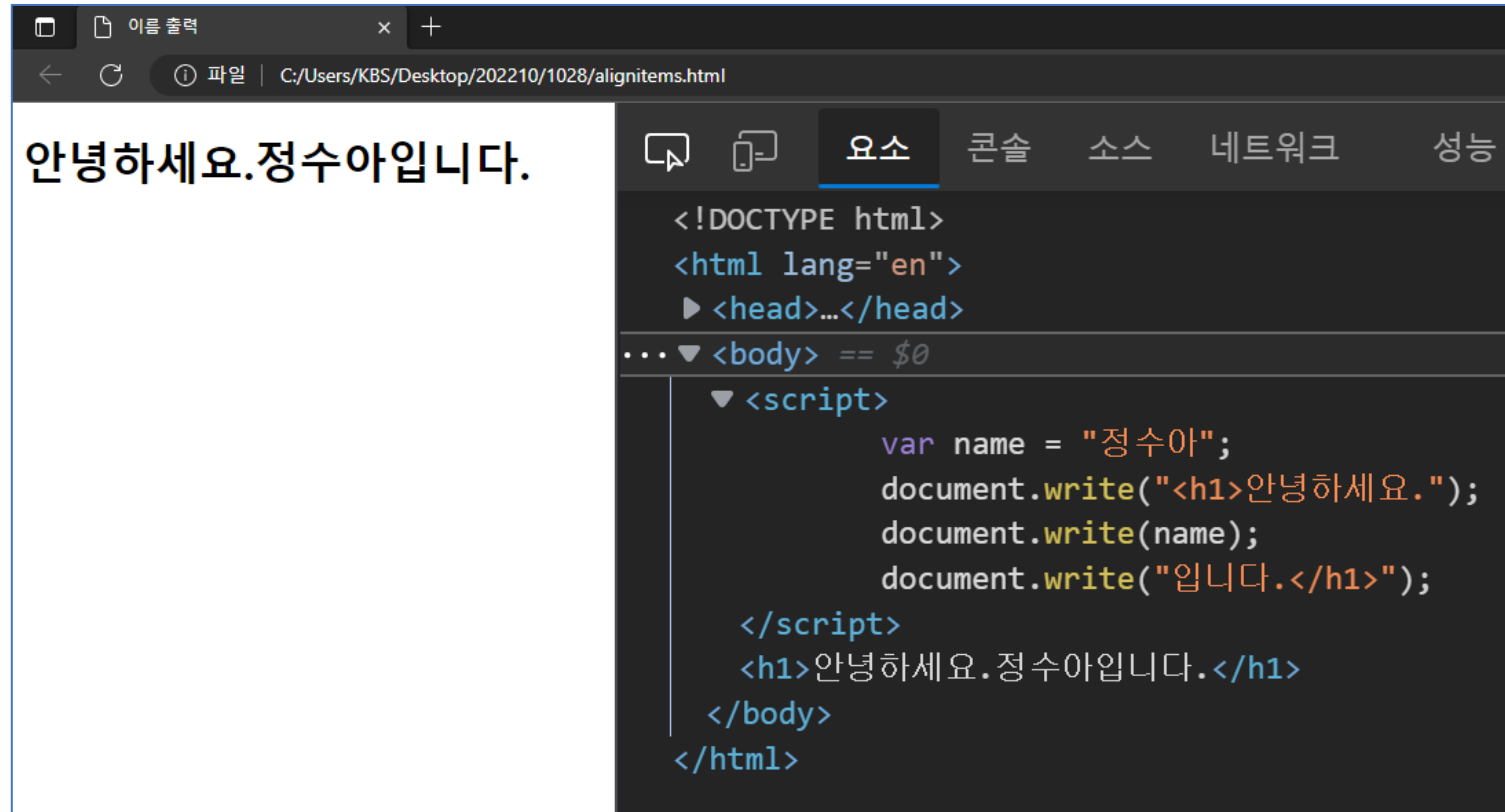
[실습] HTML에 자바스크립트 적용

❖ 실행 결과



[실습] HTML에 자바스크립트 적용

❖ 실행 결과



THANK 😊 YOU