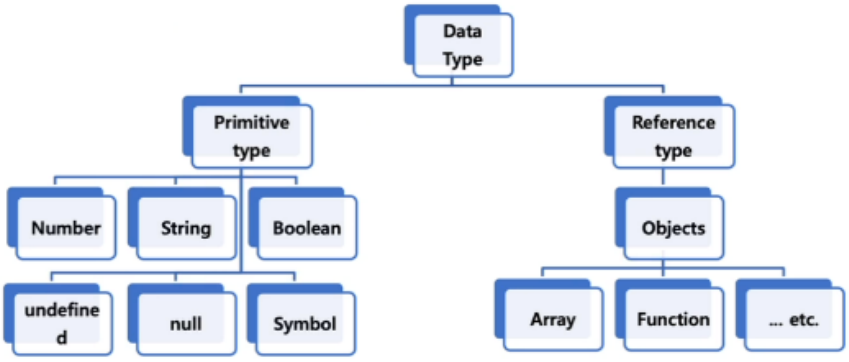
js 01

- 브라우저가 어떤역할을 하는지
- dom이 어떤건지 dom tree가 어떤건지 (중요하진 않음, 공부하는 김에 보기)
- 파싱, ECMA 용어 등등 ~~~~스타일가이드까지시험 X

- 변수부터 시작! (let, const, var : 각각 어떤 차이가 있는지 , 어떤게 안되는지)

키워드	재선언	재할당	스코프	비고
let	X	O	블록 스코프	ES6부터 도입
const	X	X	블록 스코프	ES6부터 도입
var	O	O	함수 스코프	사용 X

- 데이터 타입 (원시 타입,)
  - 자바스크립트의 모든 값은 **특정한 데이터 타입을 가짐**
  - 크게 **원시 타입* (Primitive type)**과 **참조 타입* (Reference type)**으로 분류됨



원시 타입 : 넘버, 스트링, 불린 타입 기억해야함

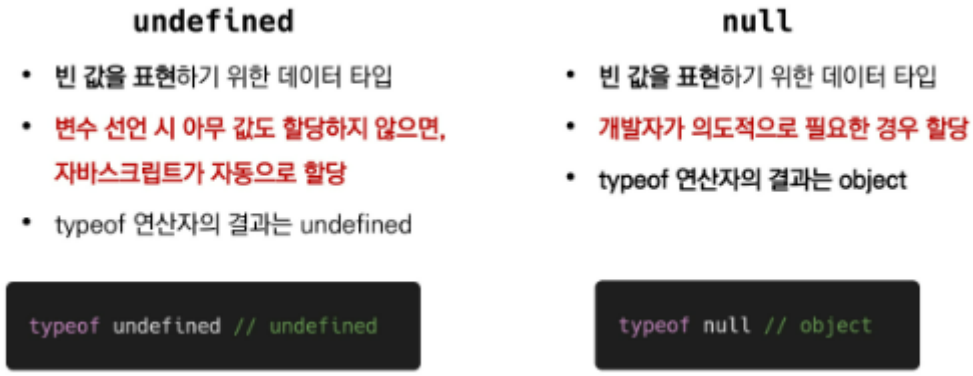
참조타입 : 객체 : array, function 기억

- 찍어보면 다른게 있음??

```
let firstName
console.log(firstName) // undefined
```

```
let firstName = null
console.log(firstName) // null

typeof null // object
```



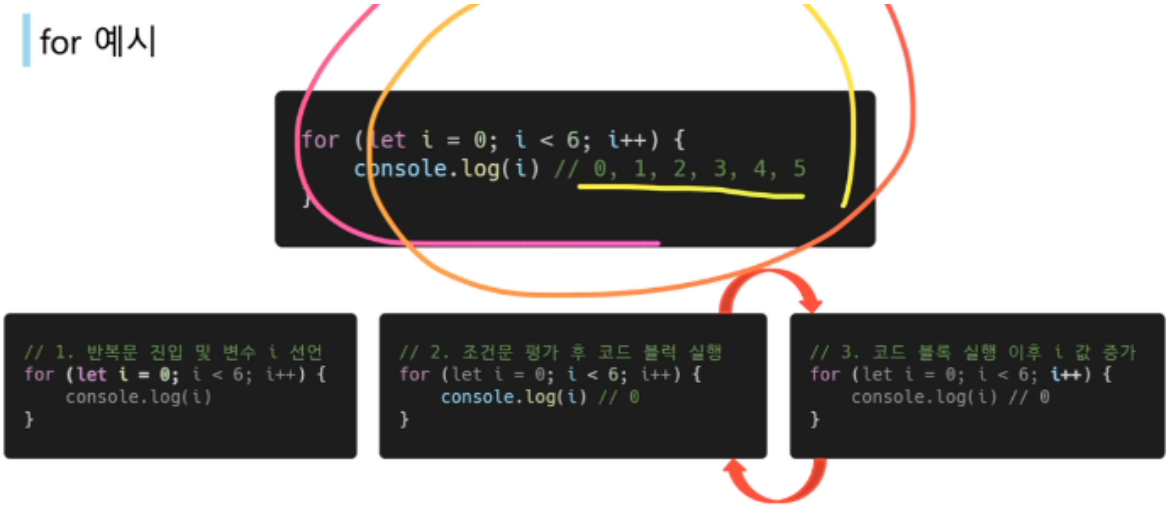
- 연산자는 ===만 알면됨 타입과 값을 모두 비교함,

==, === 은어떤차이가 있는지

- 반복문 (당연히 나옴) for, for in for of가 어떻게 다른지 알아야함

- **for**
- **for...in**
  - 주로 **객체(object)의 속성들을 순회**할 때 사용
  - 배열도 순회 가능하지만 인덱스 순으로 순회한다는 보장이 없으므로 **권장하지 않음**
- **for...of**
  - **반복 가능한(iterable)* 객체를 순회**하며 **값을 꺼낼 때 사용**
    - 반복 가능한(iterable) 객체*의 종류: Array, Map, Set, String 등

for 예시



이런 걸 보면서 0 1 2 3 4 5 가 나오겠구나 하고 알아야함

for...of 예시

```
const fruits = ['딸기', '바나나', '메론']

for (let fruit of fruits) {
  fruit = fruit + '!'
  console.log(fruit)
}

for (const fruit of fruits) {
  // fruit 재할당 불가
  console.log(fruit)
}
```

- 코드가 나왔을 때 console.log로 찍었을때 출력 결과가 어떤건지 알아야함

코드는 직접 코드 보고 해석할 수 있어야함

- 함수 : 일급객체가 중요함

- (참고) JavaScript의 함수는 일급 객체*(First-class citizen)에 해당
  - 일급 객체*: 다음의 조건들을 만족하는 객체를 의미함
    - 변수에 할당 가능
    - 함수의 매개변수로 전달 가능
    - 함수의 반환 값으로 사용가능

- 호이스팅은 자세히는 안나오니 너무 열심히 공부 x

- 어떤 경우에 뭘 생략할 수 있는지 알아야함

Arrow Function

```
const arrow1 = function (name) {
  return `hello, ${name}`
}

// 1. function 키워드 삭제
const arrow2 = (name) => { return `hello, ${name}` }

// 2. 매개변수가 1개일 경우에만 ( ) 생략 가능
const arrow3 = name => { return `hello, ${name}` }

// 3. 함수 바디가 return을 포함한 표현식 1개일 경우에 { } & return 삭제 가능
const arrow4 = name => `hello, ${name}`
```

- 문자열 메서드 다외울필요 x (한번씩은 봐라)
- 배열은 파이썬과 비슷, 그러나 앞에 추가하고 빼기, 뒤에 추가하고 빼기정도는 알아야함 (메서드)

```
const numbers = [1, 2, 3, 4, 5]

numbers.push(100)
console.log(numbers) // [1, 2, 3, 4, 5, 100]

numbers.pop()
console.log(numbers) // [1, 2, 3, 4, 5]
```

- **array.push()**
  - 배열의 가장 뒤에 요소 추가
- **array.pop()**
  - 배열의 마지막 요소 제거

- foreach (비슷한 함수 3개 알아야함)
  - array는 많이 안씀 , element, index 에 대해 콜백함수를 실행하는 메서드

array.forEach(callback(element[, index[,array]]))

```
array.forEach((element, index, array) => {
  // do something
})
```

```
const fruits = ['딸기', '수박', '사과', '체리']

fruits.forEach((fruit, index) => {
  console.log(fruit, index)
  // 딸기 0
  // 수박 1
  // 사과 2
  // 체리 3
})
```

- 배열의 각 요소에 대해 콜백 함수를 한 번씩 실행
- 콜백 함수는 3가지 매개변수로 구성
  - element: 배열의 요소
  - index: 배열 요소의 인덱스
  - array: 배열 자체
- **반환 값(return)이 없는 메서드**

- 키는 문자열만 가능, 벨류는 모든값, 자바스크립트는 json

객체와 메서드

```
const me = {
  firstName: 'John',
  lastName: 'Doe',

  fullName: this.firstName + this.lastName,

  getFullName: function () {
    return this.firstName + this.lastName
  }
}
```

- 메서드는 객체의 속성이 참조하는 함수.
- **객체.메서드명() 으로 호출 가능.**
- **메서드** 내부에서는 this 키워드가 **객체**를 의미함.
  - fullName은 메서드가 아니기 때문에 정상출력 되지 않음(NaN)
  - getFullName은 메서드이기 때문에 해당 객체의 firstName 과 lastName을 정상적으로 이어서 반환

- 구조분해할당 알아야함 (이런식으로 쓰는 구나 정도 직접 타이핑하면서 알아두기)

객체 관련 ES6 문법 (4) – 구조 분해 할당 (destructing assignment)

- 배열 또는 객체를 분해하여 속성을 변수에 쉽게 할당할 수 있는 문법
- 예시)

```
const userInformation = {
  name: 'ssafy kim',
  userId: 'ssafyStudent1234',
  phoneNumber: '010-1234-1234',
  email: 'ssafy@ssafy.com'
}

const name = userInformation.name
const userId = userInformation.userId
const phoneNumber = userInformation.phoneNumber
const email = userInformation.email
```

```
const userInformation = {
  name: 'ssafy kim',
  userId: 'ssafyStudent1234',
  phoneNumber: '010-1234-1234',
  email: 'ssafy@ssafy.com'
}

const { name } = userInformation
const { userId } = userInformation
const { phoneNumber } = userInformation
const { email } = userInformation

const { name, userId } = userInformation
```

- 교수님한테 안들은 설명 나와도 생전 처음 본거를 찍을 필요가없음
- 배운거에서 틀린게나오니 함정 조심

---

js 02

- 이 사람 알아두기

  - 브랜던 아이크 (Brendan Eich)
    - JavaScript 최초 설계자
    - 모질라 재단 공동 설립자
    - 코드네임 피닉스 프로젝트 진행
      - 파이어폭스의 전신

- 역사는 넘어가기

- 쿼리셀렉터, 쿼리셀렉터올을 쓴다 정도~

셀렉터 안쪽에 뭐가 들어가는지, 아이디 선택자 , 클랙스 선택자, 태그들이들어간다.

DOM 선택 – 선택 관련 메서드 (1/2)

- document.querySelector(selector)
  - 제공한 선택자와 일치하는 element 하나 선택
  - 제공한 CSS selector를 만족하는 첫 번째 element 객체를 반환 (없다면 null)
- document.querySelectorAll(selector)
  - 제공한 선택자와 일치하는 여러 element를 선택
  - 매칭 할 하나 이상의 셀렉터를 포함하는 유효한 CSS selector를 인자(문자열)로 받음
  - 지정된 셀렉터에 일치하는 NodeList를 반환

- append하면 뒤에 붙일수있구나 정도만 알면됨

- innerText innerHTML 이거 두개는 차이알아야함 (헷갈리는것임)
- 보기에 delete()로 나오면 틀림

- ChildNode.remove()

- setattribute()
- 이벤터는 addeventlistener()랑 onclick등

Event handler - addEventListener() (2/4)

Event handler - addEventListener() (4/4)

"대상에 특정 이벤트가 발생하면, 할 일을 등록하자"

EventTarget.addEventListener(type, listener)

타입엔 뭐가 들어가는지?>

리스너인자에 콜백함수가 들어감

- 이거 알기
  - event.preventDefault()
  - 현재 이벤트의 기본 동작을 중단

-