

* 변수

var (global) ← 전역적 선언과 초기화 가능

let (block) ← блок 선언과 초기화 가능

const (상수)

* type binding

Java

int

a;

int m[2][2]
v[2]

} static type binding

JavaScript

var

a;

string

a = "Hello";

number

a = 100;

boolean

a = true;

76 01 2d2f
18 01 2d2f
2d 0f -m 23

} dynamic

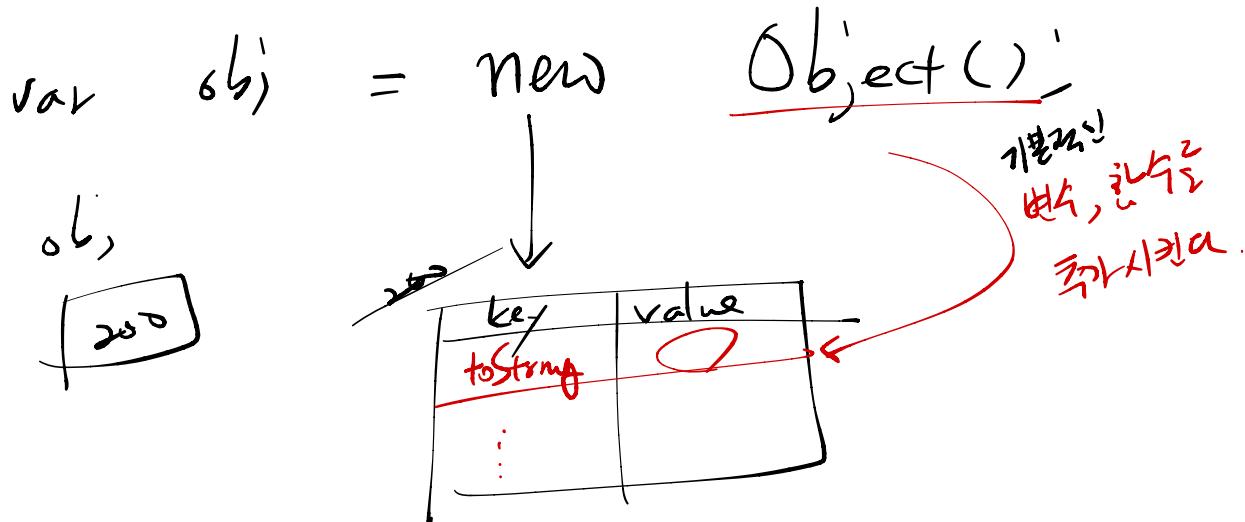
"

* function

plus
ok = plus;
 ^ ok
haha = ok;
 ^ haha

function plus(a, b) {
 return a + b;
}

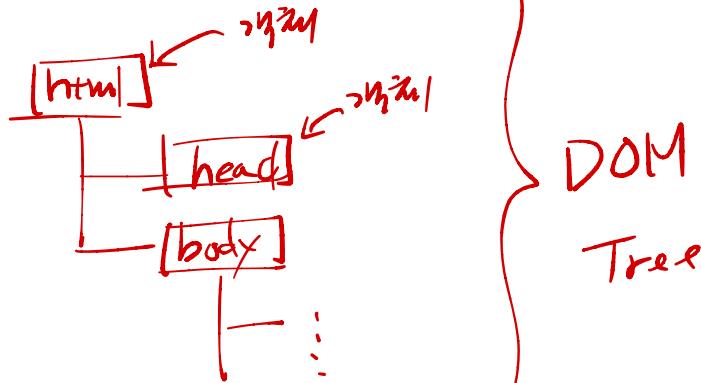
* object



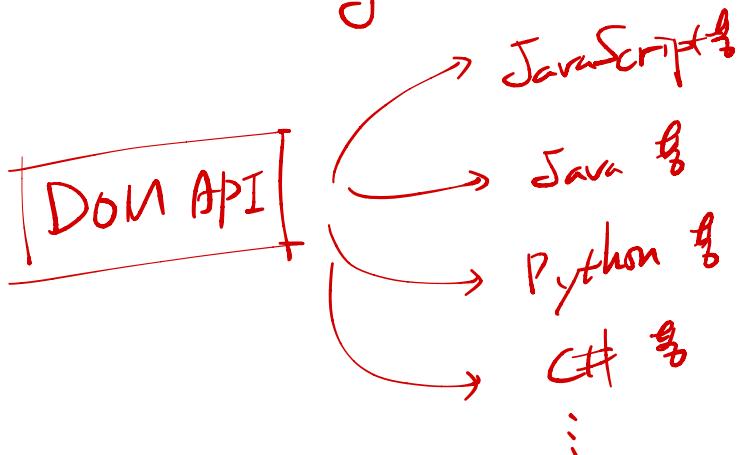
* DOM API (Document Object Model)

↳ tag է է այլ այլ API

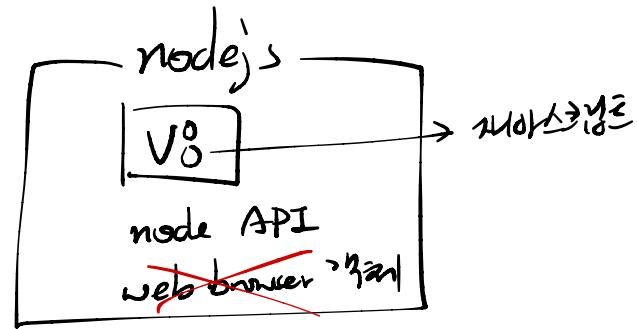
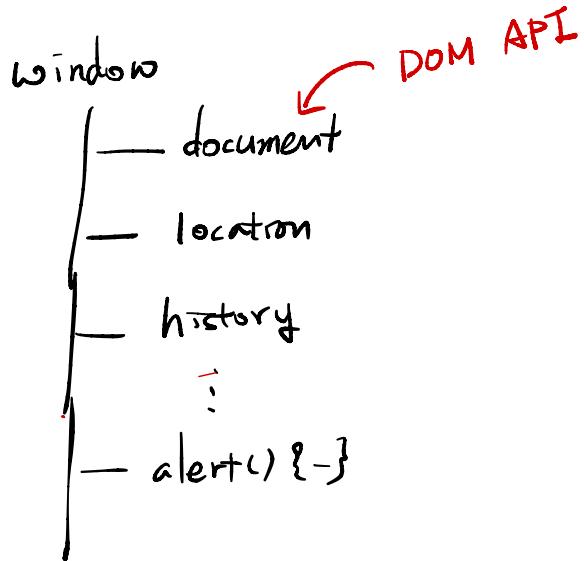
<html>
 <head> —> <head>
 <body> —> <body>
</html>

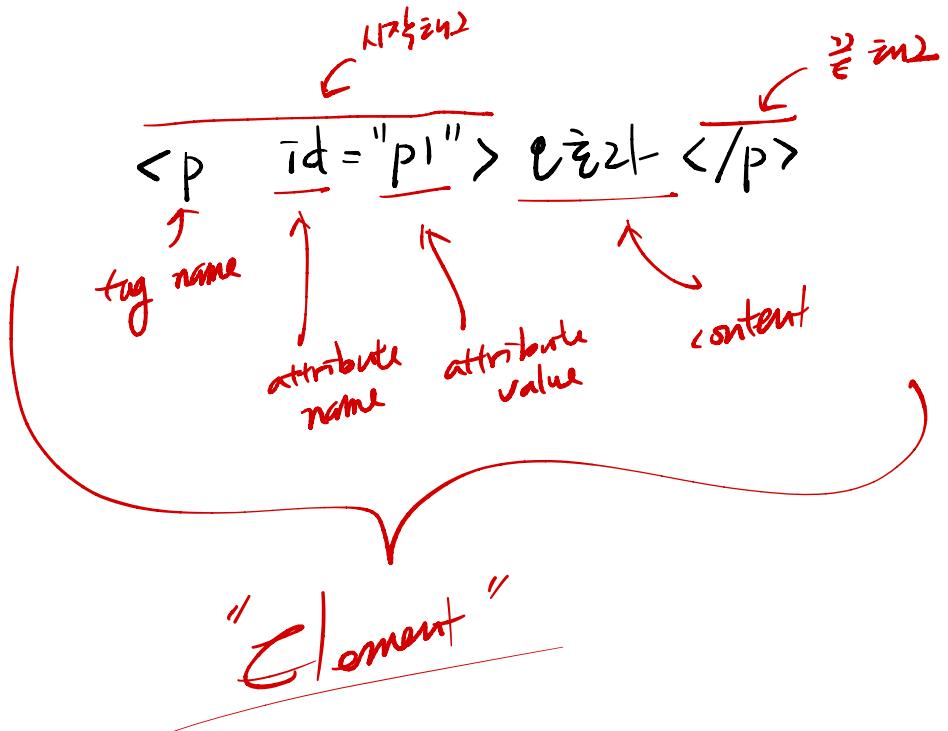


tag ← → HTML



* Web Browser 을 위한 API

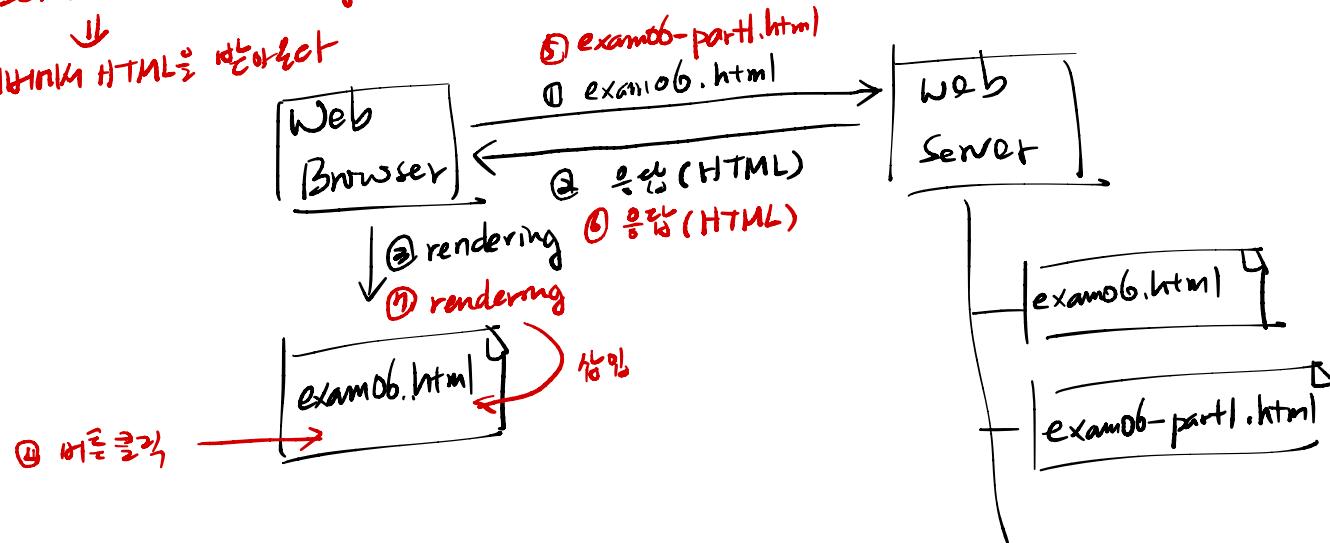




- * AJAX (Asynchronous JavaScript And XML)

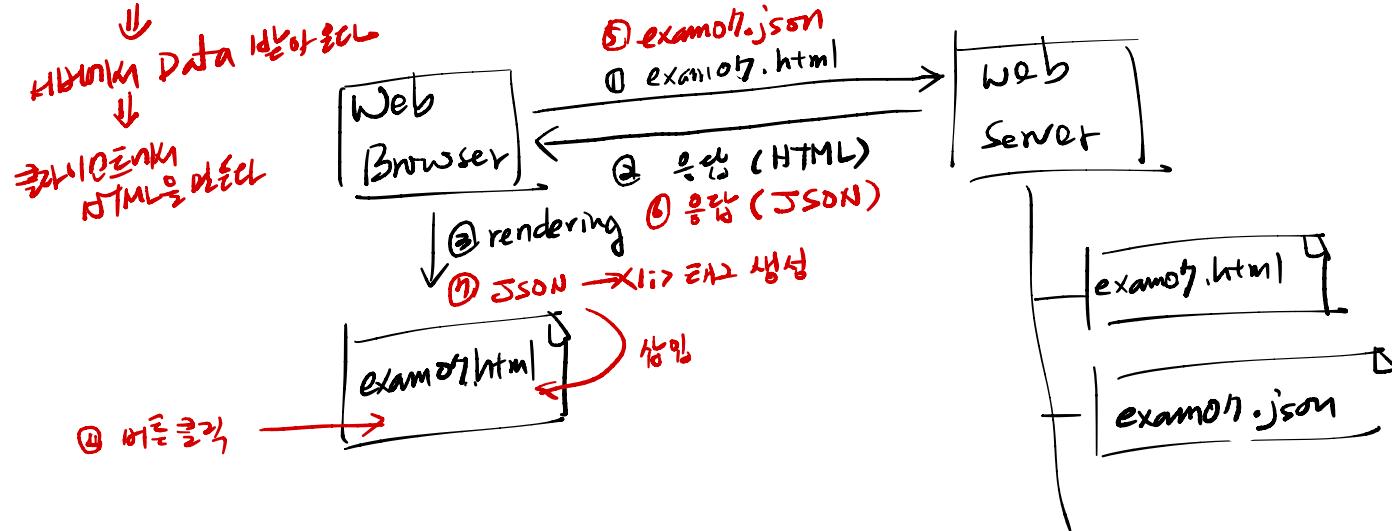
① Server-side rendering

↓
HTML이 HTML이 되도록



* AJAX (Asynchronous JavaScript And XML)

② client-side rendering



* Data Type

- ✓ string → "aaa" , 'aaa'
- ✓ number → 300 , 3.14 , $314E-2$ (314×10^{-2}) , NaN , Infinity , -Infinity
- ✓ boolean → true , false
- ✓ undefined → 미정의 상태를 가진 값은 undefined = undefined type의 대표적인 값
- ✓ object → object , null ← object 타입
- ✓ function → 함수
- ✓ symbol → 기호

* 10102

var arr = new Array();

비기본적

key	value
toString()	=
--proto--	
:	
length	

new



Object()



Array()

→ 헤



Object() 흐름 →



Array() 흐름



10102번 프로퍼티 흐름

모든 개체가
별도의 갖춰야 하는
프로퍼티(변수,
함수, 기능)
초기

```
var arr = new Array()
```

```
arr[0] = "aaa";
```

```
arr[1] = "bbb";
```

```
arr[5] = "ccc";
```

```
console.log(arr.length);
```



key	value
0	"aaa"
1	"bbb"
5	"ccc"

* 例解

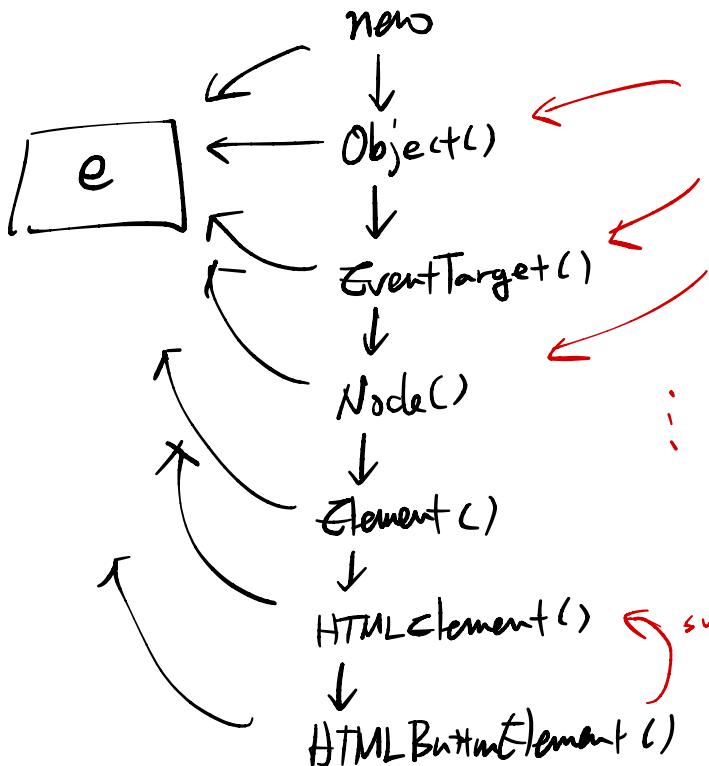
var e = document.getElementById("btn"); // <button> —</button>

~~new X()~~

~~new
Object()~~

~~⋮~~

~~X()~~



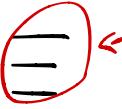
이것들이 바로 이거다!
그들이 바로 그것이다!

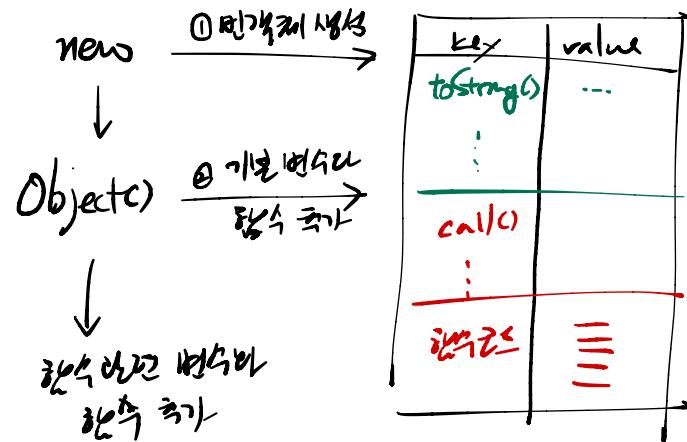
\"내가 네 (constructor)\"

⋮

super 이거다!

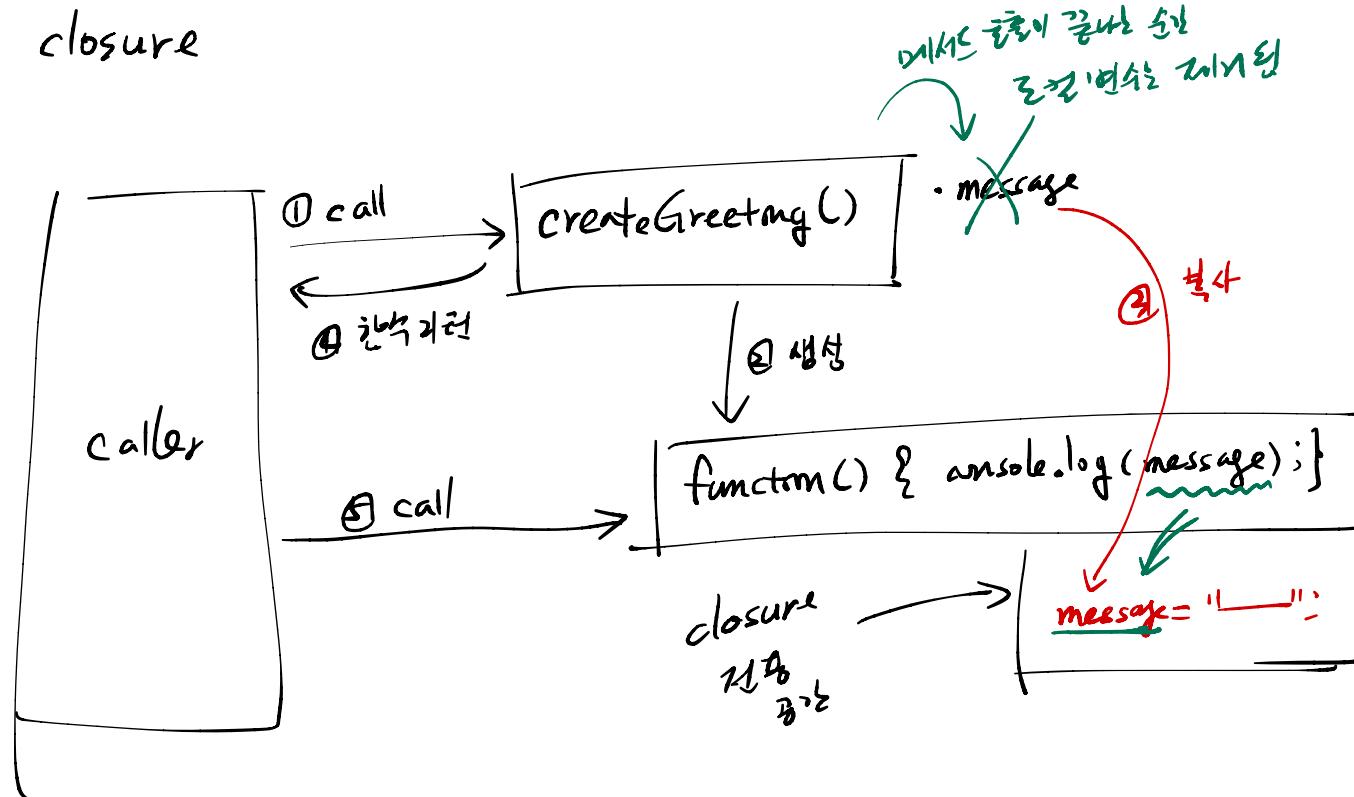
* function object

function f1() {
 
 "function body"
}

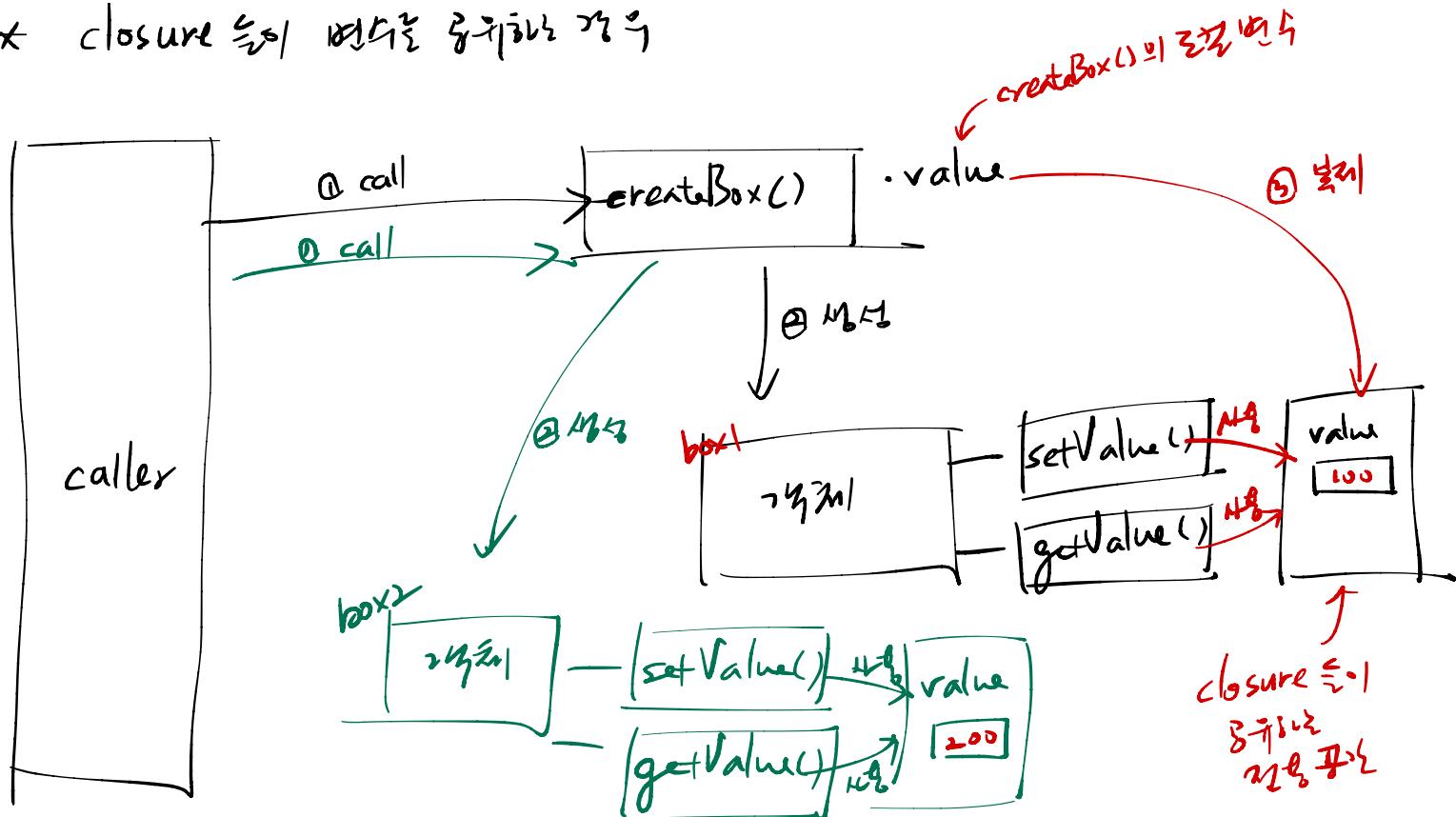


function = 구조 + 기능

* closure



* closure یعنی 떠나도 끌고가는 것



* 초기화 초기화

prototype (prototype)

① Object.create(null)

↓ 초기화

key	value

초기화

② Object.create(Object.prototype) == new Object()

원형은 부모로
인자로 초기화

원형은 부모로
인자로 초기화

→ prototype 초기화
기본 메서드/메타데이터
설정 초기화

key	value
toString()	-
:	

==

key	value
toString()	-
:	

↑ 초기화

④ {}

* 메모리와 같은 차이

var obj = new Object()

obj.f1 = function() {};

① new

메모리에 할당

key	value
f1()	-

참조

key	value
toString()	-
valueOf()	-
:	-

할당된 메모리
(= 메모리)

② Object()

f1 프로퍼티 추가

기본값은
비어 있는
상태

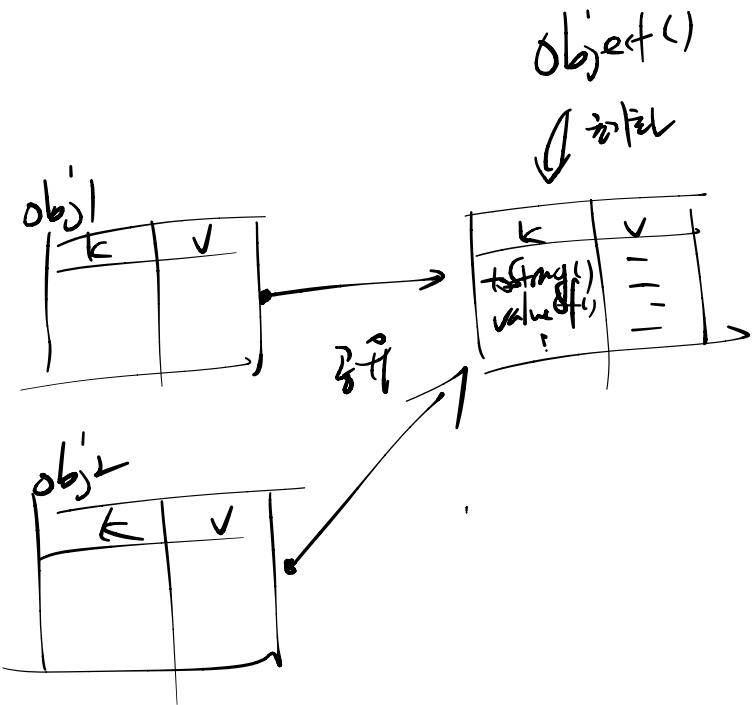
obj.f1(); → obj가 원래 f1()를 갖지 않았음

obj.toString(); → obj가 원래 f1()를 갖지 않음(X) → 전형적인 예에서 갖지 않았음!

* 오브젝트의 초기화

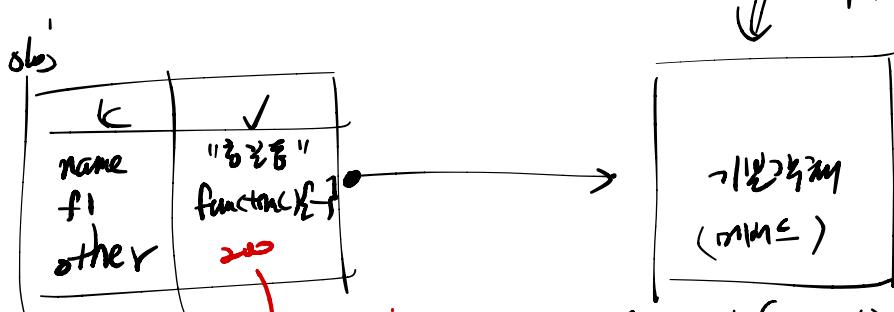
```
var obj = new Object();
```

```
var obj2 = new Object();
```

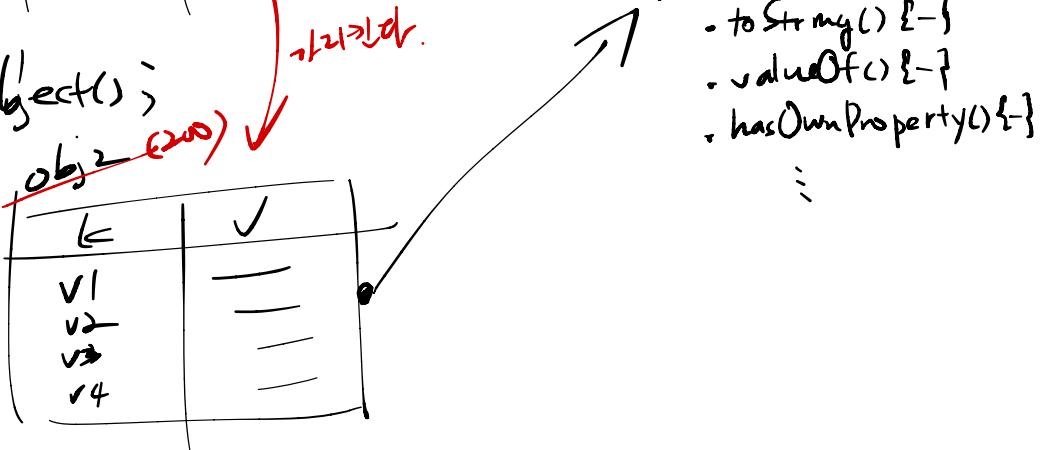


* 이란 말 → 다른 객체를 프로토로 흡수(흡연)

var obj = new Object();



var obj2 = new Object();



obj.other.v1 = 100

* JavaScript vs ECMAScript

(1996)

J-S 1.0 → 1.1 → 1.2 → 1.3 → 1.4 → 1.5 → ... → 1.8.5 /

표준으로 표기

ECMA-262
DRAFT

제작자 팀

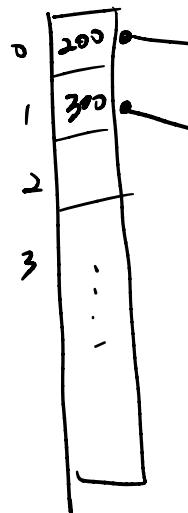
ECMAScript 1 (1991) → ES2 (1998) → ES3 (1999) → ES4 → ES5 (2009) → ES6 (2015)

ES 2022

* 例 2: 在堆中

var arr = [];

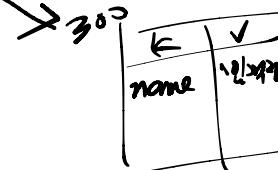
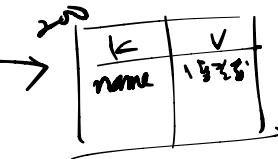
arr[0] = new Object();



arr[0].name = "张三";

arr[1].name = "李四";

arr[1] = new Object();

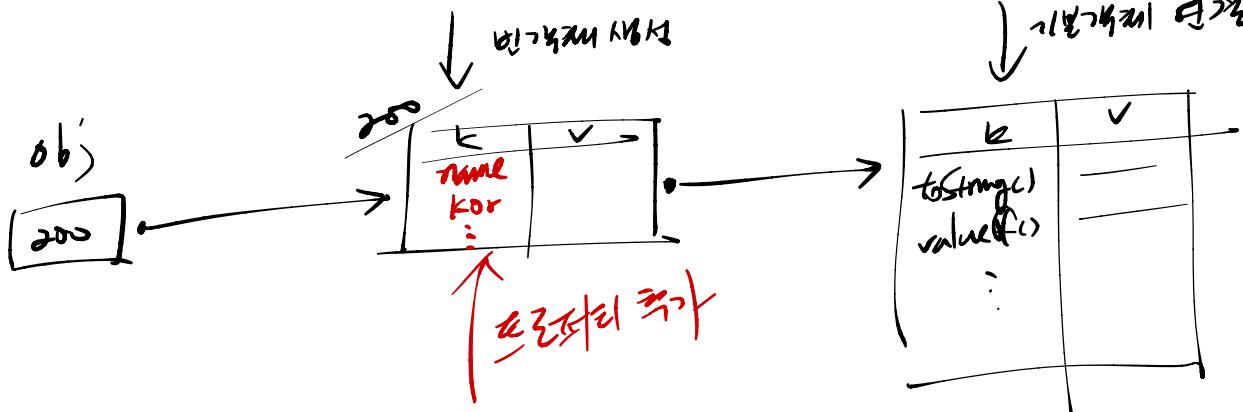


* new چیزی یعنی

var ob = new createScore();

↳ constructor
↳ new چیزی
یعنی ساخت

① new → Object() چیز



② createScore() چیز

* Object prototype

`var obj1 = new Score();`

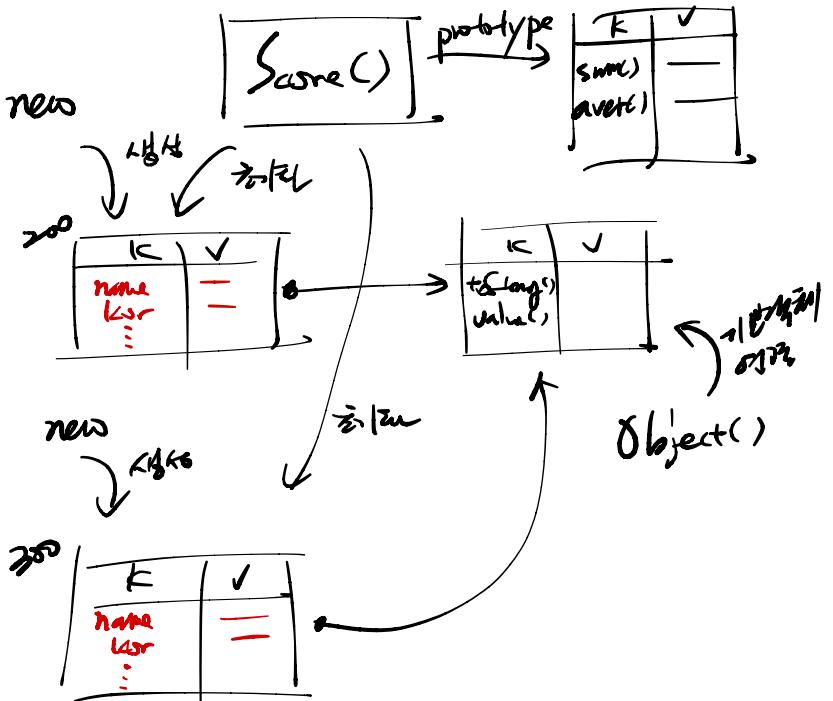
`obj1` [200]

`var obj2 = new Score();`

`obj2` [300]

`Score.prototype.sum = function() { }`

`Score.prototype.avr = function() { }`



`obj1.sum();`

↳ ① `sum()` is not defined
↓ Body

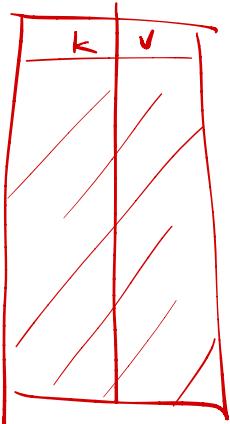
② `Object.prototype.sum()` is not defined

* မှာမျှတဲ့ ၁၂၅၄ ၁၆၆

```
Engm(va,cy,cc) {  
    this.cc = cc;  
    this.cylinder = cy;  
    this.value = va;  
}
```

```
Car(va,cy,cc,ca,au) {  
    Engme.call(this,  
               va,cy,cc);  
    this.capacity = ca;  
    this.auto = au;  
}
```

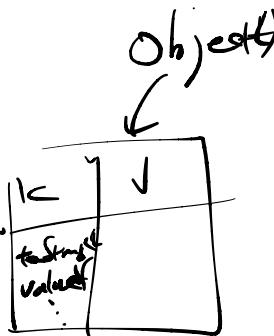
new →
↓
Object() →
↓
Engine() →
↓
Car() →



var ob = new Car();

ob
200

K	V
cc	—
cylinder	—
value	—
capacity	—
auto	—



* 헬퍼 메소드

var e = new Engine();

e.info(); → obj.info()

e.print(); → Engine.prototype.print()

e.test(); → Object.prototype.test()

Engine() {
Object.call();
this.info = function(){};
}

↓
서브자리에

Object() test(x)

↑
Engine() print()

↑
obj
info()

* မျိုး၏ ဒုပ္ပ ။

