JavaScript is …

*a dynamic, weakly typed, prototype-based anguage with first-class function
* JavaScript == ECMAScript == JScript

Dynamic:
- compilation and execution happen together

```
var propMap = {
  val: "value", html: "innerHTML"
}

for(var fnName in propMap){
  $.prototype[fnNam] = (function(prop){
    return function(){
      return this[prop];
    }
  })(propMap[fnName]);
}
```

Weakly Typed:
- type is associated with value, not variable (type travels with value not variable)

First-class function:
- can treat a function like an object [create, return, arg(s)]

Prototype-Based:
- looks up inherited and shared properties

DATA TYPES, OPERATORS & PRIMITIVES :

Data Types::
undefined undefined—> refers to a pointer who's point has not been set
Null null —> refers to null pointer (it is by intent)
Boolean true
String "hello"

Number 2
Object {name: “value”}
Array [1,2,3]
Date new Date()
RegExp /.*/g
Function function(){}

Operators::
var [var foo]
new [new Foo]
assignment [foo = {bar : “a value”}]
delete[delete foo.bar]
member[foo.bar] or [foo[“bar”]]
call [bar()]
comparison [‘==‘, ‘===‘]

```
var me = {
  name: {first: "justin"}
},
name = me.name;
name = {first:"alexis"};
me.name.first //justin
```

delete :
delete doesn’t actually delete the memory object // delete only deletes a referrence to an obj

Type	Result
Undefined	"undefined"
Null	"object"
Boolean	"boolean"
Number	"number"
NaN	"number"
String	"string"
Function	"function"
Array	"object"
Any other object	"object"