Local and Global Variables

Relevant Links

- Flanagan's book, sections 3.5, 3.9, 3.10
- MDN's guide[1]

Variables in Javascript

- Apart from a few reserved words, most other identifiers can be used as variable names.
- Modern Javascript has a number of different variable "scopes":
  - Global variables can be accessed anywhere.
  - Old-style local variables are available *anywhere* within the function on which they are declared. They are defined via the keyword var. AVOID THESE. We will not discuss them further.
  - New-style local variables are available within the innermost set of braces in which they are declared. They are defined via the keywords let and const (for constant variables).

  CAUTION: If you assign to a variable name without declaring it, this will create a global variable and you will get no warning about it.
- A local variable is not visible outside the scope 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:
  js a = 2; // Defining a global variable window.a; // That variable exists as part of the global object window.

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

  ```
  Math = {}
  Math.sin(2);     /// An error now
  ```
- Pay particular attention to the example at the top of page 54.
- Make a point to **always declare variables using let or const**.
- Here is an example of what can go horribly wrong if you are not careful: local_global.html[2] and local_global.js[3]

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

[2] ../testPages/local_global.html

[3] ../testPages/local_global.js

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