

# Closure

A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created, at function creation time.

---

# Why Clouser?

- JavaScript variables can belong to the local or global scope.
  - Global variables can be made local (private) with closures.
  - In a web page, global variables belong to the window object.
  - Global variables can be used (and changed) by all scripts in the page (and in the window).
  - A local variable can only be used inside the function where it is defined. It is hidden from other functions and other scripting code.
  - Global and local variables with the same name are different variables. Modifying one, does not modify the other.
  - Global variables live until the page is discarded, like when you navigate to another page or close the window.
  - Local variables have short lives. They are created when the function is invoked, and deleted when the function is finished.
  - **A closure is a function having access to the parent scope, even after the parent function has closed.**
-