



05. 불변객체

김서영 백수연





목차



1. 기본형과 참조형의 복사

2. 불변 객체를 만드는 방법

a. 중첩된 객체에 대한 깊은 복사

b. 객체의 깊은 복사를 수행하는 범용 함수

c. JSON 활용





기본형과 참조형의 복사



기본형과 참조형의 복사

code

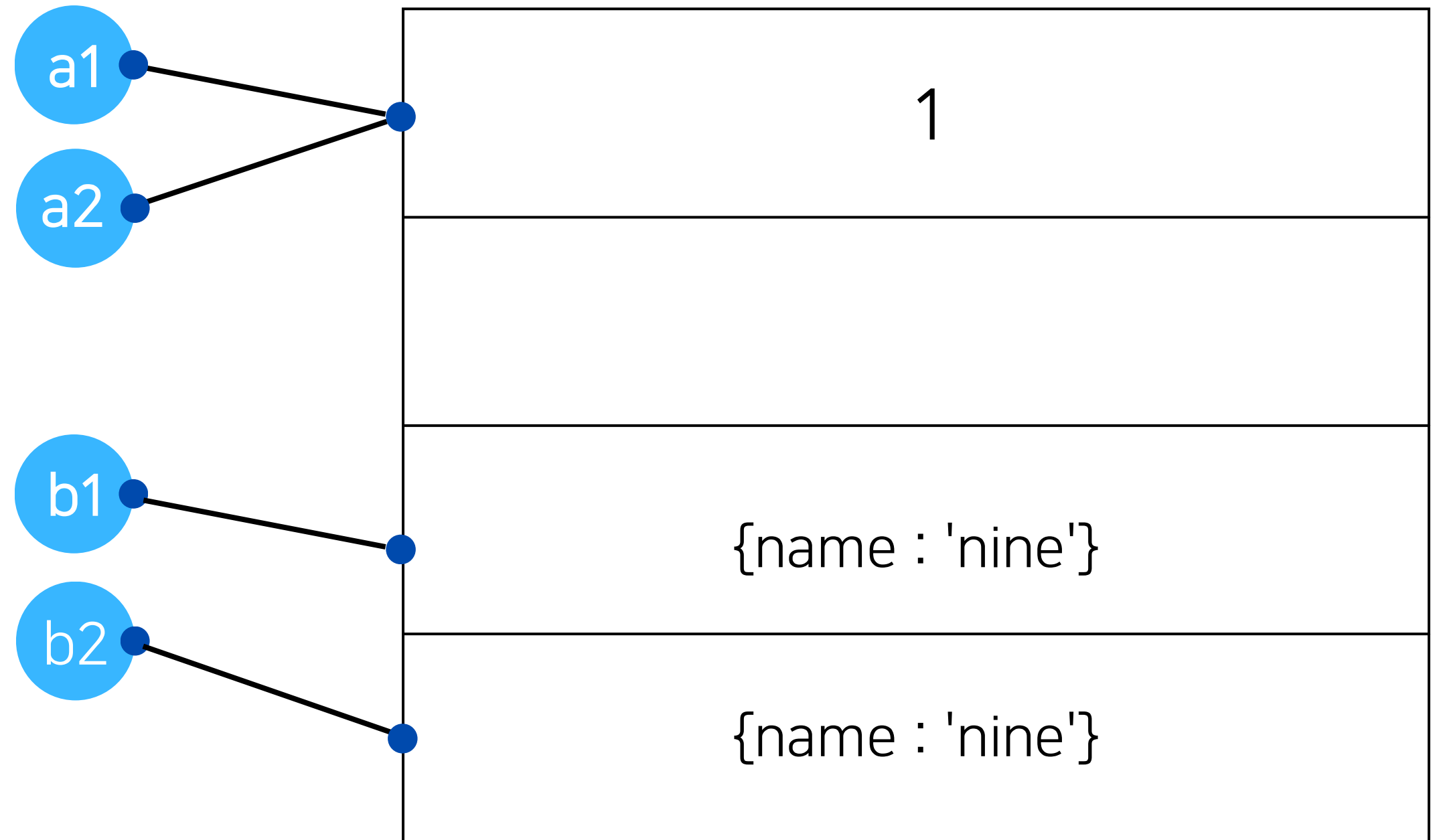
```
var a1 = 1  
var a2 = 1
```

a1 === a2
true

```
var b1 = {name : 'nine'}  
var b2 = {name : 'nine'}
```

b1 === b2
false

Memory



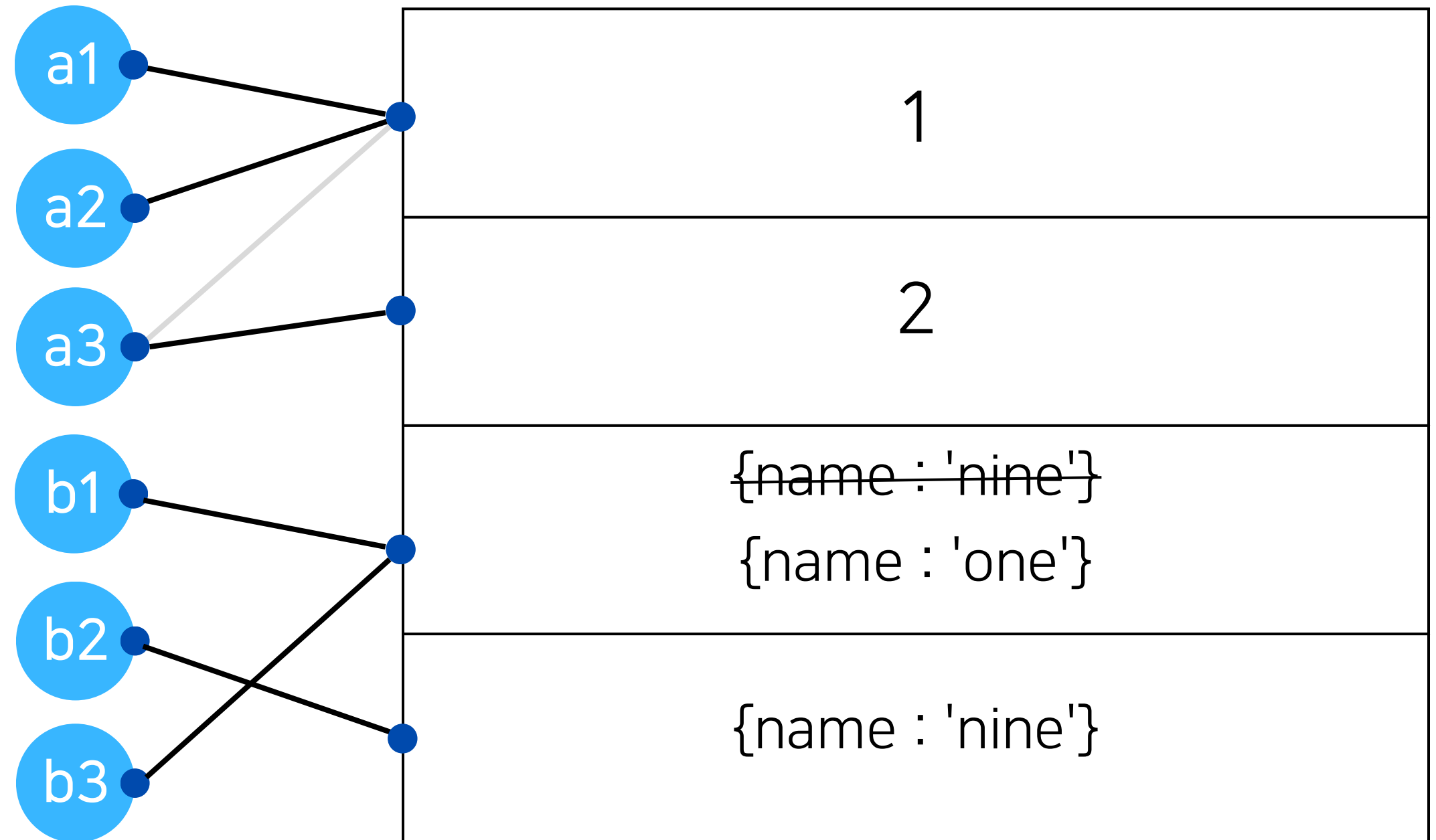
기본형과 참조형의 복사

code

```
var a1 = 1  
var a2 = 1  
var a3 = a1  
var a3 = 2
```

```
var b1 = {name : 'nine'}  
var b2 = {name : 'nine'}  
var b3 = b1  
b3.name = 'one'
```

Memory





당황해서 저렇게 마셨

불변 객체를 만드는 방법



기본형과 참조형의 복사

```
let user = {
  name : 'nine',
  age : 9
};

function changeName( user , newName ) {
  let newUser = user;
  newUser.name = newName;
  return newUser ;
}

let user2 = changeName(user, 'one') ;

if (user !== user2) {
  console.log("당신은 9조가 아닙니다. 나가주세요.");
}

console.log(user.name, user2.name); // one one
console.log(user === user2); // true
```

```
let user = {
  name : 'nine',
  age : 9
};

function changeName( user, newName ) {
  return {
    name : newName ,
    age : user.age
  };
}

let user2 = changeName(user, 'one') ;

if (user !== user2) {
  console.log("당신은 9조가 아닙니다. 나가주세요.");
  // 당신은 9조가 아닙니다. 나가주세요.
}


console.log(user.name, user2.name); // nine one
console.log(user === user2); // false
```

중첩된 객체에 대한 얕은 복사

```
var teamMSH = {  
  job: "youtuber",  
  member: {  
    name: "doong",  
    job: "princess"  
  }  
};  
  
var teamBDNS = copyObject(teamMSH);  
  
teamBDNS.job = 'developer';  
console.log(teamMSH.name === teamBDNS.name);  
  
teamMSH.memeber.name = 'seoyoung';  
console.log(teamMSH.member.name === teamBDNS.member.name);  
  
teamBDNS.member.job = "";  
console.log(teamMSH.member.job === teamBDNS.member.job);
```


방법1. 중첩된 객체에 대한 깊은 복사 (copyObject 함수)

```
var teamBDNS = copyObject(teamMSH);  
teamBDNS.member = copyObject(teamBDNS.member);  
  
teamMSH.member.name = 'seoyoung';  
console.log(teamMSH.member.name === teamBDNS.member.name);  
  
teamBDNS.member.job = '';  
console.log(teamMSH.member.job === teamBDNS.member.job);
```



방법2. 객체의 깊은 복사를 수행하는 범용 함수

```
var copyObjectDeep = function(target) {  
  var result = {};  
  if(typeof target === 'object' && target !== null) {  
    for (var prop in target) {  
      result[prop] = copyObjectDeep(target[prop]);  
    }  
  }  
  else {  
    result = target;  
  }  
  return result;  
};
```

JSON 활용

```
const obj = {  
  a: 1,  
  b: {  
    c: 9,  
  },  
}  
  
const newObj = JSON.parse(JSON.stringify(obj));  
  
newObj.b.c = 8;  
  
console.log(obj);           // {a, 1, b: {c: 9}}  
console.log(newObj);         // {a, 1, b: {c: 8}}  
console.log(obj.b.c === newObj.b.c); // false
```

stringify() parse() 메서드 활용

```
const obj = {  
  a: 1,  
  b: {  
    c: 9,  
  },  
  func: function() {  
    return this.a;  
  }  
}  
  
const newObj = JSON.parse(JSON.stringify(obj));  
  
console.log(newObj.func); // undefined
```

JSON 문법의 단점

1. 성능 문제
2. JSON.stringify 메서드는 함수를 만나면 undefined로 처리

내일 봐요
공주들

목요일은 목이 빠지게 웃는 날!