

2022/5/11

1. Closure

- 폐쇄된 공간 안의 데이터에 접근하기 위한 테크닉
- 내부 함수가 외부 함수에 접근하기 위한 테크닉
- 변수 은닉 and 메모리/코드 효율 극대화를 위해 사용 -> 클래스의 역할과 비슷하다고 볼 수 있음
- 함수에 별명을 할당하는 alias의 역할과도 비슷하다고 생각함
- 함수의 인자를 지정해서 변수에 할당할 수 있기 때문에 함수 내부의 내용을 드러내지 않을 수 있으며, 추후에 재사용 가능 및 코드의 반복을 줄일 수 있음

2. this.call() vs apply() vs bind()

2.1. this.call()

- function에 인자를 전달할 때 쉼표로 분리해서 전달 (ex. `function.call(this자리에 들어갈 변수, a, b, c);`)
- 함수에 인자를 적용한 후 즉시 실행

2.2. this.apply()

- function에 인자를 전달할 때 배열로 전달 (ex. `function.call(this자리에 들어갈 변수, [a, b, c]);`)
- 함수에 인자를 적용한 후 즉시 실행

2.3. this.bind()

- function에 인자를 전달하고 그게 적용된 후의 함수를 새롭게 반환
- 함수를 즉시 실행하지 않으므로 반환된 함수를 다른 변수에 저장 후 필요할 때 호출할 수 있음 (ex. `let binded = function.bind(this자리에 들어갈 변수, a); -> binded(b, c);`로 사용 가능)
- 추후에 이미 바인딩된 인자 대신 새로운 인자를 전달할 수도 있음 (ex. `binded(this자리에 들어갈 변수2, a2);`)