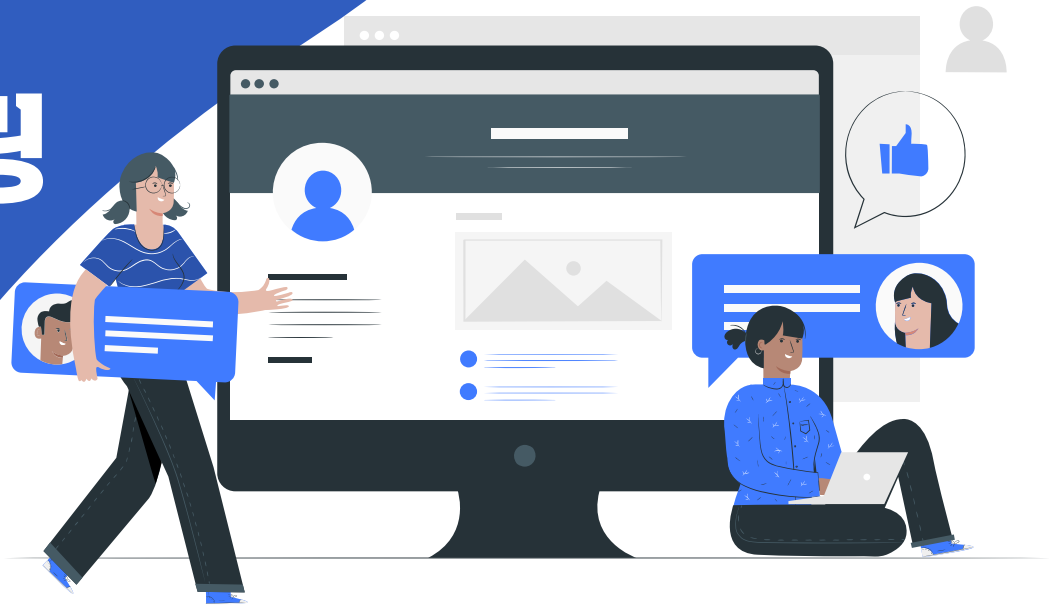
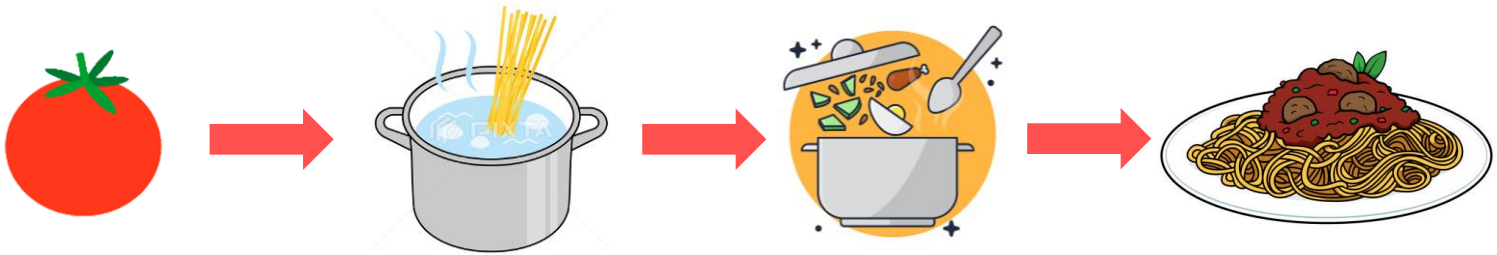
6주차 튜터링



06 함수1

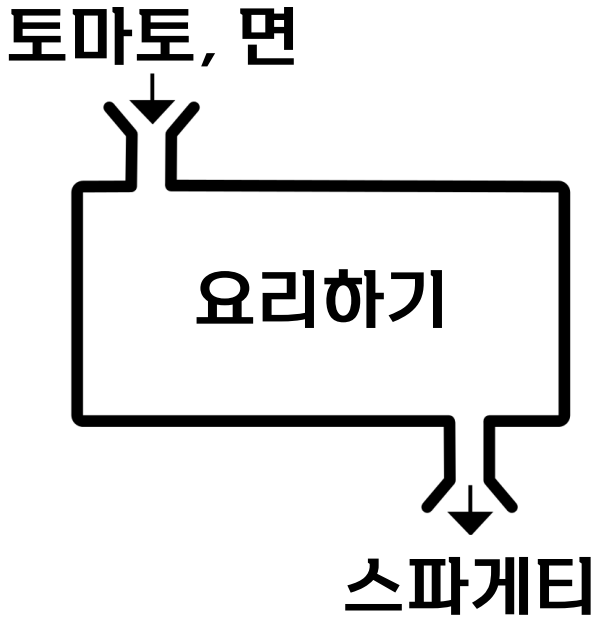
목차

함수를 쓰는 이유



요리를 하는 경우 토마토 손질부터 면 삶기 까지
스파게티를 만드는 과정은 “요리”라는 작업으로 묶을 수 있다

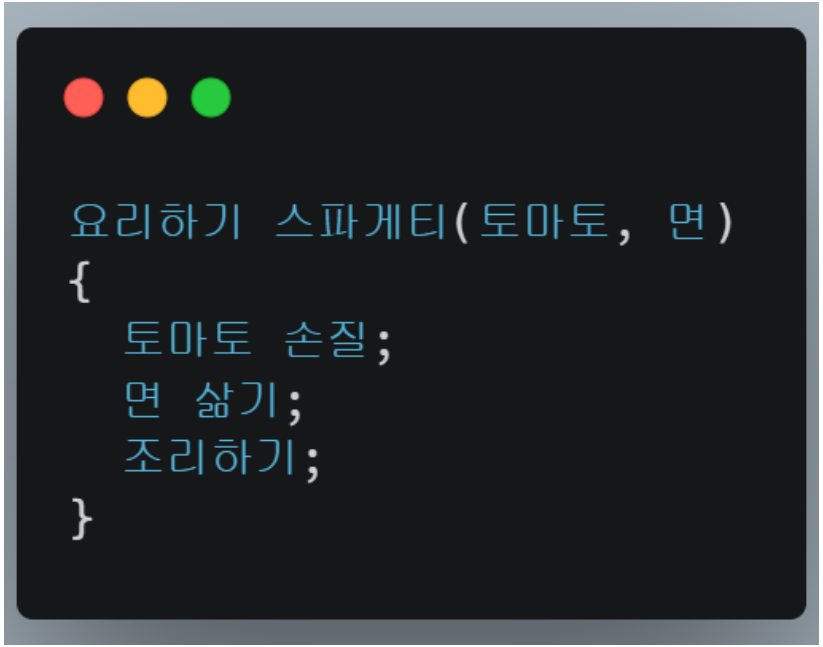
함수의 정의



- 함수란?
  프로그램에서 특정한 작업을
  수행하도록 따로 정의해 놓은
  독립적인 프로그램의 단위이다.

- 좀 더 쉽게 풀어서 설명해보면
  “정해진 작업을 하기 위해서
  명령문들을 하나의 그룹으로
  묶은 것”이라고 생각하면 쉽다.

함수의 정의



```
요리하기 스파게티(토마토, 면)
{
  토마토 손질;
  면 삶기;
  조리하기;
}
```

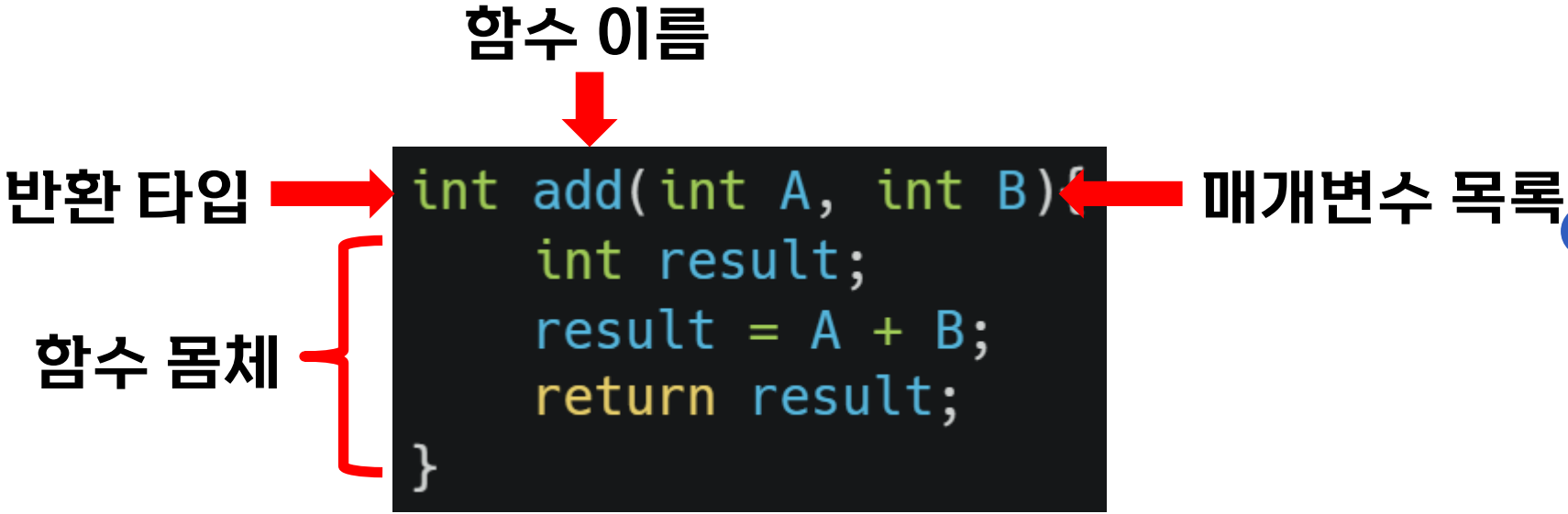
- 굳이 스파게티 요리 하는 과정을 코드로 표현 할 때는 다음과 같은 코드가 완성된다.

함수의 정의

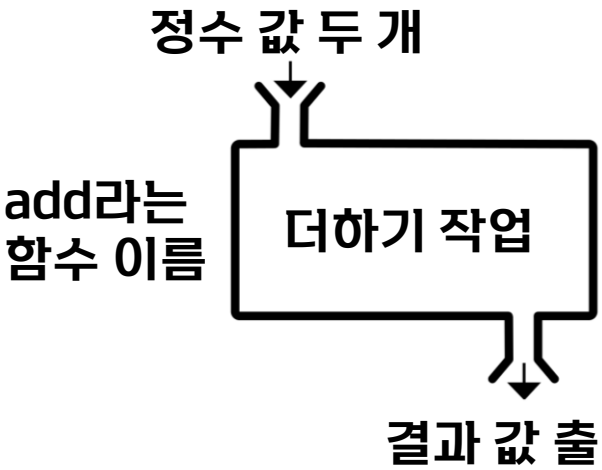
```c
#include <stdio.h>

int add(int A, int B) {
    int result;
    result = A + B;
    return result;
}

int main() {
    int a;
    int b;
    int sum;

    a = 3;  b = 5;

    sum = add(a, b);
    sum = add(4, 5);
}
```

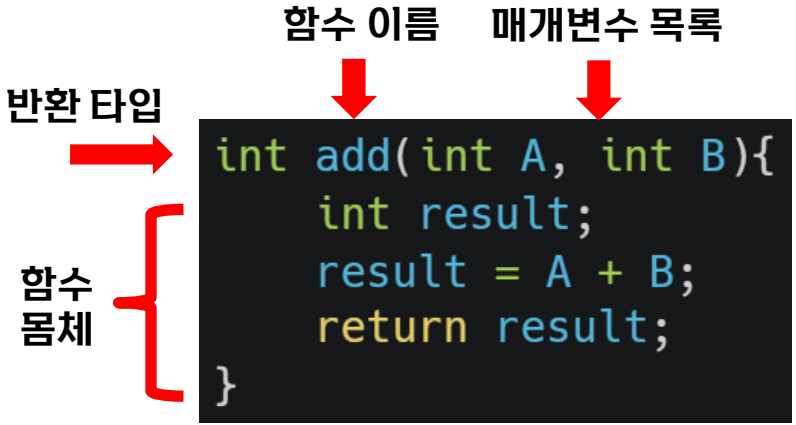
- 이 프로그램 한 번 따라서 코딩해보세요~

함수의 정의


함수 이름
반환 타입
매개변수 목록
함수 몸체
int add(int A, int B){
int result;
result = A + B;
return result;
}

함수의 정의





- 반환 타입(return type) : 함수가 모든 작업을 마치고 반환하는 데이터의 타입을 명시합니다.
- 함수 이름 : 함수를 호출하기 위한 이름을 명시합니다.
- 매개변수 목록(parameters) : 함수 호출 시에 전달되는 인수의 값을 저장할 변수들을 명시합니다.
- 함수 몸체 : 함수의 고유 기능을 수행하는 명령문의 집합입니다.

반환 타입

```
int add()
  // add 함수의 반환값은 int형

float hap()
  // hap 함수의 반환값은 float형

void tree()
  // tree 함수의 반환값은 없음.
```

- 함수에서도 변수를 선언 할 때와 마찬가지로 함수의 데이터 타입을 설정해야 한다.
- 변수에서의 데이터 타입은 성격과 크기를 지정하지만,
- 함수에서는 함수의 반환값에 대한 데이터 타입이다.

반환 타입

```
int add(int A, int B){
    int result;
    result = A + B;
    return result;
}
```

- 함수의 데이터 타입과 반환값은 반드시 일치해야 한다.

매개변수

03 매개변수로 나타낸 함수의 미분법

2 여러 가지 미분법

매개변수로 나타낸 함수

두 변수 x, y 사이의 관계가 변수 t를 매개로 하여

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases}$$

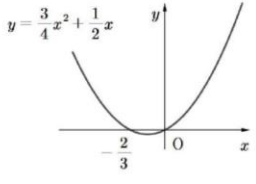
의 꼴로 주어진 함수를 매개변수로 나타낸 함수라고 하고, 변수 t를 매개변수라고 한다.

다음 매개변수로 나타낸 함수에서 매개변수 t를 소거하여 x, y의 관계식을 구하고 그 그래프를 그리시오.

(1) $\begin{cases} x = 2t \\ y = 3t^2 + t \end{cases}$ (2) $\begin{cases} x = \cos t \\ y = \sin t \end{cases}$

(1) $x = 2t$에서 $t = \frac{x}{2}$이므로

$$y = 3t^2 + t = 3\left(\frac{x}{2}\right)^2 + \frac{x}{2} = \frac{3}{4}x^2 + \frac{1}{2}x$$



변수 | 變數 | variable

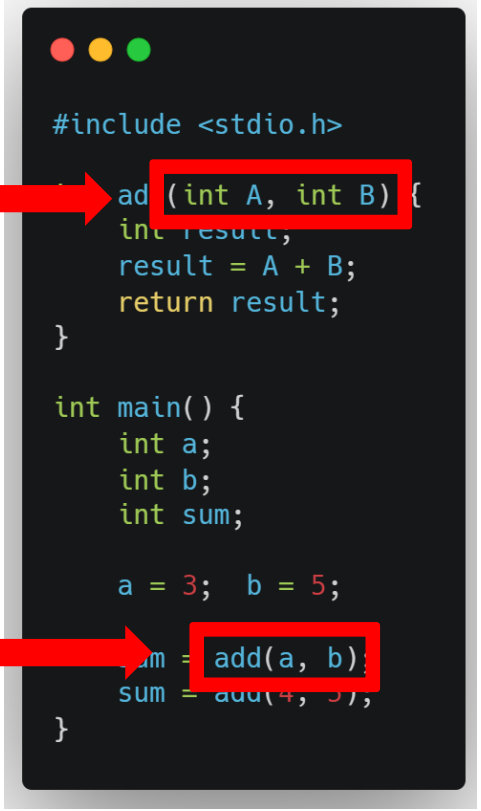
변은 변함을 뜻한다. 변수는 '변하는 수'이다. 변수에는 다양한 값을 대입할 수 있다. 함수에서 매개 변수가 대표적인 예이다.
변수는 값이 고정되지 않은 수이라는 뜻이고, 모르는 수는 아니다.

→ 매개변수는 그냥
함수 f(x, y)에서의 x,y라고
생각해도 상관 없어요~

매개변수

형식 매개 변수

실 매개 변수



- 실 매개 변수(Actual Parameters)
  : 함수를 호출하는 쪽의 매개 변수
- 형식 매개 변수(Type Parameters)
  : 함수를 호출당한 쪽의 매개 변수

함수의 사용 방법

- 방법 1 : 함수 선언 → 함수 호출 → 함수 정의
- 방법 2 : 함수 정의 → 함수 호출

함수의 사용 방법

```c
#include <stdio.h>

int add(int A, int B);

int main(){
    int a;
    int b;
    int sum;

    a = 3; b = 5;

    sum = add(a, b);
    sum = add(4, 5);
}

int add(int A, int B){
    int result;
    result = A + B;
    return result;
}
```

방법 1

함수 선언 →
함수 호출 →
함수 정의

```c
#include <stdio.h>

int add(int A, int B) {
    int result;
    result = A + B;
    return result;
}

int main() {
    int a;
    int b;
    int sum;

    a = 3;  b = 5;

    sum = add(a, b);
    sum = add(4, 5);
}
```

방법 2

함수 정의 →
함수 호출

함수의 사용 방법

- 방법 1 : 함수 선언 → 함수 호출 → 함수 정의

```
int add(int num1, int num2);     // 함수의 선언

main() 함수
...
sum = add(a,b);                  // 함수의 호출

int add(int num1, int num2)      // 함수의 정의
{
   ...
   sum = num1 + num2;
   ...
}
```

함수의 사용 방법

- 방법 2 : 함수 정의 → 함수 호출

```
int add(int num1, int num2);     // 함수의 정의
{
   ...
   sum = num1 + num2;
}
```

```
main() 함수
...
sum = add(a,b);                  // 함수의 호출
```

매개변수와 반환값의 리턴

프로그램 타입	매개변수	반환값
A 타입	X	X
B 타입	O	X
C 타입	X	O
D 타입	O	O

매개변수와 반환값의 리턴 ~A타입~

```c
#include <stdio.h>
//사용자 정의 함수 선언
void JustNumber();

//main ( ) 영역
int main()
{
    JustNumber(); //사용자 정의 함수 호출
    return 0;
}

//사용자 정의 함수
void JustNumber()
{
    int i;
    printf("정수를 입력하세요. \n");
    scanf("%d", &i);

    printf("입력한 숫자는 %d 입니다. \n", i);
}
```

- A 타입 :
  매개변수와 반환값이 모두 없는 경우
- 매개변수가 없다는 말은 main() 함수 영역으로 부터 입력받을 데이터가 없다는 의미입니다.
- 반환문이 없다는 뜻은 사용자 정의 함수에서 main() 함수로 돌려줄 값이 없다는 의미입니다.

매개변수와 반환값의 리턴 ~B타입~

```c
#include <stdio.h>
//사용자 정의 함수 선언
void JustNumber(int i);

//main () 영역
int main()
{
    int result;
    printf("정수를 입력하세요. \n");
    scanf("%d", &result);

    JustNumber(result);
    //변수 result를 사용자 정의 함수에 전달해서 다시 반환

    return 0;
}

//사용자 정의 함수
void JustNumber(int i) //전달받은 변수 result를 매개변수 i에 대입
{
    printf("입력한 숫자는 %d 입니다. \n", i);
  //main()으로 반환하지 않고 바로 출력
}
```

- B 타입 :
  매개변수는 있고, 반환값은 없는 경우
- 매개변수가 있는 경우 main() 함수로 부터 변수값을 전달받을 수 있습니다.
- 반환문이 없는 경우 void 자료형을 쓰고 return을 삽입하지 않습니다.

매개변수와 반환값의 리턴 ~C타입~

```c
#include <stdio.h>
//사용자 정의 함수 선언
int JustNumber();

//main () 영역
int main()
{
    int result;
    result = JustNumber();
  //사용자 정의 함수에서 반환받은 값을 result에 저장
    printf("입력한 숫자는 %d 입니다.", result);
    return 0;
}

//사용자 정의 함수
int JustNumber()
{
    int i;
    printf("정수를 입력하세요. \n");
    scanf("%d", &i);

    return i;
  //변수 i에 삽입된 변수값을 main () 함수로 반환합니다.
}
```

- C 타입 :
  매개변수는 없고, 반환값만 있는 경우

- 함수의 연산결과에 정수와 같은 숫자를 main() 함수에 돌려주고 싶은 경우 자료형은 int등으로 변경하고 return 문에 반환값을 작성해야 합니다.

- return 문에는 원하는 연산을 작성할 수 있습니다. void JustNumber(void) 형태도 가능합니다.

매개변수와 반환값의 리턴 ~D타입~

```c
#include <stdio.h>
//사용자 정의 함수 선언
int JustNumber(int i);

//main () 영역
int main()
{
    int result;
    printf("정수를 입력하세요. \n");
    scanf("%d", &result);

    JustNumber(result);
    //변수 result를 사용자 정의 함수에 전달해서 다시 반환

    printf("입력한 숫자는 %d 입니다.", result);
    //반환된 result의 변수값을 출력

    return 0;
}

//사용자 정의 함수
int JustNumber(int i)
//전달받은 변수 result를 매개변수 i에 대입
{
    return i;
    //매개변수 i의 값을 main() 함수에 반환
}
```

- D 타입 :
  매개변수와 반환값이 모두 있는 경우
- 매개변수가 있으면 사용자 정의 함수로 데이터가 전달됩니다.
- 또한 반환문이 있으면 함수에서 생산된 데이터가 return 문을 통해서 main() 함수로 다시 돌아옵니다.

END