

1 Estrutura léxica de JavaScript

A estrutura léxica nos indica cómo debemos escribir programas nunha linguaxe de programación. Especifica cousas como a aparencia dos nomes das variables, os caracteres que se empregan para comentarios e cómo se separan entre si as instrucións do programa.

1.1 Sumario

- 1 Como introducir JavaScript no documento HTML
- 2 Tokens
 - ◆ 2.1 Conxunto de caracteres
 - ◆ 2.2 Comentarios
 - ◆ 2.3 Literais
 - ◆ 2.4 Identificadores
- 3 Palabras reservadas
- 4 Tratamento de variables en JavaScript
 - ◆ 4.1 Nomear variables
 - ◆ 4.2 Declarar de variables
 - ◆ 4.3 Tipos de variables
 - ◆ 4.4 Ámbito das variables
- 5 Operadores
- 6 Funcións

1.2 Como introducir JavaScript no documento HTML

Podemos introducir JavaScript nun documento HTML de dous xeitos:

- Insertándoo nun documento HTML entre un conxunto de etiquetas **<script>**.
- Insertándoo dentro dunha etiqueta HTML para responder a un evento.

JavaScript insérase dentro dos documentos HTML por medio das etiquetas de comezo e fin de *script*:

```
<script language="javascript">
//Aquí vai o script ...
</script>
```

Vexamos un primeiro exemplo de JavaScript:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ola mundo con JavaScript</title>
</head>
<body>
<center>
<script language="javascript">

</script>
</center>
</body>
</html>
```

- Vese que despois da primeira etiqueta de *script* inclúense unhas etiquetas de comentarios que engloban o código fonte JavaScript, e que evitan que dito código se mostre como texto plano nos navegadores que non entenden as etiquetas *<script>*.
- As palabras **document** e **write** son recoñecidas por JavaScript e usadas conxuntamente para realizar unha acción, neste caso escribir no documento o texto incluído no interior dos parénteses.
- As palabras **document** e **write** deben escribirse en minúsculas obrigatoriamente.

1.3 Tokens

Os **tokens** son as palabras, as frases ou os caracteres mais pequenos que o JavaScript pode entender.

Os *tokens* de JavaScript agrúpanse en cinco categorías: identificadores, palabras reservadas, literais, operadores e separadores.

A **sintaxe** da linguaxe é o conxunto de regras e restricións que nos indican como combinar estes *tokens*.

1.3.1 Conxunto de caracteres

Os programas JavaScript escríbense empregando o conxunto de caracteres **Unicode**.

- **Distinguir maiúsculas e minúsculas.**

JavaScript é unha linguaxe que distingue maiúsculas de minúsculas, o que significa que as palabras clave, as variables, os nomes de función e calquera outro símbolo da linguaxe débense escribir de xeito consistente.

Neste punto, hai que ter en conta que XHTML non distingue entre maiúsculas e minúsculas. Polo que será unha boa norma empezar a escribir todas as propiedades e obxectos en "minúsculas".

- **Espazos en branco e saltos de liña.**

JavaScript ignora os espazos, as tabulacións e as novas liñas que aparecen entre os símbolos no programa. Isto facilita o formateo e sangrado do código dunha forma consistente para facilitar a súa lectura e comprensión.

- **Puntos e comas opcionais.**

En JavaScript a unha simple instrución séguelle, normalmente, un signo de punto e coma (;), igual que en C, C++ e Java. Sen embargo, en JavaScript podemos omitir dito signo se cada unha das nosas instrucións se coloca nunha liña independente (pero isto non é unha práctica de programación).

1.3.2 Comentarios

Calquera texto comprendido entre // e o final dunha liña trátase como comentario e JavaScript o ignora.

Calquera texto comprendido entre os caracteres /* e */ tamén se trata como comentario.

```
<script language="javascript">
// Isto é un comentario nunha liña
/* Isto é un comentario
    en varias liñas*/
</script>
```

1.3.3 Literais

Un **literal** é un valor dun dato que aparece directamente nun programa. Os seguintes son todos literais:

```
12                // O número doce
1.2              // O número un punto dous
"ola mundo"      // Unha cadea de texto
'ola'            // Outra cadea de texto
true             // Un valor booleano
false            // Outro valor booleano
/javascript/gi    // Un literal de "expresión regular"
null             // Ausencia dun obxecto
{ x:1, y:2 }      // Un inicializador de obxecto
[1,2,3,4,5]       // Un inicializador de matriz
```

1.3.4 Identificadores

Un **identificador** é simplemente un nome. Utilízanse para denominar variables e funcións e para proporcionar etiquetas para determinar bucles no código JavaScript.

As normas son as mesmas que en Java e outros moitos:

- O primeiro carácter debe ser unha letra, un signo de subliñado (_) ou un signo de dólar (\$).
- Os seguintes caracteres poden ser unha letra, un signo de subliñado ou un signo de dólar.

1.4 Palabras reservadas

Por palabras reservadas enténdense aquelas que non se poden empregar como identificadores.

- A seguinte táboa recolle as palabras clave estandarizadas:

break	do	if	switch	typeof
case	else	in	this	var
catch	false	instanceof	throw	void
continue	finally	new	true	while
default	for	null	try	with
delete	function	return		

- Esta outra táboa que presentamos a continuación recolle unha lista de palabras clave reservadas, estas non se utilizan actualmente pero resérvanse como posibles extensións futuras da linguaxe:

abstract	double	goto	native	static
boolean	enum	implements	package	super
byte	export	import	private	synchronized
char	extends	int	protected	throws
class	final	interface	public	transient
const	float	long	short	volatile
debugger				

- Tamén é recomendable non empregar: *as*, *is*, *namespace* e *use*.
- Tampouco se deben empregar as palabras que recolle a seguinte lista, estas son variables e funcións globais definidas polo estándar:

arguments	encodeURIComponent	Infinity	Object	String
Array	Error	isFinite	parseFloat	SyntaxError
Boolean	escape	isNaN	parseInt	TypeError
Date	eval	Math	RangeError	undefined
decodeURI	EvalError	NaN	ReferenceError	unescape
decodeURIComponent	Function	Number	RegExp	URIError

1.5 Tratamento de variables en JavaScript

Unha variable é o nome que se lle da a unha posición de memoria do equipo na que se almacena información.

1.5.1 Nomear variables

- O nome das variables en JavaScript debe estar formado por unha ou mais letras, díxitos e guións baixos.
- Non pode comezar por un número.

- JavaScript distingue entre maiúsculas e minúsculas.

1.5.2 Declarar de variables

- Para dicir a JavaScript que un **identificador** é unha variable, esta débese declarar primeiro.
- Para declarar variables en JavaScript utilízase a palabra reserva **var** seguida do nome da nova variable.
- Para declarar mais dunha variable na mesma sentencia utilízase a coma como separador.
- A inicialización das variables realízase empregando o operador de asignación (**=**).
- Pódese inicializar unha variable ao mesmo tempo que se declara ou en calquera outro punto despois no **script**.
- JavaScript lee o código de arriba cara abaixo realizando en cada paso a instrución lida. Ata que o programa inicializa unha variable esta permanece indefinida (*undefined*) e de ela non se pode extraer ningunha información.
- Pode declararse tamén unha variable simplemente iniciándoa e sen utilizar a palabra reserva *var*. É dicir, asignar un valor a unha nova variable sen declárala antes con *var*, fai que JavaScript a declare automaticamente.

1.5.3 Tipos de variables

- En JavaScript existen os seguintes tipos de datos válidos: *number*, *boolean*, *string*, *function* e *object*.
- Inicializar unha variable con **null** é unha boa forma de previr erros se non se está seguro de utilizar a variable.
- JavaScript é unha linguaxe de programación "debilmente tipado", polo que non se require que se definan os tipos de datos e non se impiden que se asignen diferentes tipos de datos a unha mesma variable.
- Vexamos un exemplo:

```
<script language="javascript" type="text/javascript">

</script>
```

1.5.4 Ámbito das variables

O ámbito dunha variable fai referencia á área ou áreas dentro do programa nas que a variable pode ser referenciada.

- **Variables globais:** As declaradas no interior dun *script* colocado na cabeceira ou no corpo dun documento.
- **Variables locais:** As variables declaradas no interior dunha función.

1.6 Operadores

Os operadores son símbolos ou identificadores que representan o modo que unha combinación de expresións se evalúan ou manipulan.

O operador que mais se emprega é o "**operador de asignación**" (**=**).

1.7 Funcións

Unha función é un *script* que se pode chamar polo seu nome en calquera momento.

Vexamos un exemplo:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<script language="javascript" type="text/javascript">

</script>

<title>Ola mundo en Javascript</title>
</head>
<body>
<script language="javascript" type="text/javascript">

</script>
</body>
</html>
```

--Vieites 10 feb 2009