



JavaScript – Data Types

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

Contents

- ▶ 데이터 타입의 종류
- ▶ 식별자와 변수
- ▶ 변수 선언과 데이터 할당
- ▶ 기본형 데이터
- ▶ 참조형 데이터
- ▶ 변수 복사
- ▶ 객체 복사

데이터 타입의 종류

- ▶ Primitive type: number, string, Boolean, null, undefined, symbol
- ▶ Reference type = object: array, function, Date, RegExp, Map, WeakMap, Set, Weak-Set

식별자와 변수

- ▶ 변수 (variable): 값이 변할 수 있는 데이터
- ▶ 식별자 (identifier): 데이터를 식별하는 데 사용하는 이름, 변수명

변수 선언과 데이터 할당

▶ 변수 선언

```
let a;
```

주소	...	1002	1003	1004	1005	...
데이터			이름: a 값:			

변수 선언과 데이터 할당

▶ 데이터 할당

```
let a;  
a = 'abc';
```

```
let a = 'abc';
```

변수 영역	주소	...	1002	1003	1004	1005	...
	데이터			이름: a 값: @5004			
데이터 영역	주소	...	5002	5003	5004	5005	...
	데이터				'abc'		

변수 선언과 데이터 할당

- ▶ 데이터 조작: 'abc' 다음에 'def' 추가

`a = 'abcdef';`

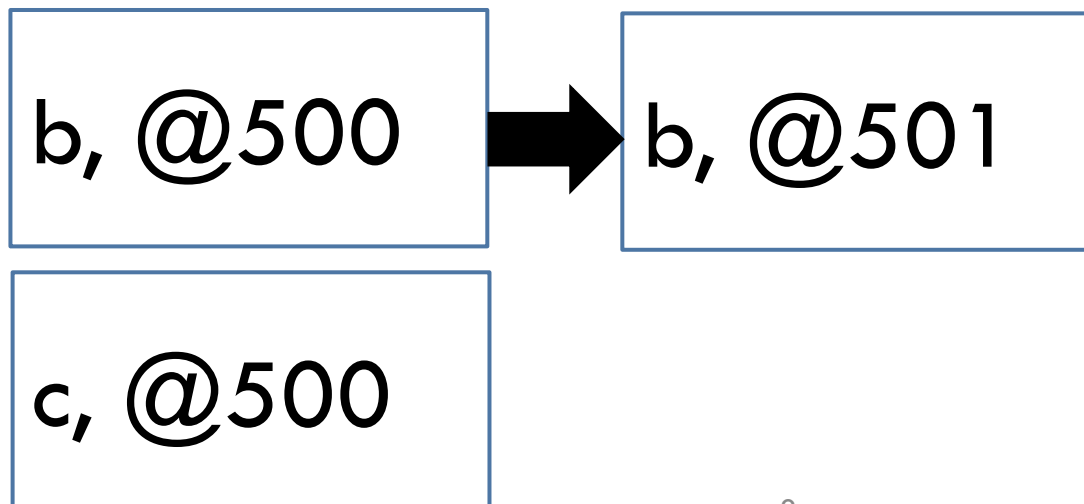
`a = 'abc' + 'def'`

주소	...	1002	1003	1004	1005	...
데이터			이름: a 값: @5005	변경됨		
주소	...	5002	5003	5004	5005	...
데이터				'abc'	'abcdef'	

기본형 데이터

▶ 불변성 (immutable)

```
let b = 5;  
let c = 5;  
b = 7;
```



500	501
5	7

참조형 데이터

- ▶ 참조형 데이터 변수 할당
 - ▶ 객체의 변수(property) 영역이 별도로 존재함

```
let obj1 = {  
  a: 1,  
  b: 'bbb'  
};
```

변수 영역	주소	1001	1002	1003	1004	...
	데이터		이름: obj1 값: @5001			
데이터 영역	주소	5001	5002	5003	5004	...
	데이터	@7103 ~ ?		1	'bbb'	
객체 @5001의 변수 영역	주소	7103	7104	7105	7106	...
	데이터	이름: a 값: @5003	이름: b 값: @5004			

참조형 데이터

- ▶ 참조형 데이터 프로퍼티 재할당

obj1.a = 2;

변수 영역	주소	1001	1002	1003	1004	1005	...
	데이터		이름: obj1 값: @5001	← 변하지 않음			
데이터 영역	주소	5001	5002	5003	5004	5005	...
	데이터	@7103 ~ ?		1	'bbb'	2	
객체 @5001의 변수 영역	주소	7103	7104	7104	7106	7107	...
	데이터	이름: a 값: @5005	이름: b 값: @5004				

참조형 데이터

▶ 중첩 객체

```
let obj = {  
  x: 3,  
  arr: [3, 4, 5]  
};
```

변수 영역	주소	1001	1002	1003	1004	1005	...
	데이터		이름: obj 값: @5001				
데이터 영역	주소	5001	5002	5003	5004	5005	...
	데이터	@7103 ~ ?	3	@8104 ~ ?	4	5	

객체 @5001의 변수 영역

주소	7103	7104	...
데이터	이름: x 값: @5002	이름: arr 값: @5003	

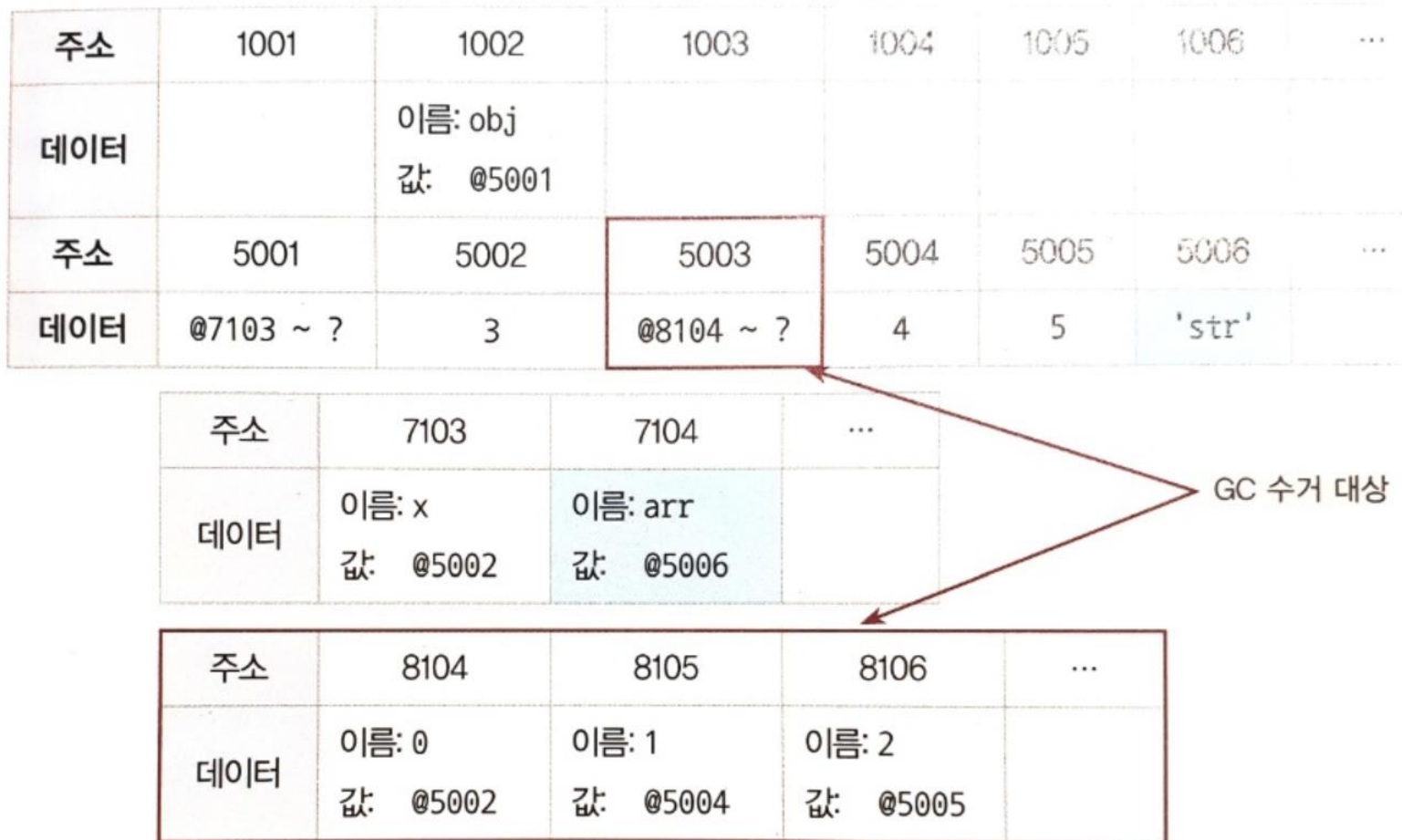
배열 @5003의 변수 영역

주소	8104	8105	8106	...
데이터	이름: 0 값: @5002	이름: 1 값: @5004	이름: 2 값: @5005	

참조형 데이터

- ▶ 중첩 객체에서의 프로퍼티 재할당

`obj.arr = 'str';`



변수의 복사 비교

▶ 변수를 복사할 때의 변화

```
let a = 10;
```

```
let b = a;
```

```
let obj1 = { c: 10, d: 'ddd' };
```

```
let obj2 = obj1;
```

변수 영역	주소	1001	1002	1003	1004	...
	데이터	이름: a 값: @5001	이름: b 값: @5001	이름: obj1 값: @5002	이름: obj2 값: @5002	
데이터 영역	주소	5001	5002	5003	5004	...
	데이터	10	@7103 ~ ?	'ddd'		

객체 @5002의 변수 영역	주소	7103	7104	...
	데이터	이름: c 값: @5001	이름: d 값: @5003	

변수의 복사 비교

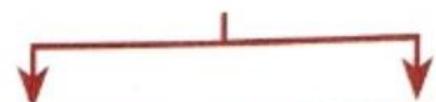
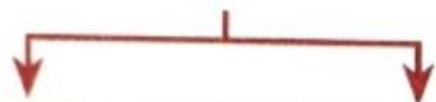
- ▶ 변수를 복사 이후 값 변경 - 객체 프로퍼티 변경

```
b = 15;  
obj2.c = 20;
```

```
a !== b;  
obj1 === obj2;
```

값이 달라짐

값이 달라지지 않음



주소	1001	1002	1003	1004	1005	
데이터	이름: a 값: @5001	이름: b 값: @5004	이름: obj1 값: @5002	이름: obj2 값: @5002		
주소	5001	5002	5003	5004	5005	
데이터	10	@7103 ~ ?	'ddd'	15	20	

주소	7103	7104	...
데이터	이름: c 값: @5005	이름: d 값: @5003	

변수의 복사 비교

- ▶ 객체 자체를 변경하였을 때

```
let a = 10;  
let b = a;  
let obj1 = { c: 10, d: 'ddd' };  
let obj2 = obj1;
```

```
b = 15;  
obj2 = { c: 20, d: 'ddd' };
```

값이 달라짐

값이 달라짐

주소	1001	1002	1003	1004	1005	1006	...
데이터	이름: a 값: @5001	이름: b 값: @5004	이름: obj1 값: @5002	이름: obj2 값: @5006			
주소	5001	5002	5003	5004	5005	5006	...
데이터	10	@7103 ~ ?	'ddd'	15	20	@8204 ~ ?	

주소	7103	7104	...
데이터	이름: c 값: @5001	이름: d 값: @5003	

주소	8204	8205	...
데이터	이름: c 값: @5005	이름: d 값: @5003	

객체 복사

▶ 객체 복사 예

```
let user = {  
  name: 'Sung',  
  gender: 'male'  
};  
let changeName = function (user, newname) {  
  let newUser = user;  
  newUser.name = newname;  
  return newUser;  
}
```

```
let user2 = changeName(user, 'Sung-Dong');  
if (user !== user2) {  
  console.log('유저 정보가 변경되었음!');  
}  
console.log(user.name, user2.name);  
console.log(user === user2);
```

객체 복사

▶ 객체 복사 개선 예

```
let user = {  
  name: 'Sung',  
  gender: 'male'  
};  
let changeName = function (user, newname) {  
  return {  
    name: newname,  
    gender: user.gender  
  };  
}
```

```
let user2 = changeName(user, 'Sung-Dong');  
if (user !== user2) {  
  console.log('유저 정보가 변경되었음!');  
}  
console.log(user.name, user2.name);  
console.log(user === user2);
```

객체 복사

- ▶ 객체의 모든 프로퍼티를 복사하는 함수

```
let copyObject = function(target) {  
  let result = {};  
  for (let prop in target) {  
    result[prop] = target[prop]  
  }  
  return result;  
}
```

```
let user2 = copyObject(user);  
user2.name = 'Sung-Dong';  
  
if (user !== user2) {  
  console.log('유저 정보가 변경되었음!');  
}  
console.log(user.name, user2.name);  
console.log(user === user2);
```

Source

- ▶ 코어 자바스크립트, 정재남 저, 위키북스