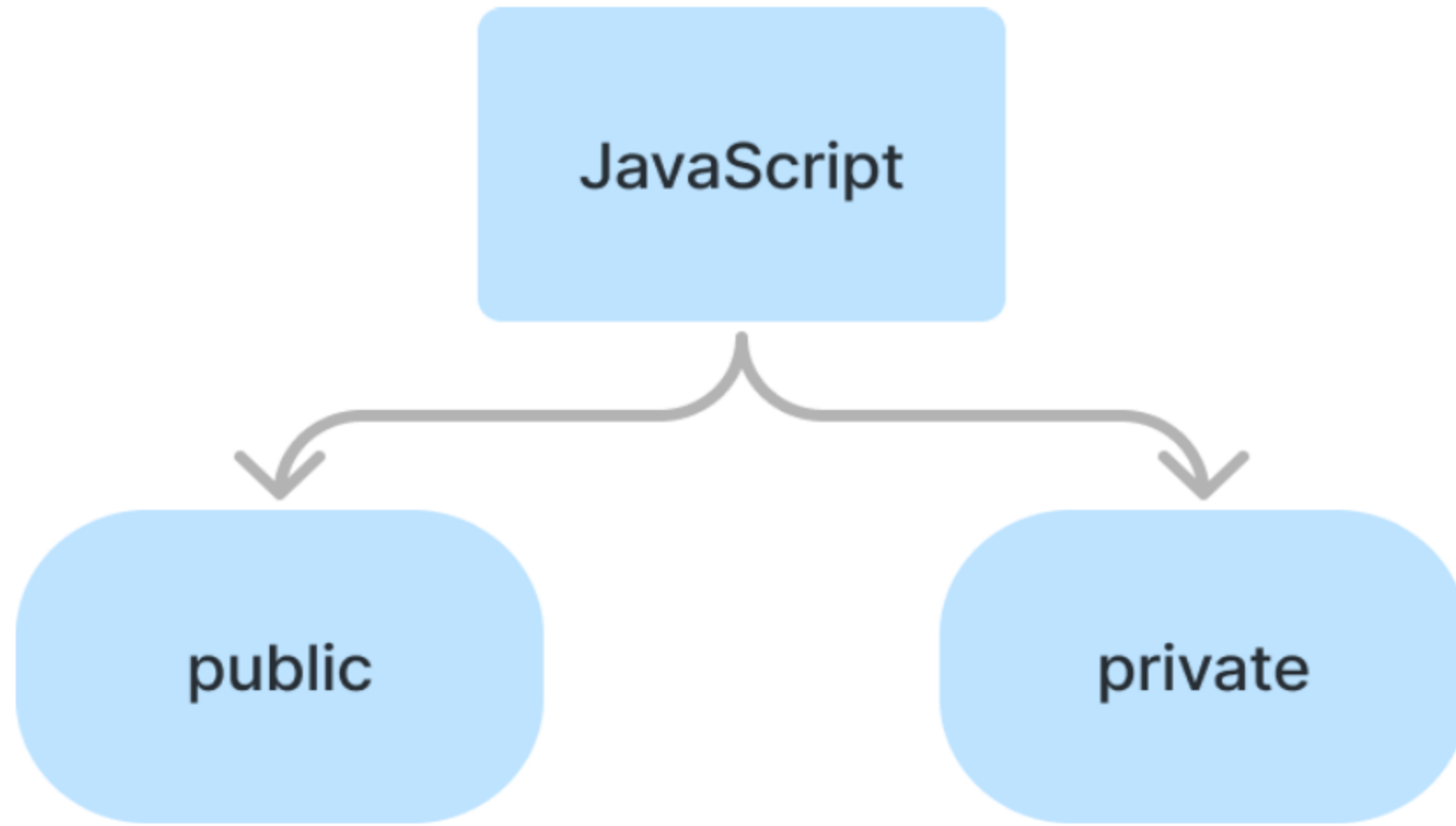


**I private,  
protected**

이은주

2023.05.27

# public, private



클래스 인스턴스의 속성으로 외부에서 직접 접근하고 조작 가능

클래스의 내부 메서드에서만 접근하고 조작할 수 있으며 외부에서 직접 접근 불가능



자바스크립트는 protect를 직접 지원X

# class



```
1 class Square {  
2     constructor(length) {  
3         this.length = length;  
4     }  
5  
6     getArea() {  
7         return this.length * this.length; // 넓이  
8     }  
9  
10    getPerimeter() {  
11        return 4 * this.length;  
12    }  
13 }  
14  
15 const square = new Square(10);  
16 console.log(square.getArea());  
17 console.log(square.getPerimeter());
```

---

100

---

40

---

# 음수 제어

```
1 class Square {
2   constructor(length) {
3     if (length <= 0) {
4       throw new Error("길이는 양수여야 합니다.");
5     }
6     this.length = length;
7   }
8
9   getArea() {
10    return this.length * this.length;
11  }
12
13  getPerimeter() {
14    return 4 * this.length;
15  }
16 }
17
18 const square = new Square(10);
19 console.log(square.getArea()); // 출력: 100
20 console.log(square.getPerimeter()); // 출력: 40
```

제약을 추가하여 음수 길이를 가진 경우 에러를 발생시키기

제약을 추가함으로써 클래스 외부에서 잘못된 값을 사용하는 경우를 방지

# 클래스 외부

```
1 class Square {
2   constructor(length) {
3     if (length <= 0) {
4       throw new Error("길이는 양수여야 합니다.");
5     }
6     this.length = length;
7   }
8
9   getArea() {
10    return this.length * this.length;
11  }
12
13  getPerimeter() {
14    return 4 * this.length;
15  }
16 }
17
18 const square = new Square(10);
19 console.log(square.getArea()); // 출력: 100
20 console.log(square.getPerimeter()); // 출력: 40
21
22 square.length = -10;
23 console.log(square.getArea());
24 console.log(square.getPerimeter());
```

100

40

100

-40

클래스 외부에서 음수를 지정하게 되면 잘 작동하게 됨

# protected



protected

속성 앞에 밑줄을 넣기  
자바스크립트에는 protected 문법 없음



private

# 기호를 앞에 붙이기  
클래스 외부에서의 직접적인 접근을 차단

# protected, private

```
class Square {
  #length;
  constructor(length) {
    if (length < 0) {
      throw "길이는 0이상이어야 함";
    }
    this.#length = length;
  }

  getArea() {
    return this.#length * this.#length; // 넓이
  }
  getPerimeter() {
    return 4 * this.#length;
  }
}

const square = new Square(10);
console.log(square.getArea());
console.log(square.getPerimeter()); // 잘 작동

square.#length = -10;    '#length' 속성은 프라이빗 식별자를 포함하기 때문에 'Square' 클래스 외부에서 액세스할 수 없습니다.
console.log(square.getArea());
console.log(square.getPerimeter()); // 오류 발생
```

#length는 private 필드로 선언되어 클래스 외부에서의 접근을 차단

코드 실행 시 음수 값을 할당하려고 하면 오류가 발생하여 잘못된 값을 사용할 수 없도록 만들어 줌

# getter, setter

```
1 class Square {
2   #length;
3   constructor(length) {
4     if (length < 0) {
5       throw "길이는 0이상이어야 함";
6     }
7     this.#length = length;
8   }
9
10  getArea() {
11    return this.#length * this.#length; // 넓이
12  }
13  getPerimeter() {
14    return 4 * this.#length;
15  }
16  setLength(length) {
17    if (length < 0) {
18      throw "길이는 0이상이어야 함";
19    }
20    this.#length = length;
21  }
22  getLength() {
23    return this.#length;
24  }
25 }
26 const square = new Square(10);
27 console.log(square.getArea());
28 console.log(square.getPerimeter());
29
30 console.log(square.getLength());
31 square.setLength(20);
32 console.log(square.getLength());
33
```

---

100

---

40

---

10

---

20

setLength : 외부에서 length 속성을 변경하기  
위한 메서드  
getLength : length 속성 값을 가져오는 getter  
메서드