

javascript

frontend 과정

수업에 들어가기 앞서

- 프로그래밍을 한번도 접하지 않은 수강생을 고려하여 진행할 예정
- 질문은 언제든지 슬랙 채널을 통해 가능
수업시간 외의 질문은 확인되면 최대한 빨리 답변드릴 예정
(태그해주셔야 확인이 빠릅니다)
- 코드 관련 질문은 슬랙 채널에 본인이 작성한 코드와 (에러가 있다면) 에러메세지를 캡처 혹은 copy&paste 하여 전송

javascript 개요

- javascript란?
- Hello, World
- 개발자도구
- 기본문법과 주석
- 변수와 명명법
- 자료형과 형변환

javascript

- HTML, CSS를 조금 더 다이나믹하게 사용하기 위해 탄생한 프로그래밍 언어
- 각 브라우저마다 사양이 달라지자, ECMAScript라는 표준 탄생
- 오늘날의 javascript는 ECMAScript 표준을 따라 개발되어 브라우저에 내장

Hello, World

- html 파일의 <script></script> 태그 사이에 javascript 작성
=> 브라우저가 html을 해석 하는 중, script를 만나면 script 실행

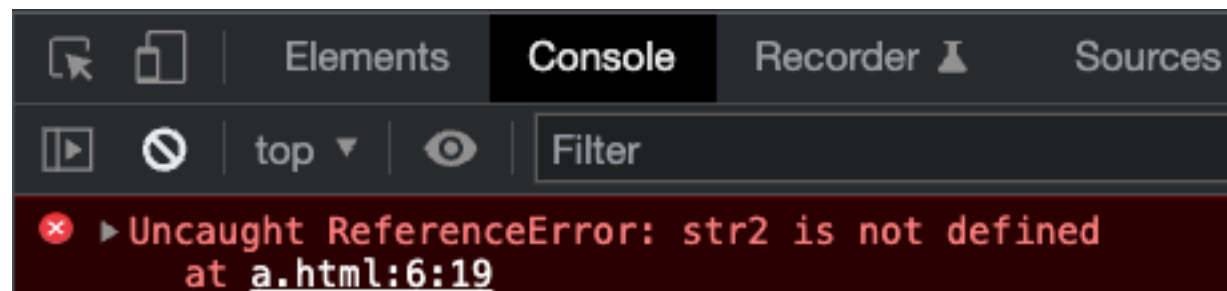
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script>
      var str = 'Hello, World';
      console.log(str);
    </script>
  </head>
  <body></body>
</html>
```

- console.log라는 함수를 호출하면
브라우저의 개발자 도구에서 확인 가능

개발자콘솔

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script>
      const str = 'Hello, World';
      console.log(str2);
    </script>
  </head>
  <body></body>
</html>
```

- 브라우저에 javascript 개발/디버깅 등을 도와주기 위해 내장
- console 탭에서 javascript 를 사용하거나, 에러를 확인할 수 있음



기본문법

- 자바스크립트는 코드라인을 세미콜론(;)으로 구분.
- 줄바꿈을 하면 세미콜론을 생략가능하지만, 가독성과 예상치못한 버그를 막기 위해 한동안은 매번 넣어주는 것을 추천

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script>
      var str = 'Hello, World'
      console.log(str);
    </script>
  </head>
  <body></body>
</html>
```

주석

- 주석이란?
=> 프로그래밍 언어에서 의도적으로 무시되는 코드
- 자신과 타인을 위한 메모의 역할

```
<script>  
  // 한줄 주석  
  /*  
    여러줄(multiline) 주석  
    console.log(111);  
  */  
  /* 한줄처럼 사용 가능 */  
</script>
```


변수

- 변수(variable)?
=> 하나의 값을 저장할 수 있는 저장공간에 변수명을 붙여 접근
- var, let, const 등의 예약어를 통해 변수를 선언(=만듦)
- 예약어(reserved word)란?
=> 프로그래밍 언어에서 기본적으로 역할을 가지고 있는 키워드

변수선언

- var라는 예약어 뒤에, 변수명 "str"을 적어 변수 선언(=생성)
- 대입 연산자 = 를 활용해 str 변수에 'Hello, World' 값 대입

```
<script>  
  var str;  
  console.log(str);  
  str = 'Hello, World';  
  console.log(str);  
</script>
```

- 선언과 값 대입을 동시에 할 수 있음

```
<script>  
  var str = 'Hello, World';  
  console.log(str);  
</script>
```

변수 명명규칙

- 문자와 숫자, \$, _만 사용 가능
- 숫자는 첫 글자에 올 수 없음
- 예약어는 변수명으로 사용할 수 없음

변수선언

- var, let, const의 차이 (더 많은 차이가 있지만 당장엔 이것만)
- var: 같은 이름으로 다시 선언 가능
여러차례 대입 가능
- let: 같은 이름으로 다시 선언 불가
대입을 통해서만 값 변경
- const: 같은 이름으로 다시 선언불가
선언할 때 대입이 필수

```
<script>  
  var str = 'Hello, World';  
  var str = 'Hello, World';  
  console.log(str);  
</script>
```

```
<script>  
  let str = 'Hello, World';  
  str = 'Hello, World';  
  let str = 'Hello, World';  
</script>
```

```
<script>  
  const str;  
  str = 'Hello, World';  
</script>
```

```
<script>  
  const str = 'Hello, World';  
  const str = 'Hello, World';  
</script>
```

연습문제

1. `admin`과 `name`이라는 변수를 선언하세요.
2. `name`에 값으로 `"John"`을 할당해 보세요.
3. `name`의 값을 `admin`에 복사해 보세요.
4. `admin`의 값을 `console`에 출력해보세요.
`"John"`이 출력되어야 합니다.

자료형

- javascript는 변수의 type(자료형)이 고정되어 있지 않음
원시(primitive) 타입과 참조(reference)타입이 존재
- 원시(primitive) 타입
값의 불변성 (값 자체가 바뀌지 않고, 새로운 값을 변수에 할당가능)
string, number, bigint, boolean, undefined, symbol, null

string

- 문자열 이라고 부름
- 문자열 선언 방법
 - double quote(큰따옴표, ")
 - single quote(작은따옴표, ')
 - back quote(억따옴표, `)
- double quote와 single quote는 역할이 동일
(취향에 따라 선택. 섞어 쓸 수 있음)
- back quote는 Template literals (문자열 내에 javascript)사용 가능

```
let str;  
str = "Hello, World";  
str = 'Hello, World';  
str = "It's me";  
str = 'Say "Hi"';  
const someVariable = 5;  
str = `someVariable's value is  
"${someVariable}"`;
```

number

- 정수, 소수 표현 가능
- 일반적으로 아는
덧셈, 뺄셈, 나눗셈, 곱셈 등
연산자 통해 수행가능
- 특수숫자값: Infinity, NaN
- 각 연산자는 숫자에만 사용할
수 있는게 아니니 주의

```
let num = 1;
num = 12.345;

let Inf = 1 / 0;
console.log(Inf); // Infinity
Inf = Infinity;
console.log(Inf); // Infinity

const notANumber = "aaa" / 2;
console.log(notANumber); // NaN

const strPlusNum = "a" + 1;
console.log(strPlusNum);

const numPlusStr = 1 + "a";
console.log(numPlusStr);
```


boolean

- 참과 거짓을 나타내는 데 쓰임
- 비교 연산, 논리 연산 등에 주로 활용

```
let t = true;  
let f = false;  
console.log(t == f);  
console.log(t != f);  
console.log(!t);  
console.log(!f);  
console.log(t || f);  
console.log(t && f);
```

- 참고) https://ko.wikipedia.org/wiki/불리언_자료형

undefined

- 변수를 선언하고 대입하지 않으면 기본적으로 undefined 상태
- undefined를 명시적으로 대입할 수도 있음

```
let str;  
console.log(str);  
let str = undefined;  
console.log(str);
```

null

- 어떤 값이 의도적으로 비어있음을 명시적으로 표현할 때 사용

```
let str;  
console.log(str); // undefined  
let str = null;  
console.log(str); // null
```

형변환

- 형변환(타입 캐스팅) 예시

```
str = '1';  
num = +str;  
num = Number(str);  
  
num = 1;  
str = '' + num;  
str = String(num);  
str = `${num}`;  
  
bool = !!str;  
bool = Boolean(str);
```

연습문제

아래 스크립트의 결과를 예측해 보세요.

```
1  let name = "Ilya";  
2  
3  console.log( `hello ${1}` ); // ?  
4  
5  console.log( `hello ${"name"}` ); // ?  
6  
7  console.log( `hello ${name}` ); // ?
```

원시타입

Step 1

let a = 100;

100

Step 2

a = 50;

100

50

원시타입

Step 1

let a = 100;

a : 100

let b = a;

b : 100

Step 2

let b = a;

a : 100

a = 50;

b : 100

a : 50