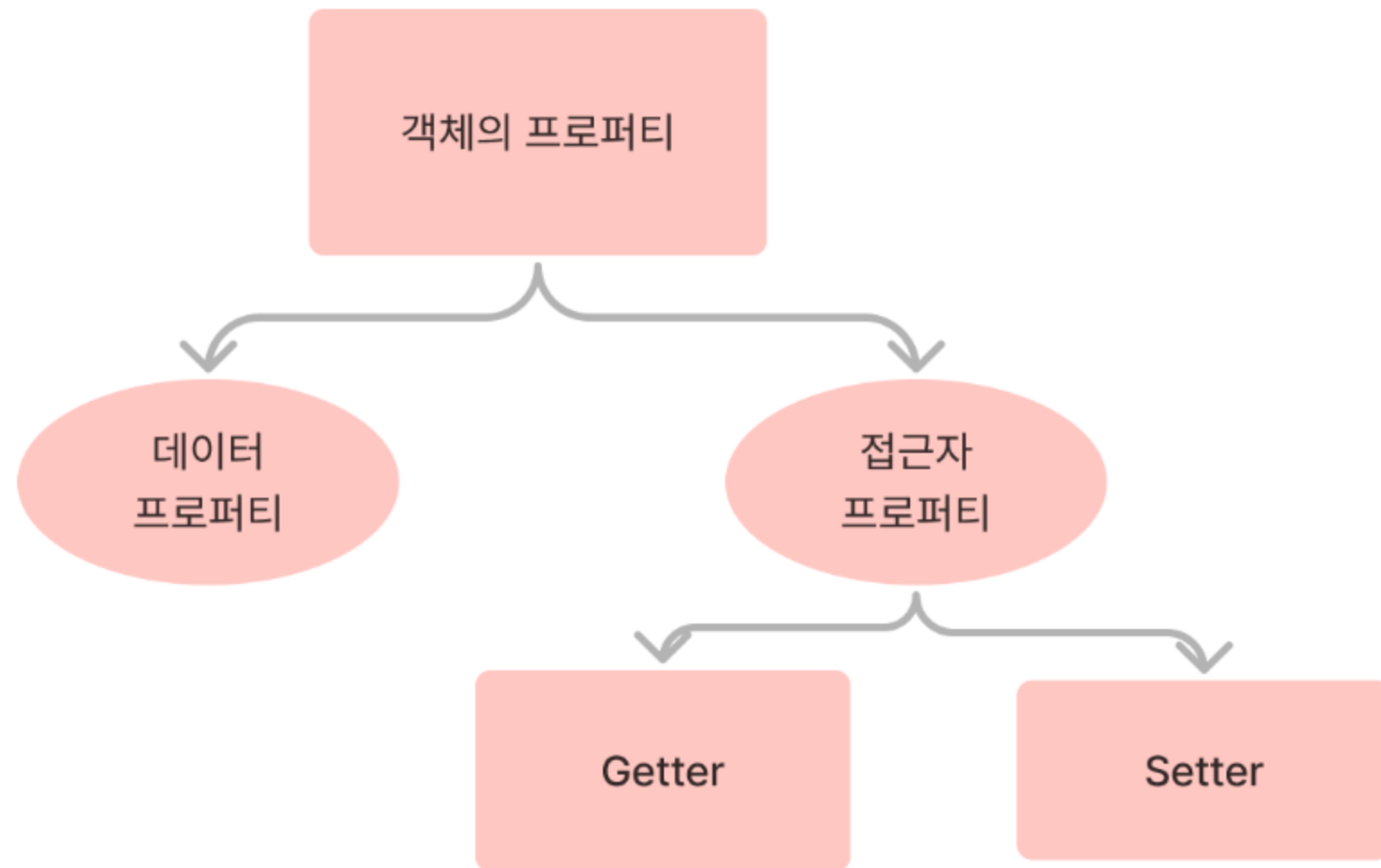


| getter와 setter

| 이은주

2023.05.21

getter setter



```
1  const aa = {  
2      get propertyName(){ // getter 로직  
3      },  
4      set propertyName(value){ // setter 로직  
5      }  
6  };
```

getter setter

```
let user = {
  firstName: "eun",
  secondName: "joo",

  get fullName() {
    return `${this.firstName} ${this.secondName}`;
  }
};

console.log(user.fullName);
eun joo
```

- getter는 객체의 속성 값을 가져오는데 사용
- 속성 이름 앞에 get 키워드 사용하여 정의

```
let user = {
  firstName: "eun",
  secondName: "joo",

  get fullName() {
    return `${this.firstName} ${this.secondName}`;
  },

  set fullName(value) {
    [this.firstName, this.secondName] = value.split(" ");
  }
};


user.fullName = "lee eunjoo";
console.log(user.firstName);
console.log(user.secondName);
lee VM36:15
eunjoo VM36:16
```

- setter는 객체의 속성 값을 설정하는데 사용
- 속성 이름 앞에 set 키워드를 사용하여 정의

접근자 프로퍼티 설명자

접근자 프로퍼티 설명자 객체의 속성

- get : 접근자 프로퍼티의 getter 함수를 정의
- set : 접근자 프로퍼티의 setter 함수를 정의
- enumerable : 프로퍼티가 열거 가능한지 여부를 나타냄. 기본값은 false
- configurable : 프로퍼티의 설정 가능 여부를 나타냄. 기본값은 false
- getPrototypeOf : 프로퍼티의 프로토타입 객체를 반환하는 함수
- setPrototypeOf : 프로퍼티의 프로토타입 객체를 설정하는 함수
- has : 프로퍼티가 존재하는지 여부를 나타내는 함수
- hasOwn : 객체 자체에 프로퍼티가 있는지 여부를 나타내는 함수
- deleteProperty : 프로퍼티를 삭제하는 함수
- ownKeys : 객체의 모든 프로퍼티 키 배열을 반환하는 함수



```
1  const aa = {};  
2  
3  Object.defineProperty(aa, "propertyName", {  
4    get: function () {},  
5    set: function (value) {},  
6    enumerable: true,  
7    configurable: true,  
8  });  
9
```