- Appendix C
- JSLint
- Undefined Variables and Functions
  - JSLint expects that all variables and functions will be declared before they are used or invoked
  - identify these to JSLint by including a comment in your file that lists the global functions and objects that your program depends on, but that are not defined in your program or script file
- Members
  - At the bottom of its report, JSLint displays /*member*/ comment. It contains all of the names and string literals that were used with dot notation, subscript notation, and object literals to name the members of objects.
- Options
  - JSLint accepts an option object that allows you to determine the subset of Javascript that is acceptable to you.
- Semicolon
  - JSLint checks that every statement be followed by ;
- Line Breaking
- Comma
- Required Blocks
  - expects the use of braces in block statements
- Forbidden blocks
  - JSLint expects blocks with function, if, switch, while, for, do na try statements and nowhere else
- Expression Statements
- for in Statement
  - loops through the names of all of the properties of the object. Unfortunately, it also loops through all of the members that were inherited through the prototype chain.
- switch Statement
  - JSLint checks that each case within the switch contains a break throw or return
- var Statement
  - JSLint
    - var declared only once, and before use
    - function declared before use
    - parameters will not also be declared as vars
- with Statement
  - JSLint does not expect to see a with statement
- =
- == and !=
  - do type coercion before comparing
  - using === and !== is the preferred method
- Labels
  - JSLint expects labels only on statements that interact with break: switch, while, do, and for
- Unreachable Code
  - JSLint expects that a return, break, continue, or throw statement will be followed by a } or case or default
- Eval
  - eval function and its relatives (Function, setTimeout, and setInteral) provide access to the Javascript compiler.

    - the most misused feature of javascript
- void
    - don’t use
- Constructors and new
- Not Looked For
    - JSLint does not do flow analysis to determine that variables are assigned values before they are used
    - JSLint attempts to determine that functions used with new are really constructors, or that method names are spelled correctly
- HTML
- Report
    - If JSLint is able to complete its scan, it generates a function report. It lists the following for each function
-