

# Basic JavaScript Part 11: Functional Initialization

---



March 4th, 2011

Here are the links to the previous installments:

1. [Functions](#)
2. [Objects](#)
3. [Prototypes](#)
4. [Enforcing New on Constructor Functions](#)
5. [Hoisting](#)
6. [Automatic Semicolon Insertion](#)
7. [Static Properties and Methods](#)
8. [Namespaces](#)
9. [Reusing Methods of Other Objects](#)
10. [The Module Pattern](#)

I just want to quickly share some beautiful JavaScript code I picked up while watching the most excellent screencast [11 More Things I Learned from the jQuery Source](#) by [Paul Irish](#).

```
var base = dom.getElementsByTagName('base')[0] || (function() {  
    // Do some stuff  
    return someElement.insertBefore(dom.createElement('base'), someElement.firstChild) ;  
})();
```

This single line of code basically checks whether there's a [base tag](#) somewhere in the DOM. If there is one, then it assigns the reference for the first element to the *base* variable. If it's not in there, then a self-executing function inserts a new base tag into the DOM and returns the reference to the new element.

I don't know about you, but I think this is pretty neat.



Jan Van Ryswyck