

new Function 문법

new Function

클로저

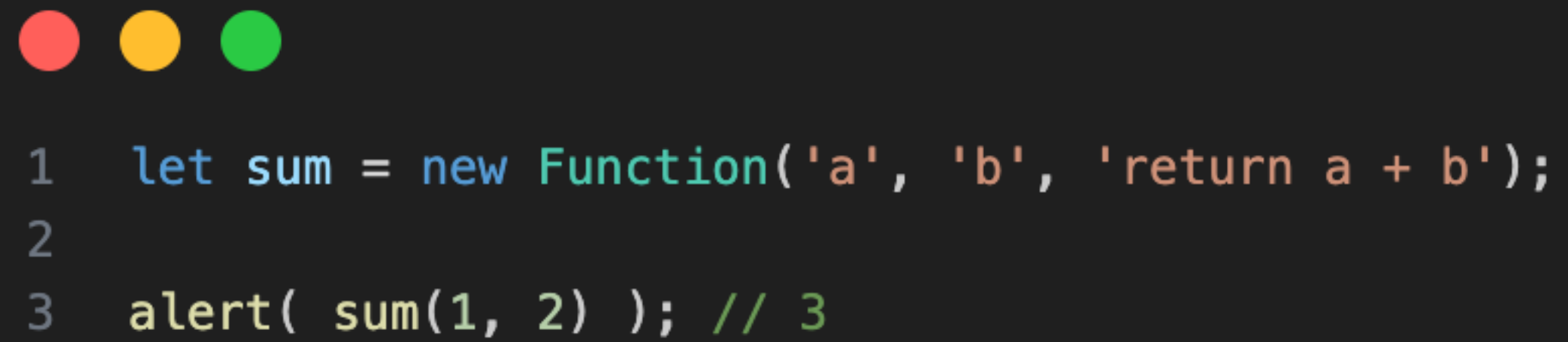
new Function



```
1 let func = new Function ([arg1, arg2, ...argN], functionBody);
```

새로 만들어지는 함수는 인수 `arg1...argN`과 함수 본문 `functionBody`로 구성됩니다.

new Function



```
1  let sum = new Function('a', 'b', 'return a + b');  
2  
3  alert( sum(1, 2) ); // 3
```

sum()은 인자 두개 (a,b)를 받아 둘의 합을 반환하는 함수

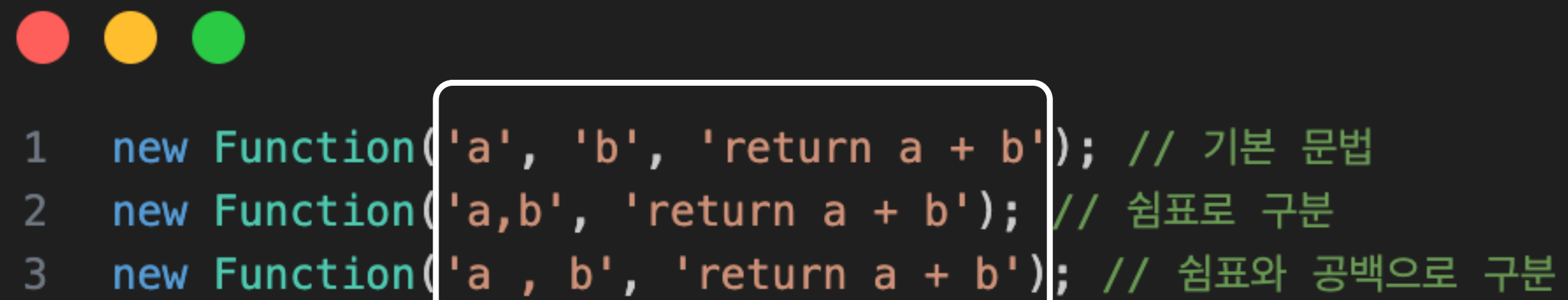
new Function



```
1 new Function('a', 'b', 'return a + b'); // 기본 문법
2 new Function('a,b', 'return a + b'); // 쉼표로 구분
3 new Function('a , b', 'return a + b'); // 쉼표와 공백으로 구분
```

3가지 모두 같은 함수를 만드는 방법

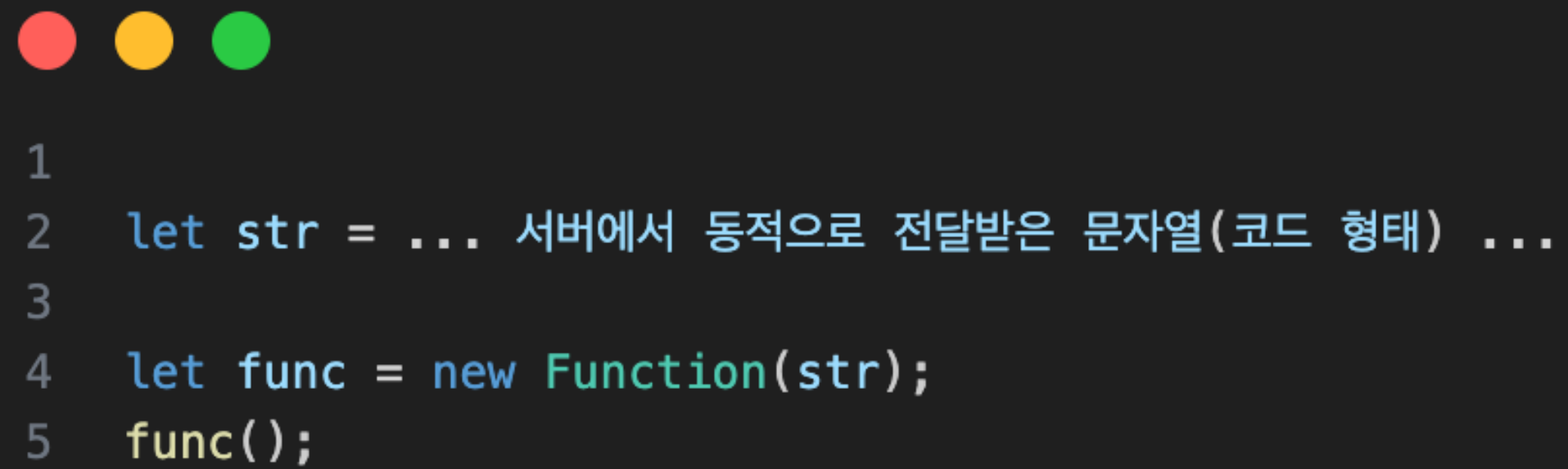
new Function



```
1 new Function('a', 'b', 'return a + b'); // 기본 문법
2 new Function('a,b', 'return a + b'); // 쉼표로 구분
3 new Function('a , b', 'return a + b'); // 쉼표와 공백으로 구분
```

런타임에 받은 문자열을 사용해 함수를 만들 수 있다!

-> 서버에서 전달받은 문자열을 이용해 새로운 함수를 만들고 이를 실행하는 것도 가능

new Function**사용하는 경우**

```
1  
2 let str = ... 서버에서 동적으로 전달받은 문자열(코드 형태) ...  
3  
4 let func = new Function(str);  
5 func();
```

1. 서버에서 코드를 받거나 템플릿을 사용해 함수를 동적으로 컴파일해야 하는 경우
2. 복잡한 웹 애플리케이션을 구현할 때와 같이 아주 특별한 경우

클로저 closure

자바스크립트에서 함수는 `[[Environment]]`를 통해 자신이 어디서 만들어졌는지(렉시컬 환경)를 참조
함수 본문에선 `[[Environment]]`를 사용해 외부 변수에 접근합니다.

JS의 함수는 모두 클로저 (외부 변수를 기억하고 이 외부 변수에 접근할 수 있는 함수)

클로저 closure

자바스크립트에서 함수는 `[[Environment]]`를 통해 자신이 어디서 만들어졌는지(렉시컬 환경)를 참조
함수 본문에선 `[[Environment]]`를 사용해 외부 변수에 접근합니다.

JS의 함수는 모두 클로저 (외부 변수를 기억하고 이 외부 변수에 접근할 수 있는 함수)

하지만

`new Function` 함수 -> `[[Environment]]` 프로퍼티가 현재 렉시컬 환경이 아닌 **전역 렉시컬 환경**을 참조
외부 변수에 접근할 수 없고, 오직 전역 변수에만 접근할 수 있습니다.

즉 `new Function` 문법을 사용한 함수는 클로저가 X!

클로저 closure

1. 일반적 문법의 함수

```
1 function getFunc() {  
2     let value = "test";  
3  
4     let func = function() { alert(value); };  
5  
6     return func;  
7 }  
8  
9 getFunc(); // getFunc의 렉시컬 환경에 있는 값 "test"가 출력됩니다.
```

외부 렉시컬 환경의 변수 사용 가능

클로저 closure

2. new Function 문법의 함수

```
1 function getFunc() {  
2     let value = "test";  
3  
4     let func = new Function('alert(value)');  
5  
6     return func;  
7 }  
8  
9 getFunc(); // ReferenceError: value is not defined
```

외부 렉시컬 환경의 변수 사용 불가능
(전역변수만 사용가능)

에러를 예방해 준다는 관점에선 장점
매개변수를 사용해 값을 받는 게 더 낫다!

문자열을 사용해서 함수를 만들어야하는 상황 등
실무에선 이 기능이 아주 유용하게 쓰인다.

THANK YOU