

Local and Global Variables

Relevant Links

- Flanagan's book, sections 3.5, 3.9, 3.10
- MDN's guide¹

Variables in Javascript

- Apart from a few reserved words, most other identifiers can be used as variable names.
- You declare a local variable via the keyword `var`. You can also give it a value at the same time.
- To begin with, things are defined in the global scope. A new scope is created within the body of any function.
- One particularly interesting behavior of Javascript is “variable hoisting”: *Variable declarations are always processed as if they happened at the top of the function.*

For example consider the following code:

```
function f() {  
    a = 2;  
    var a = 1;  
}
```

This is equivalent to:

```
function f() {  
    var a;  
    a = 2;  
    a = 1;  
}
```

- A local variable is not visible outside the scope of the function in which it was defined.
- Variables that are assigned a value without a corresponding declaration are “global”. Global variables are visible everywhere.
- Global variables are really nothing more than properties of the global object:

```
a = 2;           // Defining a global variable  
window.a;        // That variable exists as part of the global object  
window.b = 4;    // Can define global variables as properties of the global object  
b;               // Returns 4
```

¹https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Values,_variables,_and_literals#Variables

Note: Very few things in Javascript are protected. For instance this line overwrites the Math object:

```
Math = {}
```

- Here is a gotcha to watch out for, there are no other scopes than those introduced by functions. For example a for loop does NOT introduce a new scope.

```
for (var i = 0; i < 10; i += 1) {  
    i;  
}  
i;    // i is still declared outside the loop, and with value 10
```

- Pay particular attention to the example at the top of page 54.
- Make a point to **always declare variables at the top of functions/files**.
- Here is an example of what can go horribly wrong if you are not careful: [local_global.html](#)² and [local_global.js](#)³

Note: Files loaded via <script> tags all share the same global space. Whatever you do in one file can impact the other files.

²[../testPages/local_global.html](#)

³[../testPages/local_global.js](#)