

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font.

JS

Junjie Ji Chen

LP 2018-19 Q2

Índex

1. Introducció i història
2. Característiques
3. Principals aplicacions
4. Llenguatges similars
5. Exemples de codi
6. Bibliografia

Introducció i història

A causa de la necessitat de crear pàgines web amb contingut dinàmic, va sorgir LiveScript, que va ser creat l'any 1995 pel programador **Brendan Eich**. Posteriorment LiveScript va adoptar el nom de JavaScript per motius de màrqueting (ja que en aquella època la paraula Java estava de moda en el món informàtic). Actualment JavaScript és un llenguatge molt popular pel desenvolupament de pàgines web, la seva funcionalitat principal és la de poder controlar arxius de multimèdia, imatges, animacions, etc.

Característiques

Multiparadigma

JavaScript és un llenguatge que combina diferents paradigmes: Imperatiu, funcional i orientat a objectes.

Interpretat i amb compilació just-in-time

JavaScript és un llenguatge interpretat, és a dir, no cal fer el procediment de traduir el codi font a llenguatge màquina (procés de compilació), els navegadors poden llegir directament del codi font. Hi ha navegadors més moderns que tenen la capacitat de fer la compilació just-in-time.

Sistema de tipus

JavaScript és un llenguatge de tipat **feble**, és a dir, es pot barrejar valors de diferents tipus, per exemple:

```
let a = 1;
let b = "hello";

writeln(a+b); // Escriurà "hello1"
```

Com que JavaScript és un llenguatge interpretat, no té gaire sentit parlar si és de type safe, ja que no hi ha una fase de compilació prèvia. En el cas de la compilació *just-in-time*, JavaScript aplicarà operacions necessàries per evitar errors de tipus, per tant sí que és type safe.

La comprovació de tipus és **dinàmica**, és a dir, en temps d'execució, i té 6 tipus primitius que són els següents:

```
let num1 = 123;
let num2 = 12.34; // num1 i num2 són de tipus Number
let str1 = "Hola";
let str2 = 'Adeu';
let str3 = `Bon dia`; // str1, str2 i str3 són de tipus String
let bool = true; // bool és de tipus Boolean
let x = null; // x és de tipus Null (No és un punter, null significa un valor buit)
let y; // y és de tipus Undefined (Semblant a Null)
let z1 = Symbol();
let z2 = Symbol("string") // z1 i z2 són de tipus Symbol
let z3 = Symbol() == Symbol(); // Un símbol és un tipus que fa d'identificador, per tant
cada símbol és únic, és a dir, z3 sera fals.
```

Principals aplicacions

JavaScript té múltiples aplicacions, però on realment destaca és l'entorn de la **programació web**, juntament amb HTML i CSS, HTML s'encarrega la part estructural, CSS s'encarrega la part de disseny i presentació i JavaScript s'encarrega de la part dinàmica (comportament), és a dir, qualsevol cosa que canvia d'estat en aquella pàgina, com per exemple rellotges, animacions, videojocs, etc.

JavaScript actua en la part del client i no la part del servidor, això evita que el client estigui en connexió constant amb el servidor, minimitzant i evitant col·lapsant el flux d'informació per les xarxes.

Llenguatges similars

Tot i que JavaScript i Java comparteixen el mateix nom, són dos llenguatges diferents. Una de les poques similituds que comparteix és la sintaxis (de fet són semblances que comparteixen amb C).

JavaScript va està influenciat per Scheme, Perl, Python i Self.

Exemples de codi

```
console.log("Hello world!");
```

Exemple 1: Hello world! [P68688](#).

```
function isPrimeIterative(n) { // versió iterativa
    prime = true;
    if (n <= 1) prime = false;
    for (let i = 2; i*i <= n; ++i) {
        if (n%i == 0) prime = false;
    }
    return prime;
}

function isPrimeRecursiveAux(n, i) {
    if (n <= 1) return false;
    if (n == 2) return true;
    if (n%i == 0) return false;
    if (i*i <= n) return isPrimeRecursiveAux(n, i+1);
    return true;
}

function isPrimeRecursive(n) { // versió recursiva
    return isPrimeRecursiveAux(n, 2);
}
```

Exemple 2: Solució iterativa i recursiva per saber si un numero és primer.

```
function suma(y) {
  function f(x) {
    return x + y;
  }
  return f;
}
let sum = suma(4);
console.log(sum(3)); // imprimirà 7 (3 + 4)
```

Exemple 3: Ús de les funcions de curry.

```
function quicksort(v) {
  if (v.length == 0) return [];
  let left = [];
  let right = [];
  let pivot = v[0];
  divide(pivot, v, left, right);
  return merge(pivot, quicksort(left), quicksort(right));
}

function merge(pivot, resultleft, resultright) {
  return resultleft.concat(pivot, resultright);
}

function divide(pivot, v, left, right) {
  for (let i = 1; i < v.length; ++i) {
    if (pivot > v[i]) left.push(v[i]);
    else right.push(v[i]);
  }
}
```

Exemple 4: Quicksort en JavaScript.

```
class Node {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }

  addLeft(value) {
    let node = new Node(value);
    this.left = node;
  }

  addRight(value) {
    let node = new Node(value);
    this.right = node;
  }

  eraseLeft() {
    this.left = null;
  }
}
```

```

    }

    eraseRight() {
        this.right = null;
    }
}

function size(node) {
    if (node == null) return 0;
    return size(node.left) + size(node.right) + 1;
}

function max(x, y) {
    if (x > y) return x;
    return y;
}

function height(node) {
    if (node == null) return 0;
    return max(height(node.left), height(node.right))+1;
}

function preorder(node) {
    if (node == null) return [];
    else {
        let result = [node.value].concat(preorder(node.left), preorder(node.right));
        return result;
    }
}

let n = new Node(1);
n.addLeft(2);
n.addRight(3);
n.left.addLeft(4);
n.left.addRight(5);
n.right.addLeft(6);
n.right.addRight(7);

console.log(n); // imprimeix l'arbre binari
console.log(size(n)); // número de nodes que té l'arbre
console.log(height(n)); // altura de l'arbre
console.log(preorder(n)); // imprimeix una llista en preorde de l'arbre

```

Exemple 5: Possible estructura d'un arbre binari.

Bibliografia

- **JavaScript**, *Wikipedia*, Font: <https://es.wikipedia.org/wiki/JavaScript>, Ultima edició: 16/05/2019
- **JavaScript**, *Documentación web de MDN*, Font: <https://developer.mozilla.org/es/docs/Web/JavaScript>, Ultima edició: 18/03/2019
- **JavaScript Data types**, *Javascript.info*, Font: <https://javascript.info/types>
- **¿Qué es Javascript?**, *Maestros del web*, Font: <http://www.maestrosdelweb.com/que-es-javascript/>, Ultima edició: 03/07/2007
- **¿Para qué se utiliza JavaScript?**, *4r productora digital + software factory* Font: <http://www.4rsoluciones.com/blog/para-que-se-utiliza-javascript-2/>
- **Diferencias entre Java y Javascript**, *Desarrolloweb*, Font: <https://desarrolloweb.com/articulos/492.php>, Ultima edició: 16/07/2001
- **A Brief History of JavaScript**, *Auth0*, Font: <https://auth0.com/blog/a-brief-history-of-javascript/>, Ultima edició: 16/01/2017