

| 재귀와 스택



| 이은주

2023.05.15

재귀 함수

재귀 함수?

- 자기 자신을 호출하는 함수
- 종료조건이 반드시 필요
- 같은 구조의 함수를 반복 사용할 때 쓰임

스택?

- 데이터를 일시적으로 저장하는 자료구조
- 후입선출 원칙

재귀 함수



```
1 function factorial(n) {  
2   let result = 1;  
3  
4   for (let i = n; i >= 1; i--) {  
5     result *= i;  
6   }  
7  
8   return result;  
9 }
```



```
1 function factorial(n) {  
2   return n * factorial(n - 1);  
3 }  
4
```

재귀 함수



```
1 function factorial(n) {  
2   if (n === 0) {  
3     return 1;  
4   } else {  
5     return n * factorial(n - 1);  
6   }  
7 }
```

- 기본 조건과 재귀 호출로 구성
- 기본 조건은 재귀 호출을 멈추기 위한 조건으로 재귀함수가 무한히 호출되는 것을 방지

재귀 함수

재귀 함수의 깊이

- 처음 하는 호출을 포함한 중첩 호출의 최대 개수
- 자바스크립트 엔진은 호출 스택의 크기에 제한을 두고 있음
- 재귀 함수의 깊이를 너무 깊게 설정하거나 재귀 호출이 너무 많이 발생하면 스택 오버 플로우 오류 발생

재귀 함수



```
1 function factorial(n) {  
2   if (n === 0) {  
3     return 1;  
4   } else {  
5     return n * factorial(n - 1);  
6   }  
7 }
```



```
1 function factorial(n, accumulator = 1) {  
2   if (n === 0) {  
3     return accumulator;  
4   } else {  
5     return factorial(n - 1, n * accumulator);  
6   }  
7 }
```

- 함수가 호출될 때마다 스택에 새로운 프레임이 쌓이지 않고, 현재의 스택 프레임을 재활용하여 메모리 사용을 최적화 가능
- 재귀 함수의 깊이에 제한 X, 스택 오버플로우 오류를 방지