

JavaScript

함수표현식

vs 함수선언문

유하은

Contents

1 면접 등장 질문

2 함수 선언문 ?

3 함수 표현식? 화살표 함수?

4 클로저

5 IIFE (즉시 주입 함수 표현식)

6 면접 등장 질문 내꺼만들기

👉 함수를 선언하는 방식 중 함수 선언문과 표현식은 어떤 차이가 있는지 설명해주세요

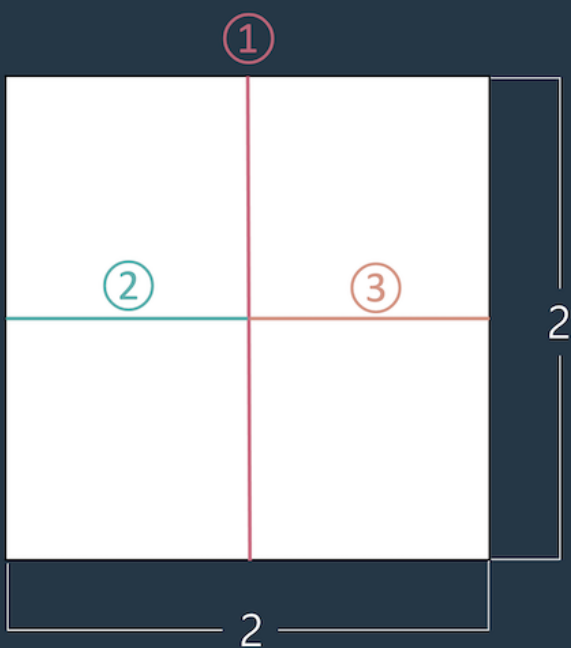


종이 자르기

dark light sublime vim emacs JavaScript

문제 설명

머쓱이는 큰 종이를 1 x 1 크기의 종이로 자르려고 합니다. 예를 들어 2 x 2 크기의 종이를 1 x 1 크기로 자르려면 최소 가위질 세 번이 필요합니다.



정수 M , N 이 매개변수로 주어질 때, $M \times N$ 크기의 종이를 최소로 가위질 해야하는 횟수를 return 하도록 solution 함수를 완성해보세요.

solution.js

```
1 function solution(M, N) {
2     var answer = 0;
3     return answer;
4 }
```

실행 결과

실행 결과가 여기에 표시됩니다.

질문하기 (5)

테스트 케이스 추가하기

🔒 다른 사람의 풀이

초기화

코드 실행

제출 후 채점하기



```
1 function answer() {  
2     return "이것은 함수입니다.";  
3 }  
4  
5 answer(); // "이것은 함수입니다."
```



```
1 var x = function (a, b) {return a * b};
```



```
1 function f () {}  
2 const f = () => {};
```



```
1 let sum = (a, b) => a + b;
```



```
1 let sum = function(a, b) {  
2     return a + b;  
3 };
```


1 함수 표현식



```
1 alert(answer()); // 에러 발생!  
2 var answer = function() { return 5; }
```

2 함수 선언문

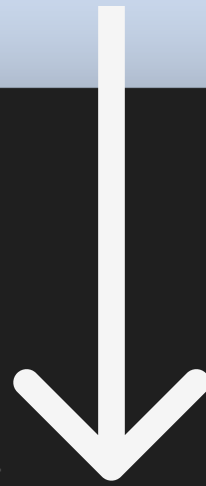


```
1 alert(answer());  
2 // Alerts 5.  
3 // 선언 전에 호출되도 정상 동작  
4  
5 function answer() { return 5; }
```

함수 선언문



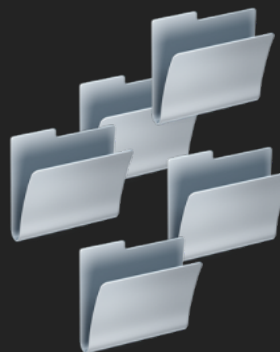
```
1 alert(answer());  
2 // Alerts 5.  
3 // 선언 전에 호출되도 정상 동작  
4  
5 function answer() { return 5; }
```



함수 선언문: 어디서든 호출할 수 있다.



```
1 alert(answer());  
2 // Alerts 5.  
3 // 선언 전에 호출되도 정상 동작  
4  
5 function answer() { return 5; }
```

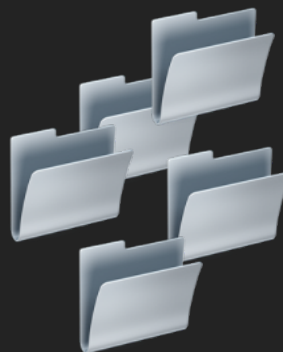


함수 선언문: 어디서든 호출할 수 있다.



```
1 alert(answer());  
2 // Alerts 5.  
3 // 선언 전에 호출되도 정상 동작  
4  
5 function answer() { return 5; }
```

Hoisting



함수 선언식과 함수 표현식은 함께 사용할 수는 있지만,
함수 표현식은 함수 이름이 필요없기에 가독성이 더 높은 장점이 있다.

- 클로저
- 인자 전달
- IIFE



```
1 function navsHandler(index) {  
2     return function navClickEvent(evt) {  
3         // 이벤트 코드  
4     }  
5 }  
6  
7 var navs = document.querySelectorAll('.nav')  
8 var i  
9  
10 for (i = 0; i < navs.length; i += 1) {  
11     navs[i].onclick = navsHandler(i)  
12 }
```



```
1 // 잘못된 예 1
2
3 var i
4
5 for (i = 0; i < list.length; i += 1) {
6     document.querySelector('#item' + i).onclick = function doSomething(ev
7     t) {
8         // i는 항상 list.length 값을 갖는다.
9     }
10 }
11
12 // 잘못된 예 2
13
14 var list = document.querySelectorAll('.item')
15 var i
16 var doSomething = function (evt) {
17     // 이 함수가 실행될 때까지 i의 값은 루프에 있던 값이 아니다.
18 };
19
20 for (i = 0; i < list.length; i += 1) {
21     item[i].onclick = doSomething;
22 }
```



```
1 var list = ['item1', 'item2', 'item3']
2 var i
3 var doSomethingHandler = function (itemIndex) {
4     return function doSomething(evt) {
5         // 클로저가 생성되어, itemIndex를 인자로 참조 할 수 있게 된다.
6         console.log(list[itemIndex]);
7     };
8 };
9
10 for (i = 0; i < list.length; i += 1) {
11     list[i].onclick = doSomethingHandler(i);
12 }
```



```
(function () {  
    // 코드 ...  
})();
```



```
1  var myModule = (function () {  
2      var privateMethod = function () {  
3          console.log('A private method');  
4      },  
5      someMethod = function () {  
6          console.log('A public method');  
7      },  
8      anotherMethod = function () {  
9          console.log('Another public method');  
10     };  
11  
12     return {  
13         someMethod: someMethod,  
14         anotherMethod: anotherMethod  
15     };  
16 }());
```

👉 함수를 선언하는 방식 중 함수 선언문과 표현식은 어떤 차이가 있는지 설명해주세요

👊 KEY WORD

1. 할당
2. 호이스팅

