

# JavaScript

**Ricard Meyerhofer**

**2015-2016 Q2**

**Albert Rubio**

# Introducció

- Creat el 1995 per Brendan Eich
- Molta gent pren com a llenguatge per a principiants.
- Llenguatge enfocat a la part del client web
- 1996 Microsoft va treure JScript com a resposta a JavaScript (per al seu ús a Internet Explorer)
- 1997 participa a ECMA on s'estandarditza sota el nom de ECMAScript
- 1998 i 1999 surten ECMAScript2 i ECMAScript3
- 2000 proliferació d'implementacions per la part del servidor

# Introducció

- Ajax prolifera frameworks i llibreries augmentant el seu ús fora de l'àmbit de web.
- 2009 CommonJS apareix per a especificar una llibreria d'ús comú principalment per a entorns fora de l'àmbit web.
- 2015 surt ECMAScript 6 on JavaScript incorpora mòduls d'organització del codi, vertaderes classes, iteradors, etc

# Llenguatges Similars

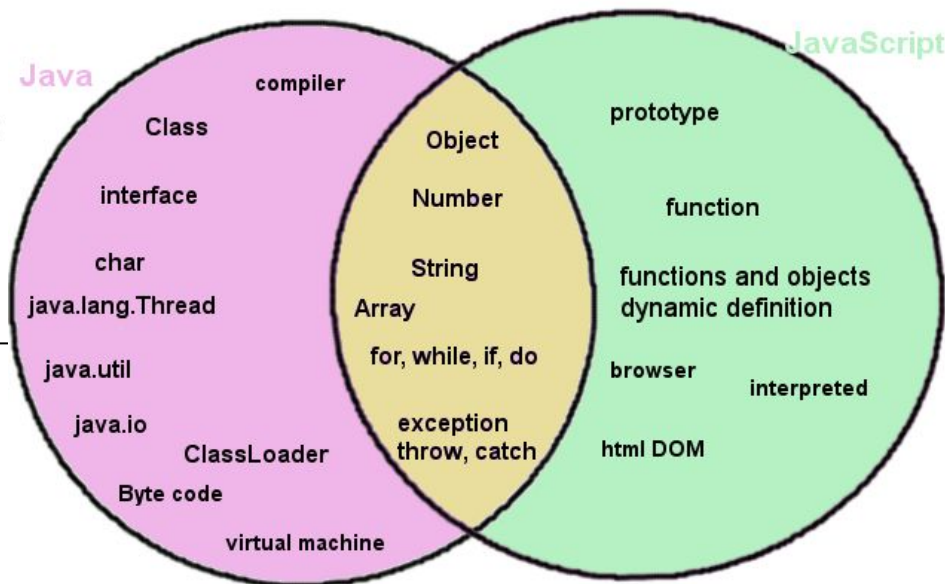
JavaScript s'ha inspirat de:

- C
- Scheme
- Self
- Perl
- Python

S'han inspirat de JavaScript:

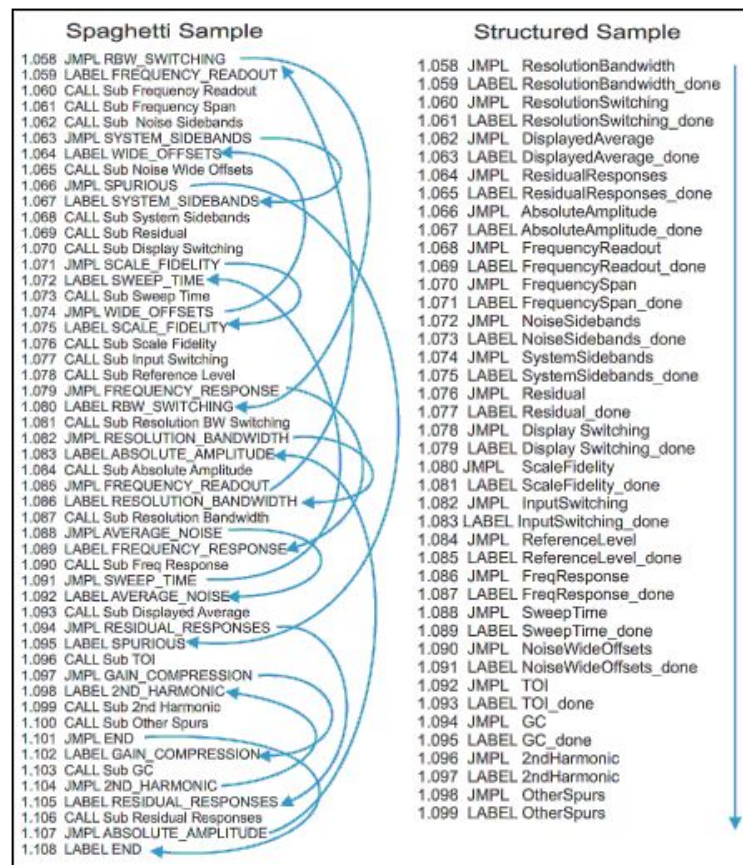
- ObjectiveJ
- JScript
- JScript.NET
- TIScript.
- CoffeeScript
- IO
- TypeScript

# Llenguatges Similars



# Paradigmes

- Imperatiu
- Procedimental
- Estructurat
- Orientat a objectes
- Basat en prototipus
- Funcional



# Paradigma funcional

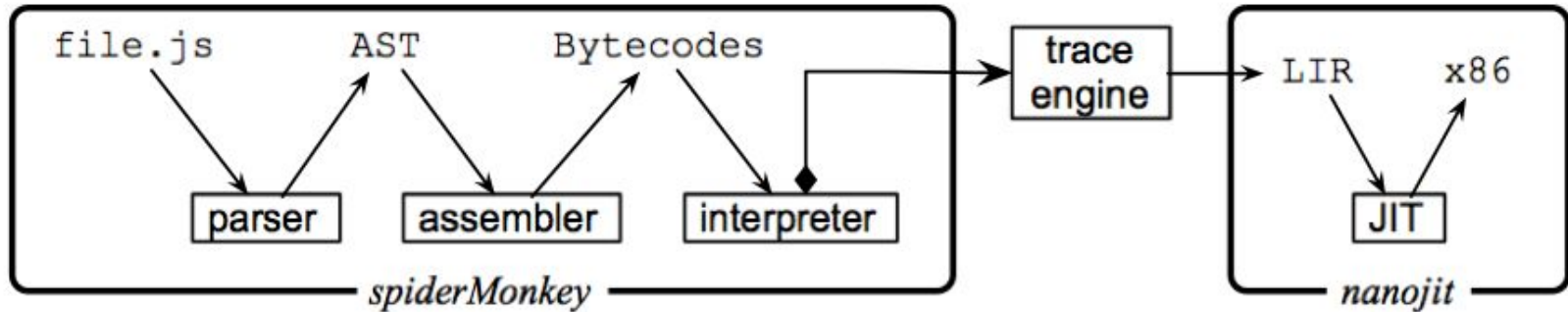
- Pattern matching
- Monads
- Closures
- Funcions d'ordre superior
- Funcions anònimes

```
var $ = fun.parameter,  
    fact = fun(  
        [0, function () { return 1; }],  
        [$, function (n) { return n * fact(n - 1); }]  
    );
```

```
var displayClosure = function() {  
    var count = 0;  
    return function () {  
        return ++count;  
    };  
}  
var inc = displayClosure();  
inc(); // devuelve 1  
inc(); // devuelve 2  
inc(); // devuelve 3
```

# Compilat/Interpretat?

- Interpretat
- Compilació JIT (Just In Time)





# Sistema de Tipus

- Feble
  - Operacions entre diversos tipus
- Dinàmic
- Llenguatge de tipus segur
- JavaScript disposa de la variable var que s'usa per a tot

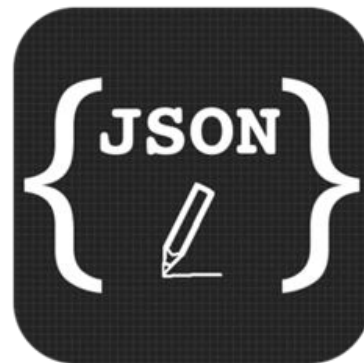
# Aplicacions

- Entorn web
  - Cercador
  - Enviar informació
  - Validació de continguts
- Programació de servidors amb Node.js
- Desenvolupament de videjocs
- Aplicacions



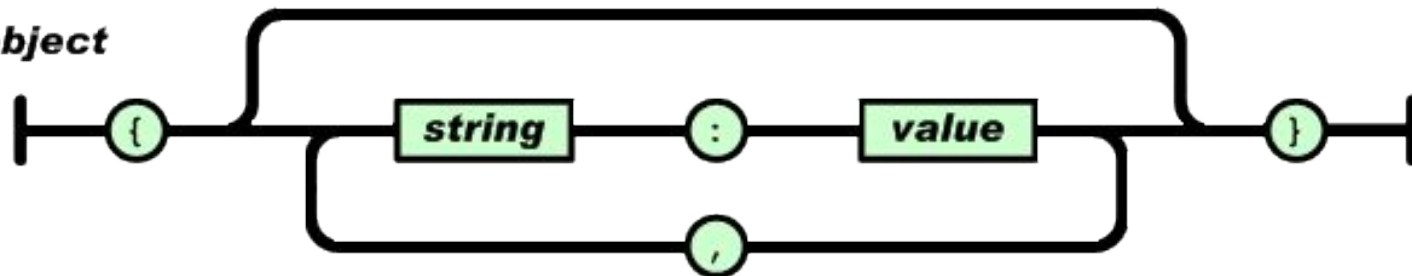
# Altres Característiques

- Expressions regulars
- JavaScript **O**bject **N**otation
  - Format lleuger d'intercanvi de dades
  - Constituit per dues estructures:
    - Parells nom/valor (diccionari)
    - Llista ordenada de valors (vectors)
  - Actualment molt popular per la seva facilitat de lectura i tractament, competència directa del XML

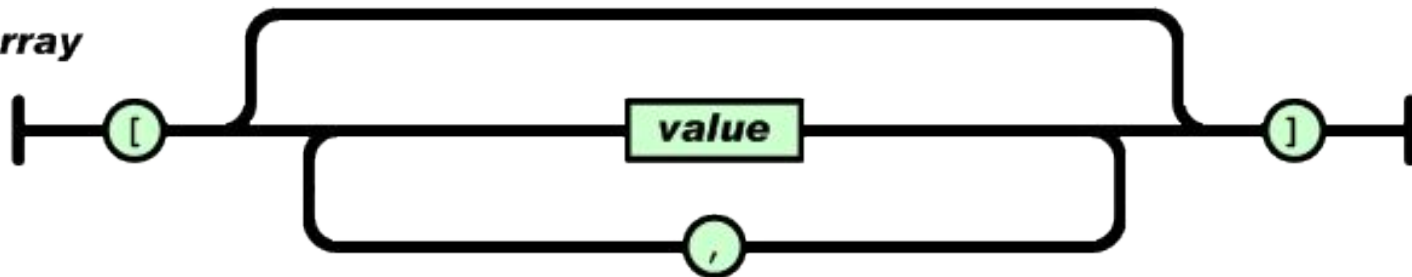


# Altres Característiques

**object**



**array**



object  
  {  
    { members }  
 members  
  pair  
  pair , members  
pair  
  string : value  
array  
  [  
  [ elements ]  
elements  
  value  
  value , elements  
value  
  string  
  number  
  object  
  array  
  true  
  false  
  null

# Altres Característiques

```
{ "employees": [  
  { "firstName": "John", "lastName": "Doe" },  
  { "firstName": "Anna", "lastName": "Smith" },  
  { "firstName": "Peter", "lastName": "Jones" }  
]}
```

```
<employees>  
  <employee>  
    <firstName>John</firstName> <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName> <lastName>Smith</lastName>  
  </employee>  
  <employee>  
    <firstName>Peter</firstName> <lastName>Jones</lastName>  
  </employee>  
</employees>
```

# Exemple

```
function hamming() {  
  var queues = {2: [], 3: [], 5: []};  
  var base;  
  var next_ham = 1;  
  while (true) {  
    yield next_ham;  
  
    for (base in queues) {queues[base].push(next_ham * base)}  
  
    next_ham = [ queue[0] for each (queue in queues) ].reduce(function(min, val) {  
      return Math.min(min, val)  
    });  
  
    for (base in queues) {if (queues[base][0] == next_ham) queues[base].shift()}  
  }  
}
```