

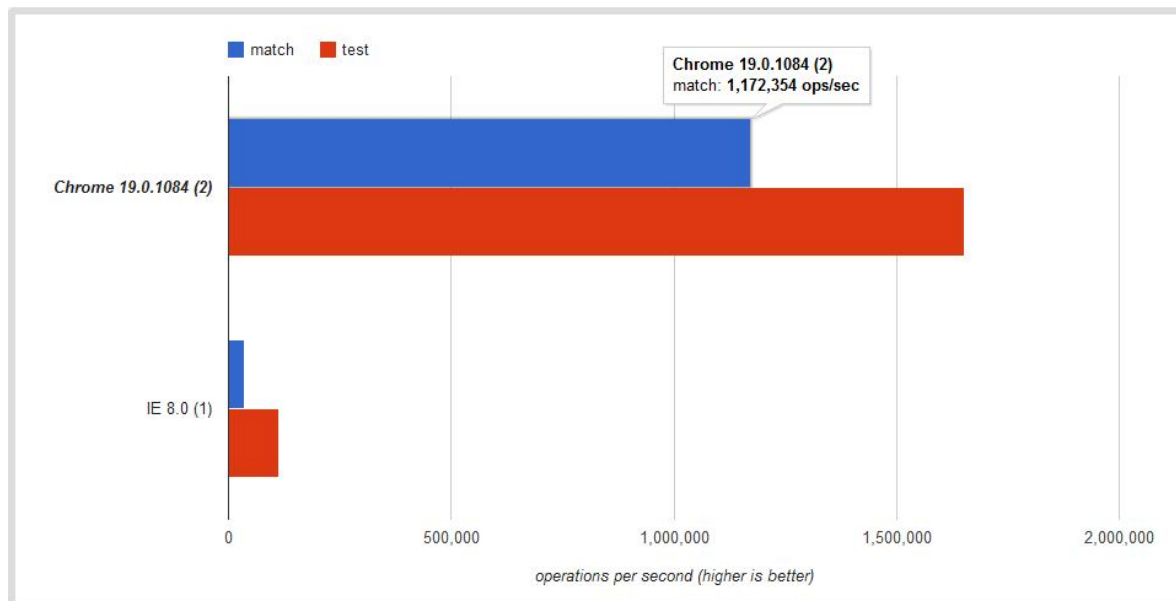
# 1 Some combined hate on Javascript

## 1.1 RegEx

Regex in .js are the same as goddamn ploughin anywhere, but the implementation of this greasy-bottom muffPutter kinda differs from the usual Java.

- `regexObject.test( String )`  
Executes the search for a match between a regular expression and a specified string.  
Returns true or false.
- `string.match( RegExp )`  
Used to retrieve the matches when matching a string against a regular expression.  
Returns an array with the matches or null if there are none.  
and null evaluates to false.

Performance-Wise, use `.test()`.



*Wasted Time: About 20 Minutes*

## 1.2 Objects

Objectdeclaration in Javas@#t is really goddamn alienlike.

```
var car = {  
  type:"Fiat",  
  model:"500",  
  color:"white"
```

```
};
```

or in 1 Line:

```
var car = {type:"Fiat", model:"500", color:"white"};
```

Access via object.attribute. Don't ever try to leave this heresy in your code:

```
var x = new String(); || new Number(); || new Boolean();
```

You could've just global-var that sucker. But!

Do NOT create global variables unless you intend to.

*Wasted Time: 2 Hours and stacking*

### 1.3 Calling Functions

There is no 'get-over-here by Reference', unless objectified like women.

```
function passVar(obj1, obj2, num) {  
  obj1.prop = "laptop";           // will CHANGE original  
  obj2 = { prop: "computer" };    //will NOT affect original  
  num = num + 1;                  // will NOT affect original  
}
```

```
var object1 = {  
  prop: "car"  
};  
var object2 = {  
  prop: "bike"  
};  
var number1 = 10;
```

```
passVar(object1, object2, number1);  
console.log(object1); //output: Object {item:"laptop"}  
console.log(object2); //output: Object {item:"bike"}  
console.log(number1); //ouput: 10
```

*Wasted Time: About 5 Minutes*