

JSArrowFunction

arrowFunction

- 인자가 없을 때

```
const f= function(){  
  return '33'}  
const arrowFunc={()=>{return'33'}}  
같은 결과를 불러옴
```

- 인자가 있을 때

```
const add100 = function (val) { return 100 + val }  
  
console.log("add100(3): " + add100(3))  
  
const arrow100 = (val) => { return 100 + val }  
  
console.log("arrow100(3): " + arrow100(3))
```

=>같은 값을 불러옴

```
let add100= (val)=>100+val  
console.log(add100(4))  
let add100= (val)=>{ return 100+val}  
console.log(add100(4))  
와 동일함 {}중괄호랑 return 없애도 됨
```

JS익명 객체 만드는 법(<https://sas-study.tistory.com/99>)

익명 객체라서 중괄호 벗어나면 기능을 못하게 됨

```
const calc = {  
  // 클래스 멤버 변수 만드는 작업 같은것이라고 생각하면됨  
  first: 33,  
  second: 777,  
  
  // 현재 케이스처럼 this.first는 calc의 내부 정보가되는데  
  // 케이스 바이 케이스 형태로 다뤄 보는 것이 가장 좋다.  
  print: function () {  
    const add = () => {  
      console.log(this)  
      return this.first + this.second  
    }  
  
    console.log(this.first + " + " + this.second + " = " + add())  
  }  
}  
calc.print()
```

```
▼ {first: 33, second: 777, print: f} 3  
  first: 33  
  ► print: f ()  
    second: 777  
  ► [[Prototype]]: Object  
33 + 777 = 810
```

spread 와 연동

설명:

현재 args는 spread로 인하여

[3,4,5]인 배열로 되어있다.

따라서,for문 내부에서 sum값(3)에
args의 배열 element들을 더하는 작업이
이루어진다.

결과적으로 add(1,2,3,4,5)는

1~5까지 다 더한 값을 뽑아내게 된다.

```
1 2 ▼ (3) [3, 4, 5]
    0: 3
    1: 4
    2: 5
    length: 3
    ▶ [[Prototype]]: Array(0)
sum = 15
```

```
const add = (first, second, ... args) => {
  console.log(first, second, args)

  let sum = first + second

  for (let i = 0; i < args.length; i++) {
    sum += args[i]
  }

  return sum
}

let sum = add(1, 2, 3, 4, 5)

console.log("sum = " + sum)
```

JSMAP

(<https://velog.io/@daybreak/javascript-map%ED%95%A8%EC%88%98>)

.get(key)->key값을 통해 value값 호출

.has(key)->key값이 있는가? true/false로 return

.delete(key)->특정 Key-value값만 삭제

.clear()->인자들 싹 다 삭제

선언 인자가 없을 때

```
let map = new Map()

console.log("map: " + map)
```

```
map: [object Map]
```

인자가 있을 때

```
let map2 = new Map([["Apple", "green"], ["Banana", "yellow"]])
```

```
console.log("map2: " + map2)
console.log("map2 size: " + map2.size)

console.log("map2.get('test1'): " + map2.get("test1"))
console.log("map2.get('Banana'): " + map2.get("Banana"))
console.log("map2.has('test2'): " + map2.has("test2"))
console.log("map2.has('Apple'): " + map2.has("Apple"))
```

```
map2: [object Map]
map2 size: 2
map2.get('test1'): undefined
map2.get('Banana'): yellow
map2.has('test2'): false
map2.has('Apple'): true
```

<key,value>값은? new Map내부에서만 해 준다

key값만 추출하고 싶을 때(<https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Statements/for...of>)

for(let key of map.keys())

```
let map3 = new Map([["Apple", "green"], ["Banana", "yellow"]])

let keyIter = map3.keys()

for (let key of keyIter) {
  console.log("key: " + key)
}
```

key: Apple

key: Banana

value값만 추출하고 싶을 때

for(let value of map名.values())

```
let valueIter = map3.values()

for (let value of valueIter) {
  console.log("value: " + value)
}
```

value: green

value: yellow

entry값(key,value둘 다) 추출하고 싶을 때

```
let entryIter = map3.entries()

for (let entry of entryIter) {
  console.log("entry: " + entry)
}
```

entry: Apple,green

entry: Banana,yellow

OR forEach사용

```
map3.forEach(function(value, key) {
  console.log(key + " : " + value)
})
```

Apple : green

Banana : yellow

수업 후 arrow로 바꿔보자

return key + ":" + value는 되려나?

Set

선언 (인자 없을 때)

```
let set = new Set()

console.log("set: " + set)
```

있을 때

```
let set2 = new Set(["Banana", "Watermelon"])
```

그 외 요소

.add(element) 요소 추가

.has(element) 해당 element를 가지고 있냐? true/false로 return

.clear() 요소 전부 삭제

for...of 명령문으로 요소 뽑아내기 map과 동일

```
for (const element of object) {  
  console.log(element);  
}
```

```
set2.add("Cherry")  
set2.add("Grape")  
  
console.log(set2.has("Grape"))  
console.log(set2.has("Strawberry"))  
  
set2.clear()  
console.log(set2.has("Grape"))
```

```
let setKeyIter = set2.keys()  
for (let key of setKeyIter) {  
  console.log("key: " + key)  
}  
  
let setEntryIter = set2.entries()  
for (let entry of setEntryIter) {  
  console.log("entry: " + entry)  
}  
  
set2.forEach(function(val1, val2) {  
  console.log(val1 + " : " + val2)  
})
```