I feel like some of y'all are intimidated by functions, so I wanna try my hand at explaining how this works.  Please tell me if this is helpful!!!  I'm hoping to be a tutor for this class next semester & I appreciate your feedback.  (Feel free to reply here OR send me a private message in canvas.)

Let's look at the flow of data through this program:

**function main() {**  
**let name = getName();**  
**let number = getNumber();  
    number += 10;  
    display(name, number);**  
**}**

**function getName() {**  
**let x = prompt('enter your name');  
    let y = 100;  
    return x;**  
**}**

**function getNumber() {**  
**let x = Number(prompt('enter your favorite number'));**  
**return x;**  
**}**

**function display(aaa, bbb) {**  
    **console.log(aaa + '\'s favorite number is ' + bbb);  
    console.log(name, number);  
}  
/// start doing stuff after this line ///  
main();**

First, all four functions get their behavior's loaded into memory, but do NOTHING because they weren't actually invoked yet...  THEN you see the comment, and right below it the **main()** function is called. In this type of program, a 'main' function (which can actually be called whatever you want) will serve as a controller to direct the flow of your code.  Remember that whenever a function is completed, the interpreter will move to the line immediately after that function was called.

First, **main()** calls to **getName()** and then **getNumber()** each time storing the returned value as a new variable (existing within the scope of 'main'). Then **main()** changes the number and finally calls to **display()**, passing the '**name**' and '**number**' as arguments.

If you look, both **getName()** and **getNumber()** return '**x**' but since each '**x**' was defined in a separate function, they may as well have different names because they were defined within different scopes. Each time the value (formerly known as) '**x**' is returned, it immediately gets re-assigned by the **=** operator, and '**x**' is essentially dropped from memory.  Note that the '**y**' is also dropped once **getName()** completes because the **return** can only hold one piece of data and '**y**' was left hanging.

Now look at the **display()** function.  See how the parameters are '**aaa**' and '**bbb**' ? When **display(name, number)** was called, the arguments '**name**' and '**number**' have their values inserted into the places of parameters '**aaa**' and '**bbb**' for processing by **display()**.

'**aaa**' and '**bbb**' only exist within the scope of **display()**, so if you were to try to use them anywhere else, they would essentially not exist... AND as far as **display()** is concerned '**name**' or '**number**' don't exist! **display()** will take whatever two values are passed to it and assign them to '**aaa**' and '**bbb**' so the second **console.log()** will either say 'undefined' or generate an error message (depending on the browser).

Edited by [Pete Wurster](https://ccp.instructure.com/courses/1344587/users/4218136) on Oct 25 at 10:05pm