

Load and Report
*.pdf, *.xdp

Show scripts with errors only Show formcalc scripts

Stop on First Error

Strict white space

Strict white space indentation 4

Allow one var statement per function

Tolerate eval

Tolerate sloppy line breaking

Tolerate unfiltered for in

Tolerate inefficient subscripting

Disallow undefined variables

Disallow no-op expressions

Disallow leading _ in identifiers

Disallow == and !=

Disallow ++ and --

Disallow bitwise operators

Disallow . in RegExp literals

Require Initial Caps for constructors

Require parens around immediate invocations

Usage:

For explanations of the options, please refer to: <http://www.jshint.com/#options>

Note that JSLINT does not handle E4X literals and expressions.

In the script display, global references are highlighted. Make sure these refer to valid form objects and are not mis-spelled variables.

JStests.pdf

form1.#subform.check.result : calculate

```
1/* equality statement where the
2  operands are coerced to the
3  same type
4 */
5
6 0 == "\n";
   ^
```

Warning: Use '===' to compare with '0'.

form1.#subform.check.result : calculate

```
1// Missing break statements
2
3var myString = "";
4
5switch (4) {
6  case 0: myString = "Zero";
   ^
```

```
Warning: Expected a 'break' statement before 'case'.
7   case 1: myString = "One";
      ^

Warning: Expected a 'break' statement before 'case'.
8   case 2: myString = "Two";
      ^

Warning: Expected a 'break' statement before 'case'.
9   case 3: myString = "Three";
      ^

Warning: Expected a 'break' statement before 'case'.
10  case 4: myString = "Four";
      ^

Warning: Expected a 'break' statement before 'case'.
11  case 5: myString = "Five";
12}
13myString;
```

form1.#subform.check.result:calculate

```
1 // Ambiguous if/else statement
2
3 var result = "unknown";
4 var card = "KING";
5 var suit = "HEARTS";
6
7 if (card == "ACE")
8     if (suit == "SPADES")
9         result = "Winner";
10
11 else
12     result = "Loser";
13
14 this.rawValue = result;
```

form1.#subform.check.result:calculate

```
1 // Nested comments
2
3 var q;
4 var s = 1;
5 /*
```

```
6 s = 2;
7
8 // /* */
   ^
   Warning: Nested comment.
   Warning: Stopping, unable to continue. (50% scanned).
9
10 s = 3;
11
12 /* */
13
14 s;
```

form1.#subform.check.result: calculate

```
1 // Use XFA reserved properties as variables
2 var x = 4;
3 var y = 6;
4 this.rawValue = x + y;
   Warning: "x" conflicts with an XFA property or method name.
   Warning: "y" conflicts with an XFA property or method name.
```

form1.#subform.check.result: calculate

```
1 /*members length rawValue */
2
3 var fruits = ["apple",
4              "orange",
5              "peach",
6              "plum"];
7 // truncate the array
8 fruits.lenght = 2;
   ^
   Warning: Unexpected /*member 'lenght'*.
9
10 this.rawValue = fruits.length;
```