

JavaScript: There's a real language in there

Venkat Subramaniam

venkats@agiledeveloper.com

twitter: [venkat_s](https://twitter.com/venkat_s)



What's Wrong with JavaScript?

JavaScript is a very powerful language, yet

- ✿ often hated
- ✿ suffers browser inconsistencies
- ✿ misunderstood
- ✿ developers find it painful
- ✿ lags in tool support
- ✿ bad name for a language!

However,...

In reality, however, it is a very elegant, powerful, object-oriented, and very expressive language

What does it feel like?

It's feels like Java and C for most part

Similar to Perl in some ways

Untyped language

Some Basics

Case sensitive

It is a good idea to use ; to separate statements, though it is optional

Same commenting style as C++/Java

Types

null and undefined are used to indicate null

Strings are single quoted or double quoted with \ used for escape

Strings are immutable

Functional in Nature

Functions are first-class citizens

Variables

Variables are typeless

Highly recommend that you use `var` to declare variables

If you omit `var`, variable becomes global

Any change made elsewhere will affect it!

Variable Scope

Global and local (function)

No block level scope

Variable defined anywhere in a function has function scope, though not initialized until var is reached

Control Structure

Most control structures are like C

Statements

Expressions

if, switch, while, ...

Exception handling much like Java—try, catch, finally

Two types of for loops

C like for(...; ...; ...) { ... }

for ([var] variable in object) { ... }

iterates over all properties of the object

Functions

You can pass arguments

Function may return a value

```
function name(argumentName, ...) {  
    ... code ...  
    return value;  
}
```

To call

```
var someThing = name(arg1, arg2);
```

Functions are Objects

```
function foo() {...}
```

is the same as

```
var foo = function() {...}
```

This is very useful to assign handlers to objects for events, etc.

Working with Objects?

Use new to create objects

```
var obj = new SomeThing();
```

Objects have properties and methods

```
obj.someProperty = value;
```

```
var someValue = obj.someProperty;
```

```
obj.someMethod(...);
```

Working with Objects?

You can also treat an object as associative array

This offers quite a bit of convenience to make your code dynamic

```
obj["somePropertyName"] = value
```

```
var someValue = obj["somePropertyName"]
```

How to create a Class?

OK, but how do we create a class?

Using a function!

Notice how we assign property of the class

Creating a Class

```
function SomeThing() {  
    this.someproperty = 0;  
}
```

Everything in a class is public

Adding Methods

Each class has a prototype member that holds its properties and methods

So, add the method to the prototype

```
Something.prototype.someMethod = function(val) {  
    alert(val);  
}
```

A Sample Class

```
function Car() {  
    this.miles = 0;  
}  
  
Car.prototype.drive = function(dist) {  
    this.miles += dist  
}  
  
var car = new Car()  
  
println(car.miles)  
  
car.drive(12)  
  
println(car.miles)
```

How's an object created?

An object is created by copying the prototype to `__proto__` property of an object

An object passes unhandled calls to the `__proto__`

Navigating

```
for(var property in obj) { println(property); }
```

Will list properties and methods

```
for(var method in obj.__proto__) { println(method); }
```

Inheritance?

No and Yes!

Don't view inheritance like you view it in Java/C++/C#

You inherit methods, but in a distinctively different manner!

Inheritance

```
function someThingMore() {}
```

```
someThingMore.prototype = new SomeThing()
```

Copies methods of someThing to SomeThingMore

```
var obj2 = new SomeThingMore();
```

```
obj2.someMethod(...);
```

Remember to set prototype before adding any methods

Code Quality

jslint is a nice tool to check code quality

Unit Testing

JavaScript is highly unit testable

You need to separate the logic from the HTML page

Thank You!

Venkat Subramaniam
venkats@agiledeveloper.com
twitter: venkat_s

