

예제로 알아보는

재귀 함수.

| 김도경

Date. 2023-04-09

목차

01. 함수

1. 함수의 기본

02. 재귀 함수

1. 재귀 함수 란
2. 반복문과 재귀 함수
3. 재귀 함수 중요 원칙
4. 재귀 함수 동작 및 원리

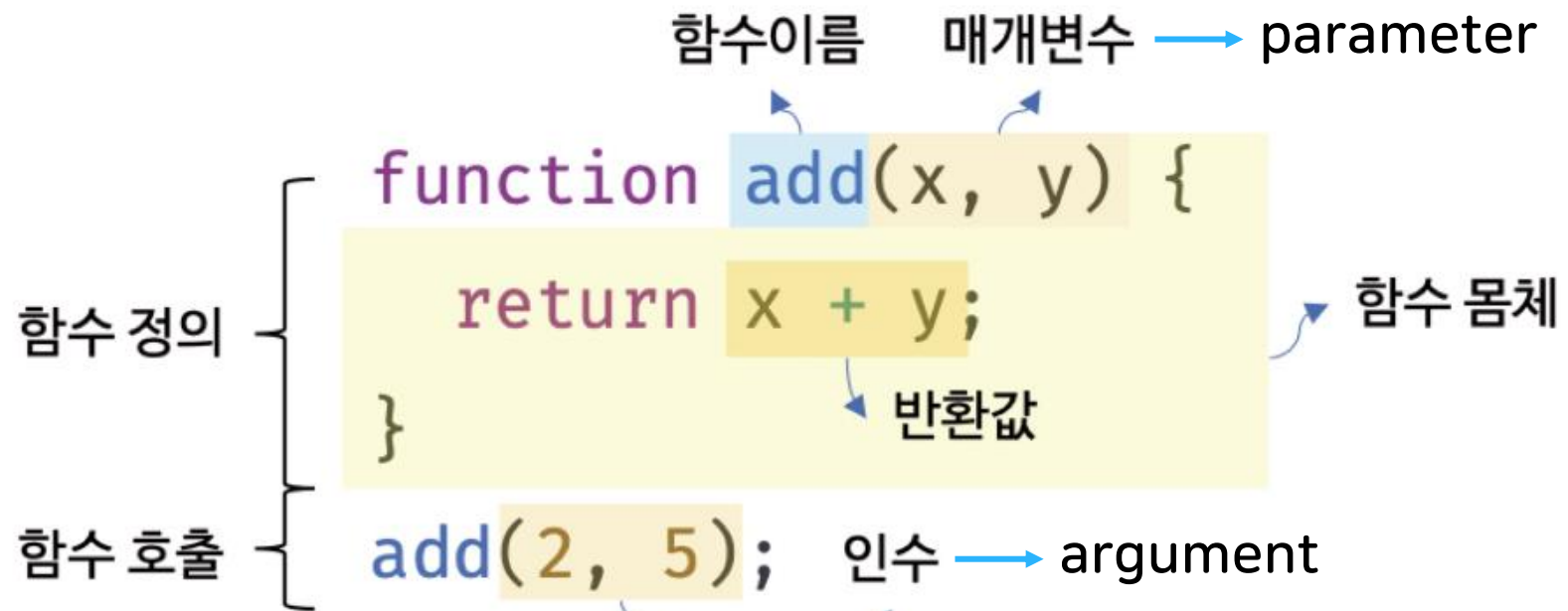
03. 예제로 알아보는 재귀 함수

1. 1번
2. 2번

04. 마무리, 하노이의 탑

01. 함수

1. 함수의 기본



02. 재귀 함수

1. 재귀 함수란

재귀 함수: 재귀 호출을 수행하는 함수.

재귀 호출: 자기 자신을 호출 하는 것.

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```

02. 재귀 함수

1. 재귀 함수란

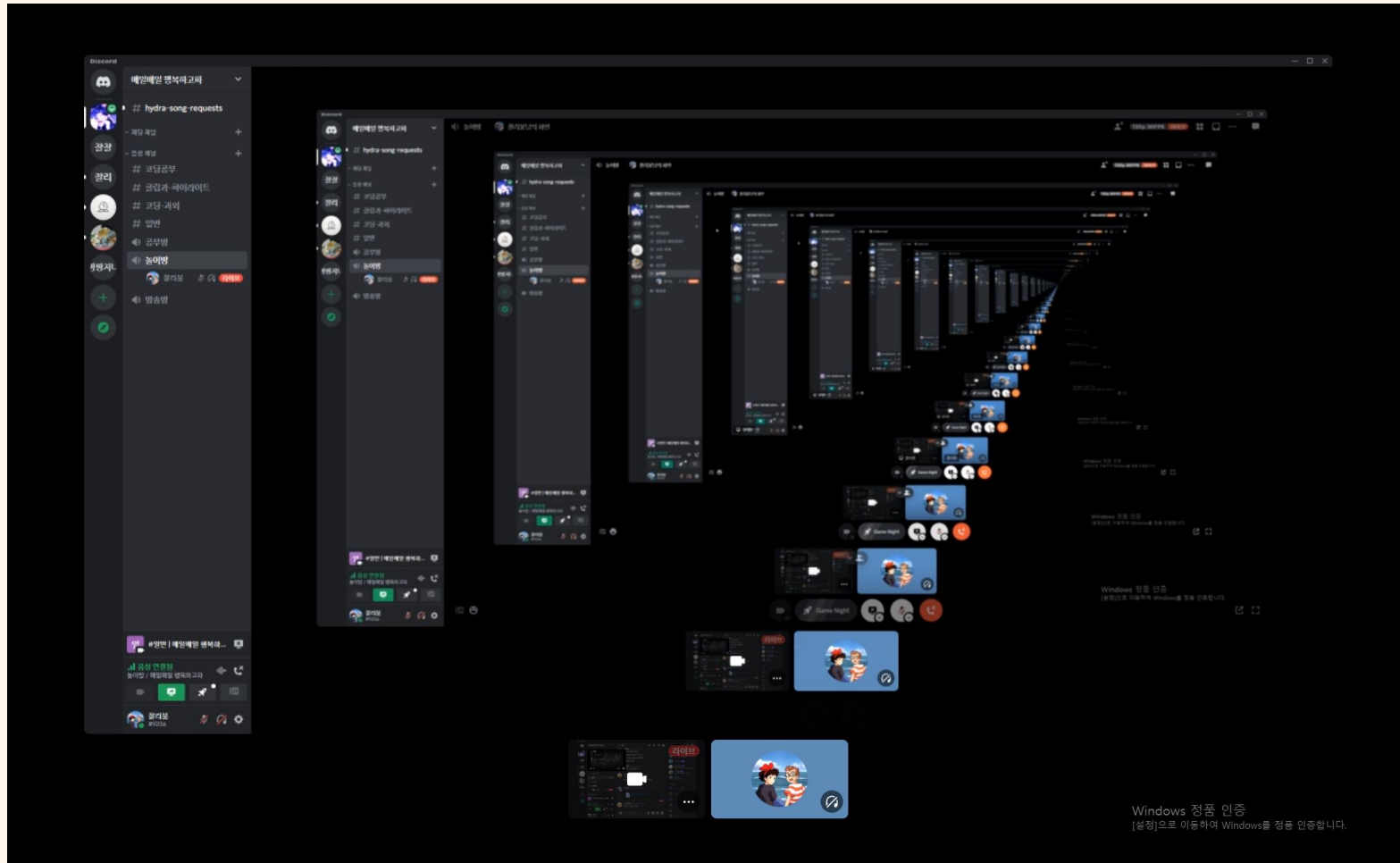
재귀 함수: 재귀 호출을 수행하는 함수.

재귀 호출: 자기 자신을 호출 하는 것.

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```

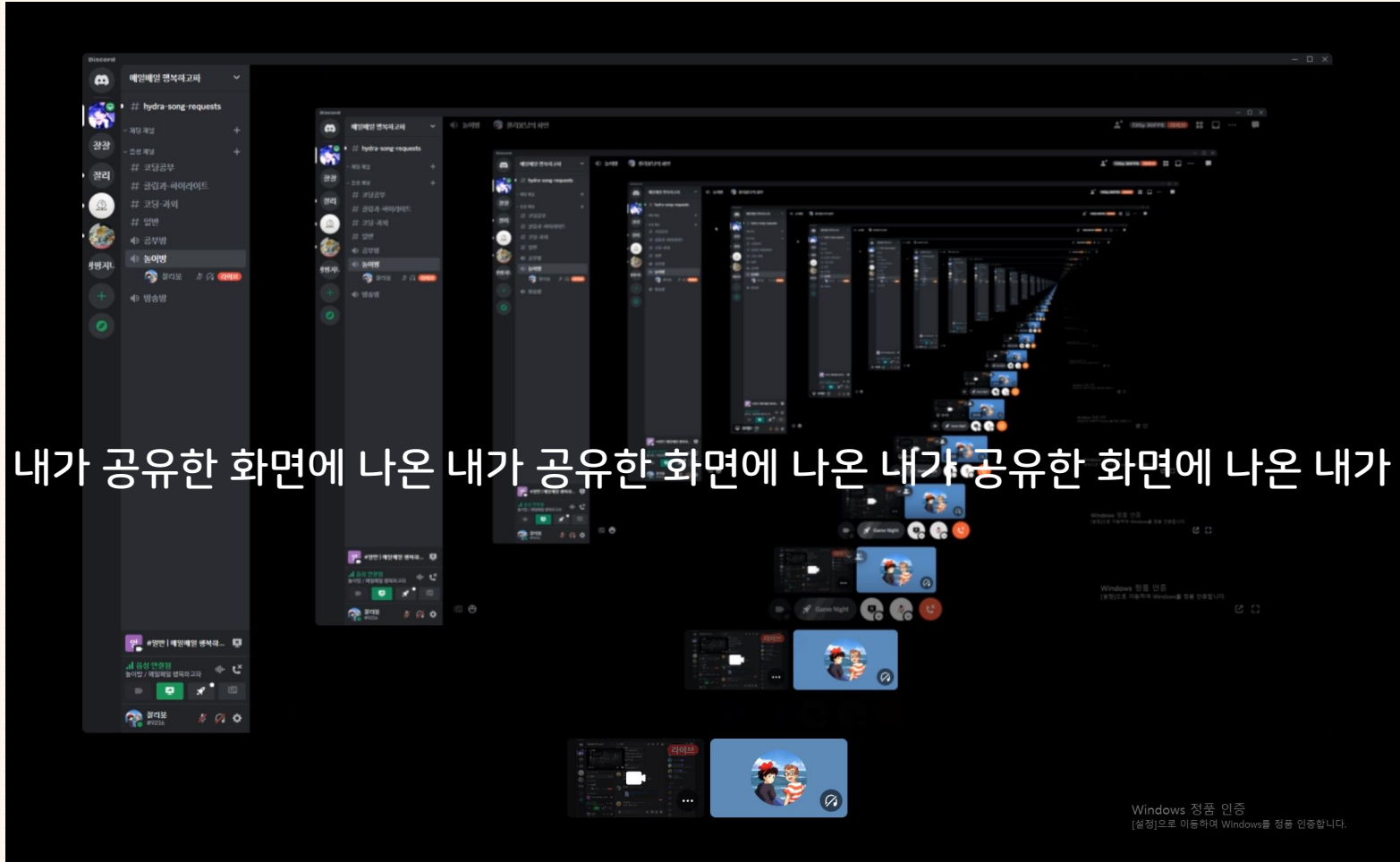
02. 재귀 함수

1. 재귀 함수란



02. 재귀 함수

1. 재귀 함수 란



02. 재귀 함수

2. 반복문과 재귀 함수

숫자 1부터 N까지 출력하는 함수

```
function loop(N) {  
  for (let i = 1; i <= N; i++) {  
    console.log(i);  
  }  
}
```

반복문 활용



```
function recursive(N) {  
  if (N < 1)  
    return;  
  recursive(N - 1);  
  console.log(N);  
}
```

재귀 활용

02. 재귀 함수

3. 재귀 함수 중요 원칙

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```

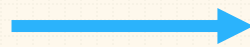
Q

무엇이 문제일까요?

02. 재귀 함수

3. 재귀 함수 중요 원칙

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```

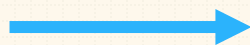


무한 반복

02. 재귀 함수

3. 재귀 함수 중요 원칙

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```



무한 반복

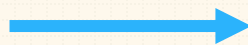
탈출 조건(종료 조건) 필요

탈출 조건: 재귀 호출을 멈출 수 있는 조건

02. 재귀 함수

3. 재귀 함수 중요 원칙

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```



```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

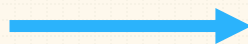
탈출 조건(종료 조건) 필요

탈출 조건: 재귀 호출을 멈출 수 있는 조건

02. 재귀 함수

3. 재귀 함수 중요 원칙

```
function recursive(num) {  
  console.log(num);  
  
  recursive(num + 1);  
}
```



```
function recursive(num) {  
  if (num > 10) {  
    return;  
  }  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

탈출 조건(종료 조건) 필요

탈출 조건: 재귀 호출을 멈출 수 있는 조건

02. 재귀 함수

4. 재귀 함수 동작 및 원리

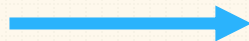
```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
recursive(1);
```

Q

입력 받은 숫자부터 10까지 순서대로 출력하는 함수
($1 \leq \text{num} \leq 10$)

의 출력 결과는 무엇일까요?

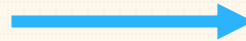


02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
recursive(1);
```



Console

1

2

3

4

5

6

7

8

9

10

02. 재귀 함수

4. 재귀 함수 동작 및 원리

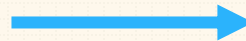
```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```

Q

입력 받은 숫자부터 10까지 순서대로 출력하는 함수
($1 \leq \text{num} \leq 10$)

의 반환값은 무엇일까요?

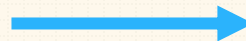


02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```



Console

1

2

3

4

5

6

7

8

9

10

undefined

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```



Console

1

2

3

4

5

6

7

8

9

10

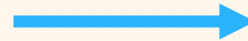
undefined

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return;  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```



Console

1

2

3

4

5

6

7

8

9

10

undefined

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```

Q

입력 받은 숫자부터 10까지 순서대로 출력하는 함수
($1 \leq \text{num} \leq 10$)

의 반환값은 무엇일까요?

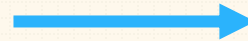


02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```



Console

1

2

3

4

5

6

7

8

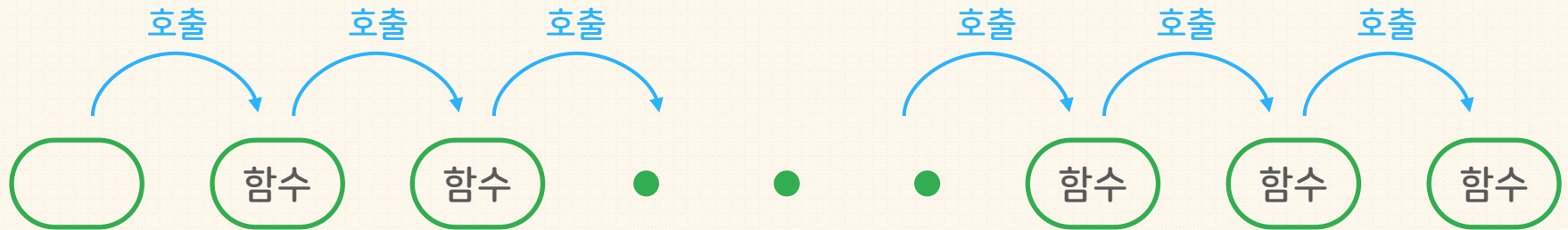
9

10

undefined

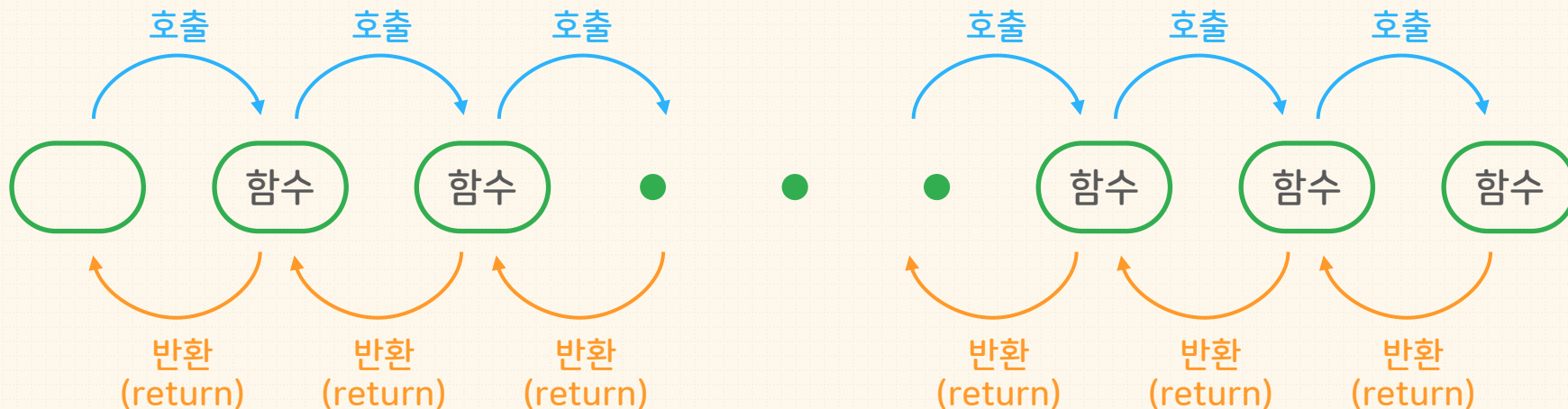
02. 재귀 함수

4. 재귀 함수 동작 및 원리



02. 재귀 함수

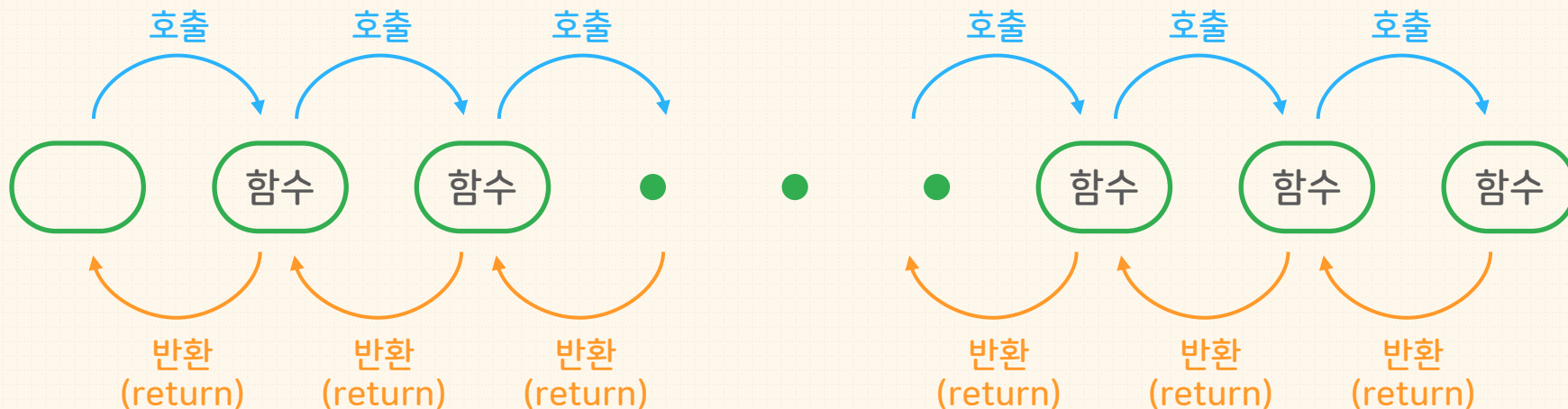
4. 재귀 함수 동작 및 원리



* JavaScript는 return이 명시되어 있지 않으면, 자동으로 **undefined**를 return 한다.

02. 재귀 함수

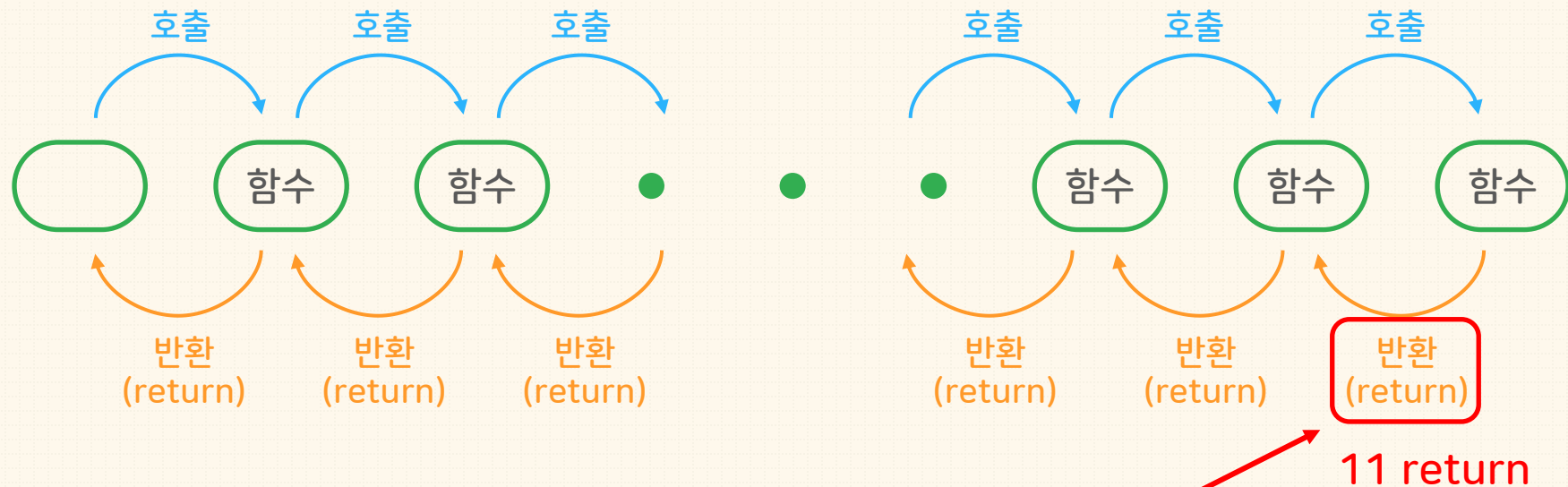
4. 재귀 함수 동작 및 원리



```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```


02. 재귀 함수

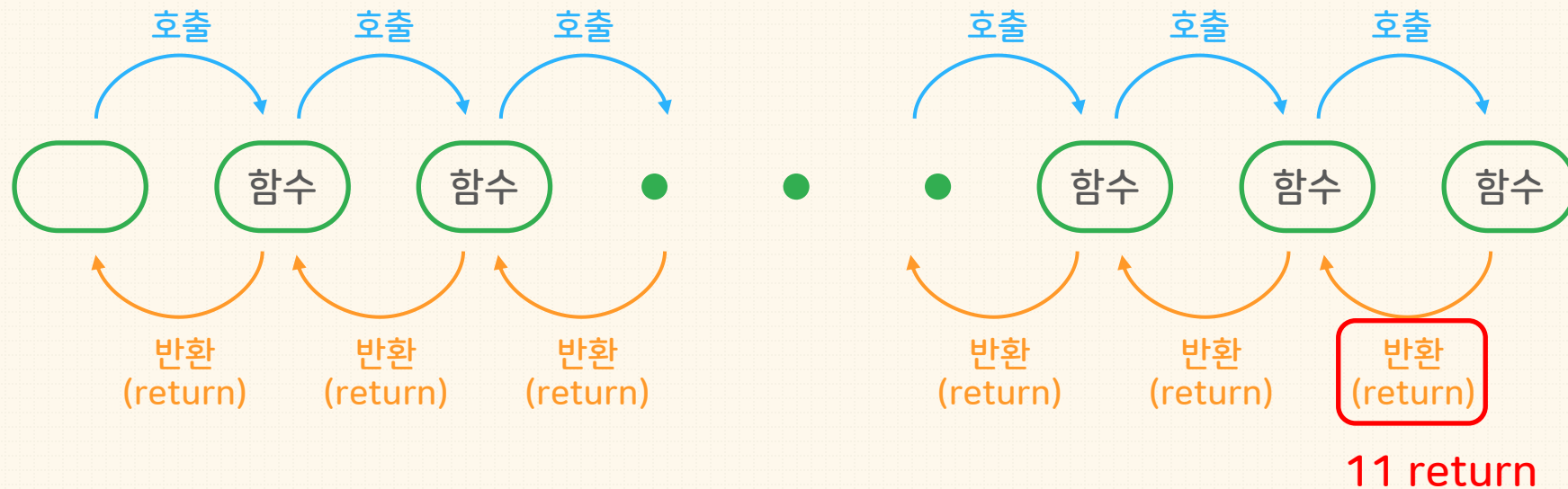
4. 재귀 함수 동작 및 원리



```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

02. 재귀 함수

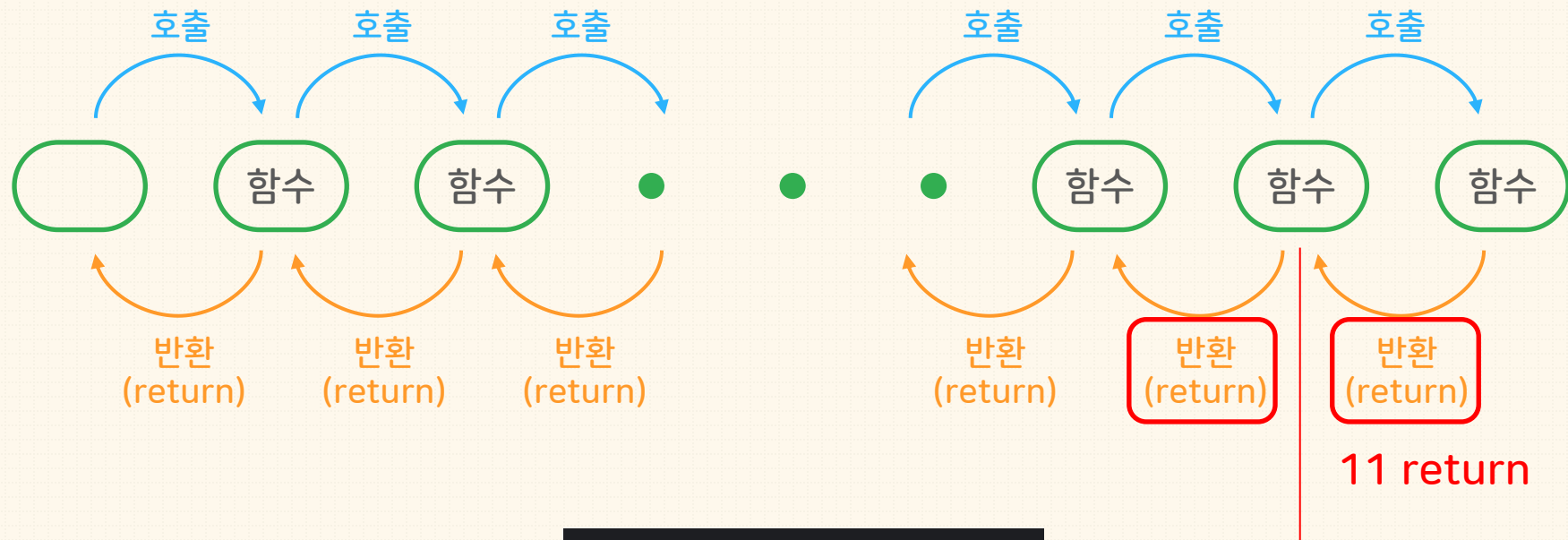
4. 재귀 함수 동작 및 원리



```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리



이후
return 없음

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  recursive(num + 1);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리

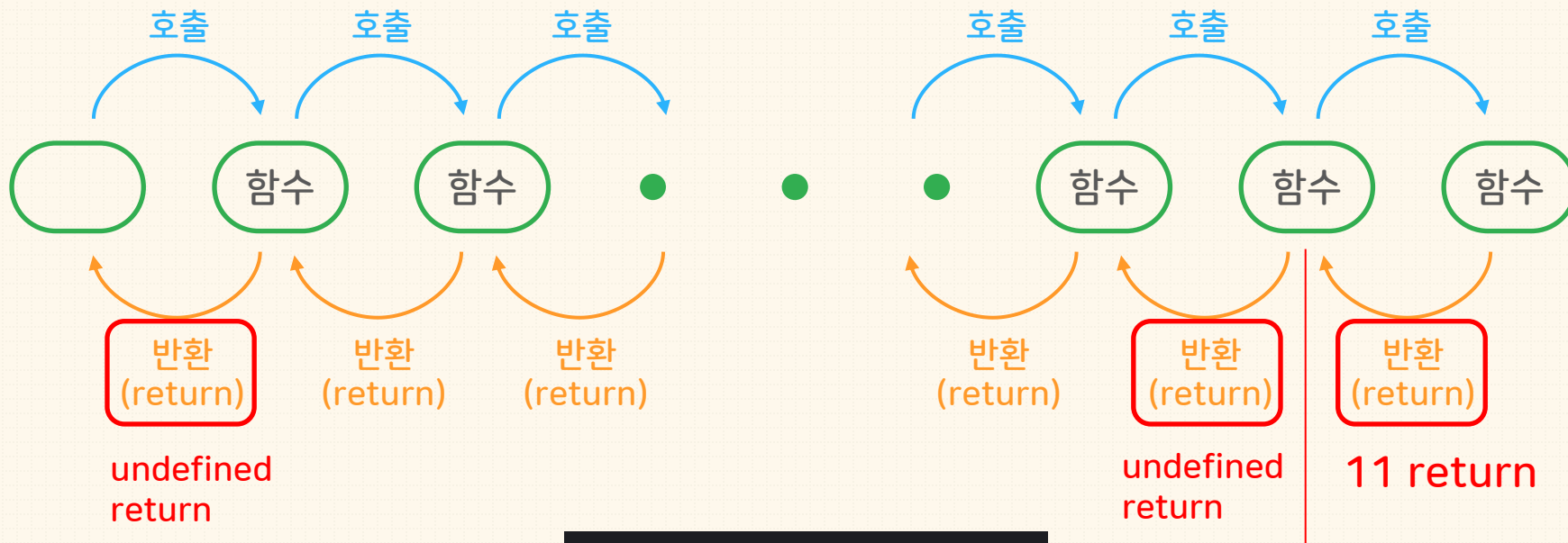


이후
return 없음

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  recursive(num + 1);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리



이후
return 없음

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  recursive(num + 1);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```

```
console.log(recursive(1));
```

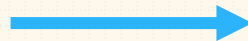
Q

최종 결과로 11이 반환되게 하려면
어떻게 해야할까요?

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
}
```



```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
  return (11);  
}
```

return (11); 추가

```
console.log(recursive(1));
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  recursive(num + 1);  
  return (11);  
}
```

=>

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  return (recursive(num + 1));  
}
```

이렇게 작성할 수도 있습니다.

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  return (recursive(num + 1));  
}
```

```
console.log(recursive(1));
```

어떻게 진행되는지 이해가 되시나요?

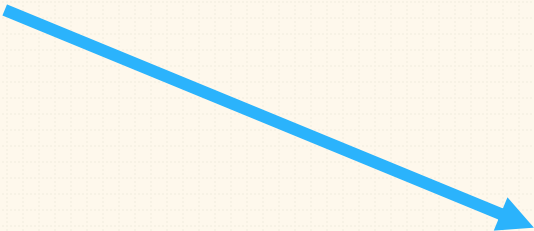
02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  console.log(num);  
  return (recursive(num + 1));  
}
```

```
console.log(recursive(1));
```

어떻게 진행되는지 이해가 되시나요?



```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

반환

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

반환

이어서
진행

02. 재귀 함수

4. 재귀 함수 동작 및 원리

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

호출

```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

반환

호출

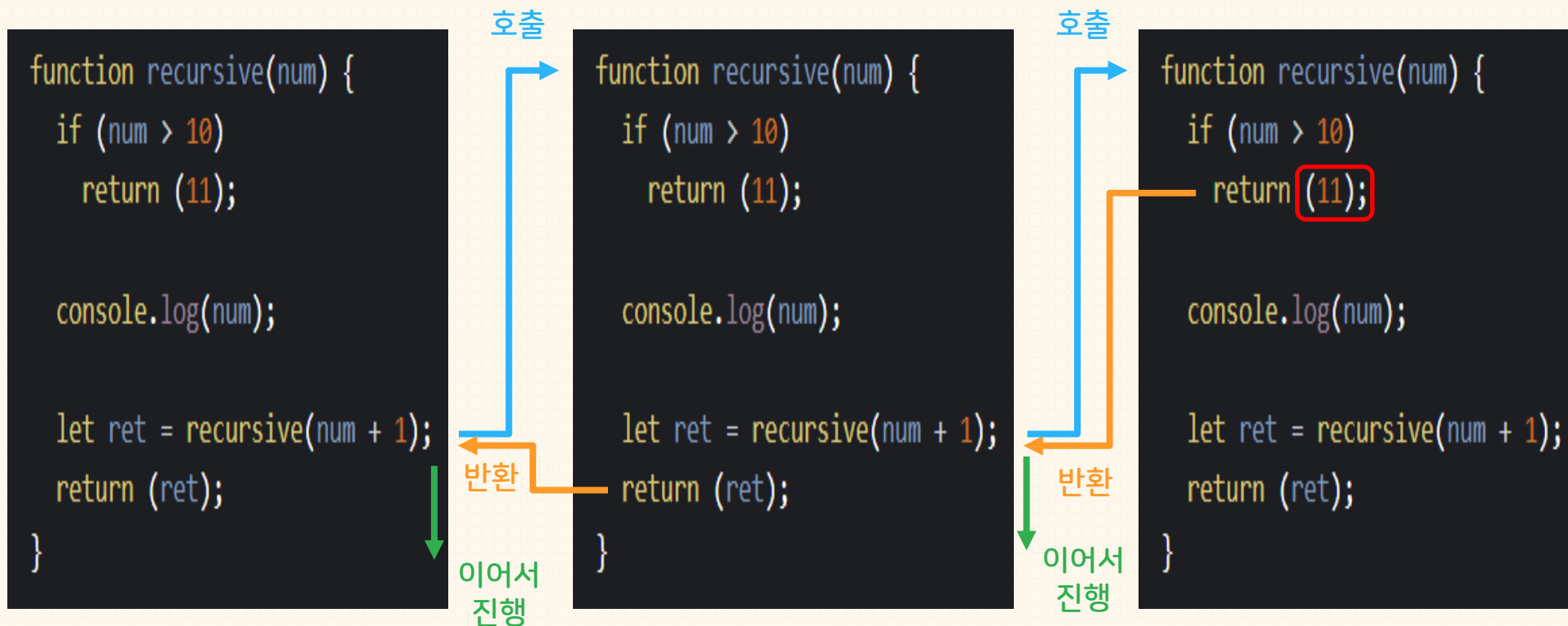
```
function recursive(num) {  
  if (num > 10)  
    return (11);  
  
  console.log(num);  
  
  let ret = recursive(num + 1);  
  return (ret);  
}
```

반환

이어서
진행

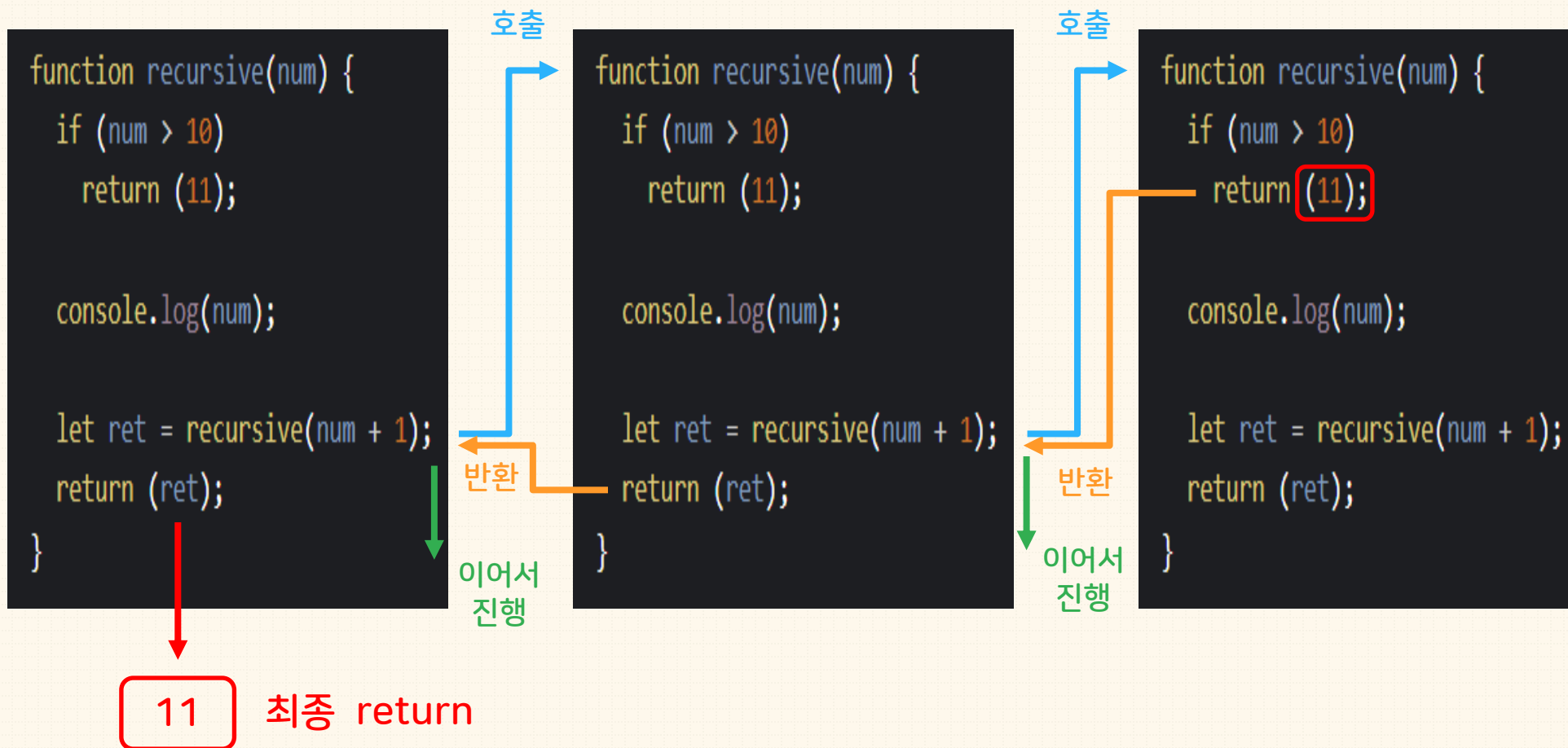
02. 재귀 함수

4. 재귀 함수 동작 및 원리



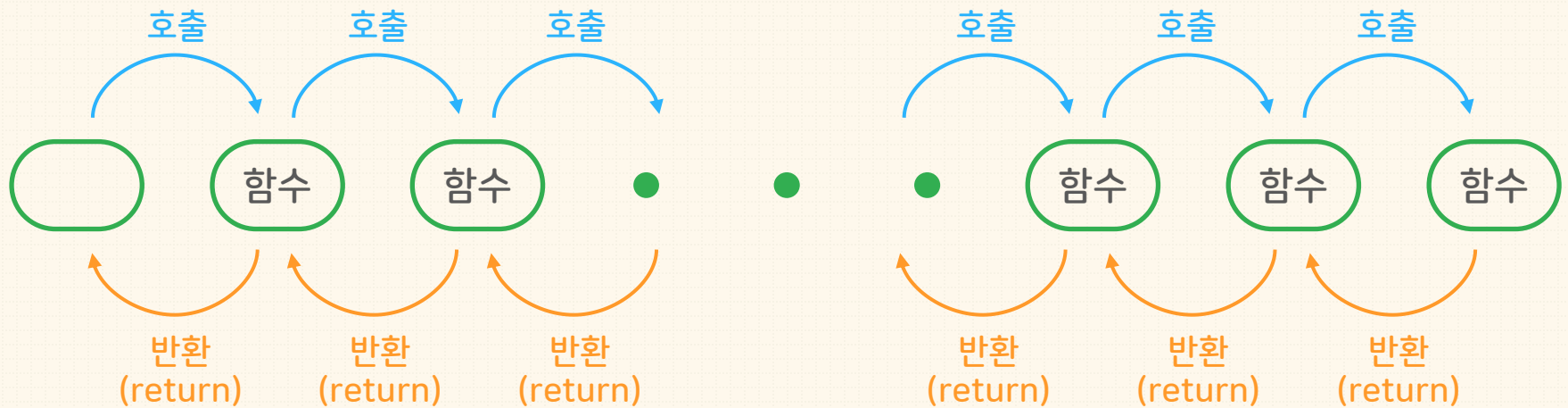
02. 재귀 함수

4. 재귀 함수 동작 및 원리



02. 재귀 함수

4. 재귀 함수 동작 및 원리



03. 예제로 알아보는 재귀 함수

1. 1번

```
function recursive(N) {  
  if (N < 1)  
    return;  
  recursive(N - 1);  
  console.log(N);  
}
```

```
recursive(10);
```

Q

왼쪽의 recursive 함수는 어떤 함수일까요?

또 출력 결과는 어떻게 될까요?

03. 예제로 알아보는 재귀 함수

1. 1부터 N까지 출력하는 함수

```
function recursive(N) {  
  if (N < 1)  
    return;  
  recursive(N - 1);  
  console.log(N);  
}
```

```
recursive(10);
```

1부터 N까지 출력하는 함수

Console

1

2

3

4

5

6

7

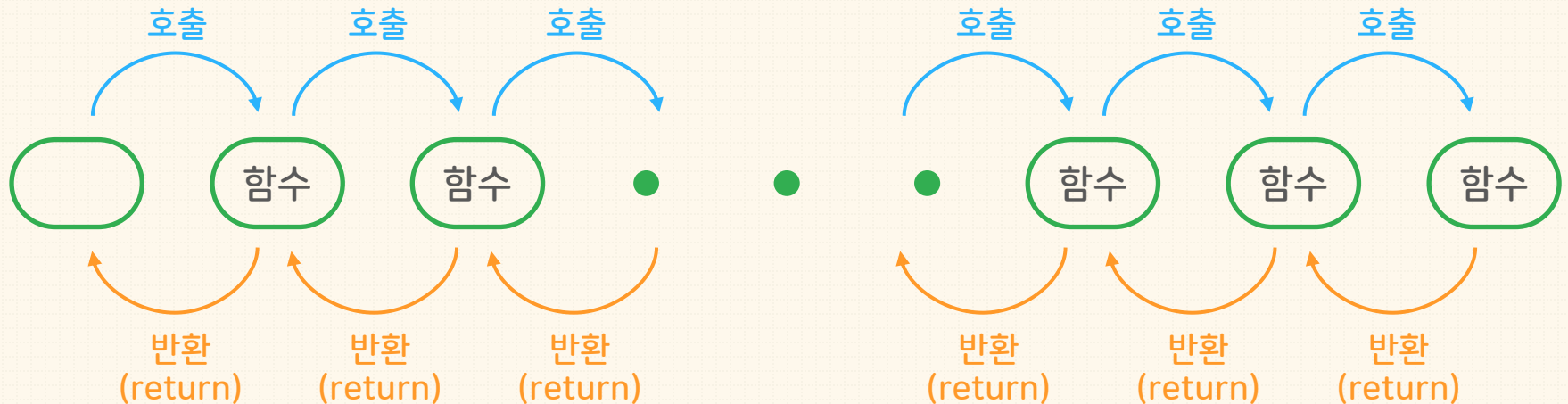
8

9

10

03. 예제로 알아보는 재귀 함수

1. 1부터 N까지 출력하는 함수

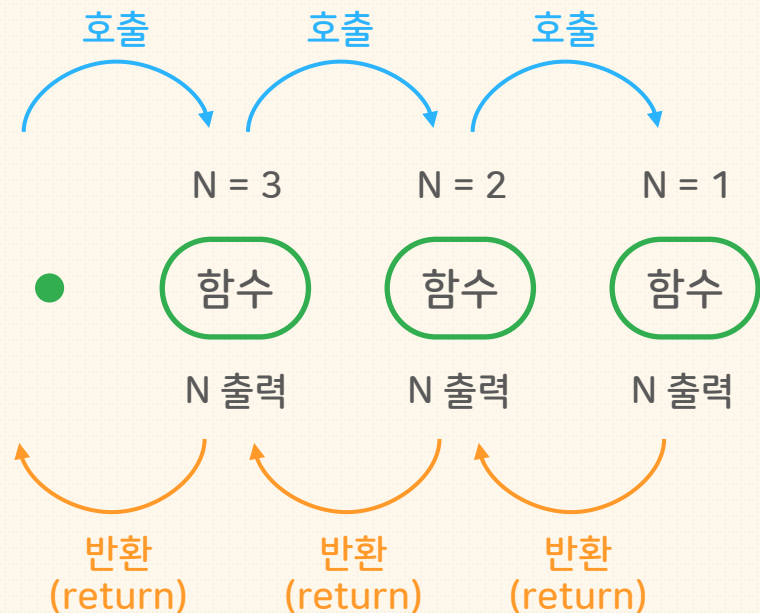
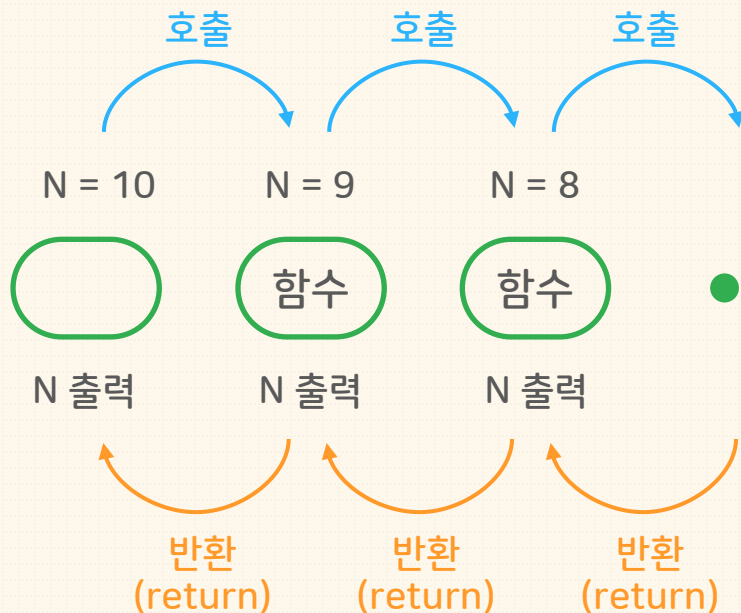


03. 예제로 알아보는 재귀 함수

1. 1부터 N까지 출력하는 함수

```
function recursive(N) {  
  if (N < 1)  
    return;  
  recursive(N - 1);  
  console.log(N);  
}
```

```
recursive(10);
```



03. 예제로 알아보는 재귀 함수

2. 2번

질문 타임

```
function recursive(N) {  
    if (N <= 1)  
        return N;  
  
    return (N * recursive(N - 1));  
}
```

Q recursive 함수는 어떤 함수일까요?

03. 예제로 알아보는 재귀 함수

2. 팩토리얼

```
function recursive(N) {  
    if (N <= 1)  
        return N;  
  
    return (N * recursive(N - 1));  
}
```

N 팩토리얼의 값을 반환하는 함수

03. 예제로 알아보는 재귀 함수

2. 팩토리얼

```
function recursive(N) {  
  if (N <= 1)  
    return N;  
  
  return (N * recursive(N - 1));  
}
```

$\text{recursive}(10) = 10 * \text{recursive}(9)$

$\text{recursive}(9) = 9 * \text{recursive}(8)$

$\text{recursive}(8) = 8 * \text{recursive}(7)$

...

04. 마무리

프로그래머스 재귀 연습 문제

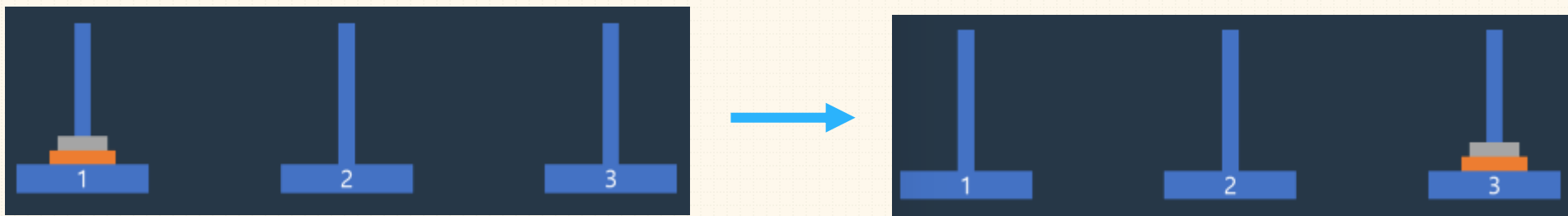
피보나치 수 <https://school.programmers.co.kr/learn/courses/30/lessons/120848>

팩토리얼 <https://school.programmers.co.kr/learn/courses/30/lessons/12945>

04. 마무리

하노이의 탑

하노이의 탑 <https://school.programmers.co.kr/learn/courses/30/lessons/12946>



1번 기둥에 있는 원판의 개수 n 이 매개변수로 주어질 때,
두 가지 조건을 만족시키면서
 n 개의 원판을 3번 원판으로 최소로 옮기는 방법을 return하는 함수를 작성하세요.

- * 한 번에 하나의 원판만 옮길 수 있습니다.
- * 큰 원판이 작은 원판 위에 있어서는 안됩니다.

04. 마무리

재귀 함수 사용 이유 및 주의점

재귀 함수는 반복되는 처리를 **반복문 없이** 구현할 수 있다는 장점이 있지만 **무한 반복**에 빠질 위험이 있고, 이로 인해 **스택 오버플로우 에러**를 발생시킬 수 있으므로 **주의**해서 사용해야 한다.

따라서 재귀 함수는 **반복문을 사용하는 것보다 재귀 함수를 사용하는 편이 더 직관적으로 이해하기 쉬울 때만** 한정적으로 사용하는 것이 바람직하다.

Q & A

감사합니다.

김도경