Posted [on Jul 7, 2015](https://teamtreehouse.com/community/function-parameter-and-global-variable-question) by Gabriel D. Celery

# Function parameter and global variable question

I have a little bit of an issue understanding how JS handles local and global variables if they have the same name.

If I type in this:

var number = 10;

function test(parameter) {

number = parameter + 2;

}

test(number);

document.write(number);

I get the desired effect and the function changes the global variable.

But if I type in this:

var number = 10;

function test(number) {

number = number + 2;

}

test(number);

document.write(number);

Then the global variable is not affected. I assume this happens because JS treats the left "number" in the line of

number = number + 2;

as a local variable since it has the same name as the parameter, but is there any way to tell JS that I want to affect the global variable, or I have to pay attention every time how I name all the stuff and write the code?

[Jason Desiderio](https://teamtreehouse.com/jasondesiderio)

on Jul 8, 2015

It is something you'll have to pay attention to while you're writing the code. As you start writing larger applications, name spacing becomes a much much bigger concern. You will find quickly that writing a variable to the global name space is generally a bad idea and you should keep everything at the function level.

## **1 Answer**

**[Igor Yamshchykov](https://teamtreehouse.com/ihoryamshchykov)**

on Jul 10, 2015

As Jason said it is actually a bad idea to keep your data in global scope, but sometimes you might need that.

and to answer your question:

The global variable is not affected in local function, because there is a parameter with same name and function changes that parameter. You are correct with that.

If you want to modify the global variable you might use this approach

var number = 10;

function test(number) {

return number + 2;

}

number = test(number);

it will return the modified value and you can assign that value to global variable.

Also you can use objects for that, but it might be a little more complicated

var number = {

value:10

}

function test(number) {

number.value = number.value + 2;

}

test(number);

you might got an error here if number wont have any value, or you don't pass a number at all; so you should also use check

function test(number) {

if(number && number.value){

number.value = number.value + 2;

}

}

This will work, because when you pass an object (or array) you are passing a reference to that object, and it is possible to modify the contents of that object.

Hope that helps and wont confuse you.