

**function return**

# function - definition recap

“.. is a sequence of program instructions that perform a specific task, packaged as a unit.”

<https://en.wikipedia.org/wiki/Subroutine>

# function parameters

“used in a subroutine to refer to one of the pieces of data provided as input to the subroutine”

[https://en.wikipedia.org/wiki/Parameter\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Parameter_(computer_programming))

# function example

Imagine we have a function that can calculate the age of a person. Let's call it **howOld**

# function example

When calling that function the age is calculated  
... but how do we store the result?

# function example

If I were to ask you how old you are, you give me back the resulting answer.

Imagine I ask you to write the answer down on piece of paper....

# function return value

That would be returning the answer (value)

# function return

add(2,3)

store the result (returned value)



# function return

```
int result = add(2,3)
```

the return value is now stored in result

# function return

```
int result = add(2,3)
```

the return value is now stored in result, so  
result has the value 5

# function return

```
int result1 = add(2,3)
```

```
int result2 = add(7,9)
```

.. result1 has the value 5 and result2 has the value 16

# function return syntax

<return type> name (<parameters>)

Example:

```
int add(int a, int b)
```

```
int priceAfterVAT(int price, int vat)
```

# Void

If we want to return nothing?

We have a type for that, void

# void

“..is the type for the result of a function that returns normally, but does not provide a result value to its caller”

[https://en.wikipedia.org/wiki/Void\\_type](https://en.wikipedia.org/wiki/Void_type)

**function return - pt II**

# return type - a program

```
bla();  
int result = add(23, 45);  
bleh();
```



# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int add(int a, int b) {  
    int sum = a+b;  
    return sum;  
}
```

# return type - a program explained

```
bla();  
int result = add(23, 45);  
bleh();
```

# return type - a program explained

```
bla();  
int result = add(23, 45);  
bleh();
```

# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int a  
int b
```

# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int a = 23  
int b = 45
```

# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int a = 23  
int b = 45
```

# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int a = 23
```

```
int b = 45
```

```
a + b => 23 + 45 => 68
```



# return type

```
int add(int a, int b) {  
    return a+b;  
}
```

```
int a = 23
```

```
int b = 45
```

```
return 68
```

# return type - a program explained

```
bla();  
int result = add(23, 45); ← 68  
bleh();
```

# return type

```
int addStrange(int a, int b) {  
    if (a==0) {  
        return 0;  
    } else {  
        return a+b;  
    }  
}
```

# return type

```
int addStrange(int a, int b) {  
    if (a==0) {  
        return 0;  
    } else {  
        return a+b;  
    }  
}
```

# return type

```
int addStrange(int a, int b) {  
    if (a==0) {  
        return 0;  
    }  
    return a+b;  
}
```