



Crash Course: Functions



ST. MARY'S HIGH SCHOOL



What is a function?

Functions are like commands that we use to get the program to execute specific blocks of code.

So far, we have been using predefined functions like `rect()` and `ellipse()`. These are functions that come ready with processing. Now, it is time for us to write our own functions.

Using functions makes our code simpler and easier to read, which can be extremely important when writing large complex programs.

Try to avoid copying and pasting code and use functions as often as possible.



Syntax

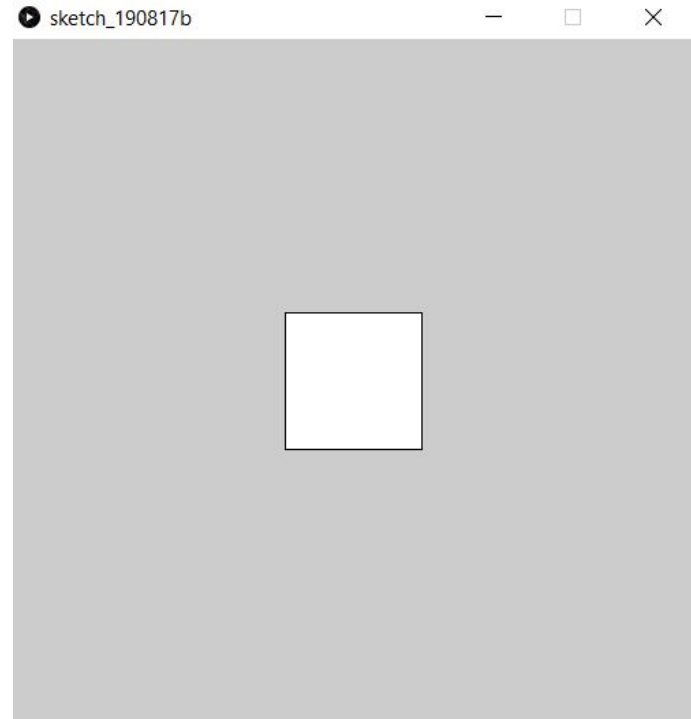
Functions must be created or *declared* outside of `setup()` and `draw()`. A function declaration would look something like this:

```
File Edit Sketch Debug Tools Help
sketch_190817b
1 void cube(float cubeX, float cubeY)
2
3
4   rect(cubeX, cubeY, 100, 100);
5
6 }
7
8
9
10
```



After the function is declared, it can be used in draw() or setup():

```
1 void cube(float cubeX, float cubeY)
2 {
3
4   rect(cubeX, cubeY, 100, 100);
5
6 }
7
8 void setup(){
9   size(500,500);
10 }
11
12 void draw(){
13
14   cube(200, 200);
15
16 }
17
18
19
```





Now, let's look at the parts and syntax of that function.

The *parameters* are variables that are created and used within the function.

The return type determines the type of value the function will hold.

Name of the Function

Parameters of the Function

Return Type

```
File Edit Sketch Debug Tools Help  
sketch_190817b  
1 void cube(float cubeX, float cubeY)  
2  
3  
4 rect(cubeX, cubeY, 100, 100);  
5  
6 }  
7
```

Code that will run when the function is called.



Function Use

After you name the function, you then later use that name to use or *call* the function. When called, the program will then run the code inside the function using the parameters it was given.

```
void cube(float cubeX, float cubeY)
{
    rect(cubeX, cubeY, 100, 100);
}
```



Function is named "cube"

"cube" is called in draw()

```
void draw(){
    cube(200, 200);
}
```





Parameters

The parameters of the function are the values that must be put in when the function is used. These values are then used inside the function. You may have any number of parameters in a program, and they must be separated by a comma. Parameters are defined inside the brackets as the function is declared and cannot be used outside of the function.

Parameter Variables Used

```
void cube(float cubeX, float cubeY)
{
    rect(cubeX, cubeY, 100, 100);
}
```

Parameters Declared

```
void draw(){
    cube(200, 200);
}
```

Parameters input



Return type

The return type has to do with the type of value that the function will hold. “Void” means that the function will hold no value. “Void” functions are simply called and will execute the code inside them and will not give back a value.

Return type



```
void cube(float cubeX, float cubeY)
{
    rect(cubeX, cubeY, 100, 100);
}
```




Return Type Example

Here is an example of a program with a float return type. This program receives the side length of a cube, and returns the volume. Notice that in the program, we use the line:
`x = cubeVolume(10);`

The variable “x” becomes equal to the return value of the function.

```
File Edit Sketch Debug Tools Help

sketch_190818a

1 float cubeVolume(float sideLen){
2   float volume = sideLen * sideLen * sideLen;
3   return volume;
4 }
5
6 void setup() {
7
8   float x;
9   x = cubeVolume(10);
10  println(x);
11 }
12
13
14

1000.0

Console Errors
```



Return Value

A function with a return type other than void must have a return statement in it. The value after return is the value that will be given back when the function is called, it will be what the function is “equal to” when called. The function must return a value of the same type as it’s return type, so in the case of our example, it must return a float. Any variable type can be used as the return type.

Return Statement

The function will be equal to the value of “volume”.

```
float cubeVolume(float sideLen){  
    float volume = sideLen * sideLen * sideLen;  
    return volume;  
}
```