

Fucntion 요약

Function, 함수

- 수행하고자 하는 일련의 동작들에 붙여진 이름
- 프로그램을 이해하고 수정하는데 도움이 됨
 - Definition, 정의
 - Calling, 호출
 - Arguments, 인자
 - return, 리턴
 - recursion, 재귀

Function, 함수: Definition, 정의

- 호출하려는 함수보다 먼저 함수의 정의가 작성되어야함
 - 함수 선언:
 - 함수의 내용은 없이 앞으로 사용할 함수의 이름과 인자값을 프로그램에 등록함.
 - 작성 스타일에 따라 항상 필요하진 않음.
 - 함수 정의:
 - 함수의 실제 내용이 기록됨

Function, 함수: Declaration, 선언

```
#include <stdio.h>
```

```
int sum(int a, int b) ; 함수 선언
```

```
int main() {
```

```
    ...
```

```
}
```

```
int sum(int a, int b) 함수 정의
```

```
{
```

```
    함수 내용
```

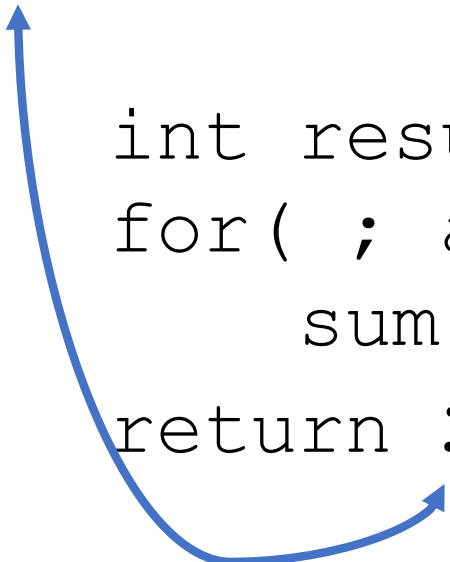
```
}
```

Function, 함수: Definition, 정의

리턴 데이터타입	함수 이름	함수 내에서 사용될 인자들
<i>return-type</i>	<i>function-name</i>	(<i>parameters</i>)
{		
<i>declarations</i>	함수 내에서 사용될 변수 선언	
<i>statements</i>	함수 내에서 실행할 문장들	
}		

```
int sum(int a, int b)
{
    int result=0;
    for( ; a <= b; a++)
        sum += a;
    return result;
}
```

타입 일치



Function, 함수: Calling, 호출

- 코드 내에 함수 이름을 작성하여 호출

```
int main() {
```

```
...
```

```
answer = sum(val1, val2);
```

```
...
```

```
}
```

기존 코드의 흐름은 위에서 아래로 실행

함수 호출 시 함수 정의 위치로 실행의 흐름이 이동함

```
int sum(int a, int b)
```

```
{
```

```
...
```

```
return result;
```

```
}
```

Function, 함수: Arguments, 인자

- 함수 밖에서 정의된 값으로 함수 내에서 활용할 변수 이름

```
int main() {
```

```
...
```

```
answer = sum(val1, val2);
```

```
...
```

```
}
```

메인 코드에서 선언된 val1과 val2를
sum이라는 함수에 활용할 수 있도록 인자로 넣었음

함수에서 활용할 변수 이름은 메인코드의 변수 이름과 달라도 됨

```
int sum(int a, int b)
```

```
{
```

```
...
```

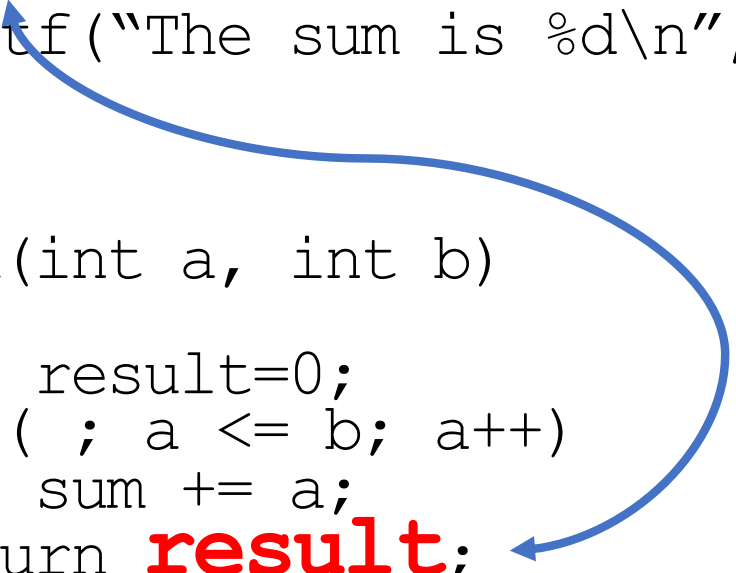
```
return a + b;
```

```
}
```

Function, 함수: 종합 예제

```
#include <stdio.h>
int sum(int a, int b);
int main()
{
    int val1, val2, answer;
    printf("Input two numbers in increasing order: ");
    scanf("%d %d", &val1, &val2);
    answer = sum(val1, val2);호출
    printf("The sum is %d\n", answer);
}

int sum(int a, int b)
{
    int result=0;
    for( ; a <= b; a++)
        sum += a;
    return result;
}
```



결과 값 반환