

Guia definitiva de JavaScript

Sintaxis	Tipo de dato
<code>X = 1;</code>	Numérico entero
<code>X = 0.1;</code>	Numérico decimal
<code>X = "Hola Mundo";</code>	Strings o cadena de texto doble comillas
<code>X = 'Hola Mundo'</code>	Strings o cadena de texto comilla simple
<code>X = true;</code>	Valor lógico o boolean
<code>X = false;</code>	Valor lógico o boolean
<code>X = null;</code>	Valor nulo o sin valor
<code>X = undefined;</code>	Valor indefinido similar a nulo

Sintaxis	Descripción
<code>//</code>	Uso de doble diagonal para comentar
<code>;</code>	Uso de punto y coma para finalizar comando
<code>[]</code>	Uso de corchetes para declarar matrices (vector) o array
<code>{ }</code>	Uso de llaves para declarar bloques de comandos o propiedades
<code>+</code>	Uso de comando para sumar
<code>-</code>	Uso de comando para restar
<code>*</code>	Uso de comando para multiplicar
<code>/</code>	Uso de comando para dividir
<code>==</code>	Uso de comando para comparar igualdad
<code>!=</code>	Uso de comando para comparar desigualdades
<code><=</code>	Uso de comando para comparar valores menores o igual
<code><</code>	Uso de comando para comparar valores menores
<code>></code>	Uso de comando para comparar valores mayores
<code>>=</code>	Uso de comando para comparar valores mayores o igual
<code> </code>	Uso de comando símbolo lógico O comparar dos sentencias
<code>&&</code>	Uso de comando símbolo lógico Y comparar dos sentencias
<code>!</code>	Uso de comando símbolo lógico de negación
<code>var</code>	Las variables son declaradas con el comando var
<code>var X;</code>	Declarar el nombre de una variable
<code>X = 0;</code>	Declarar valor a la variable
<code>X++</code>	Declara incremento de variable
<code>X--</code>	Declara disminución de variable
<code>X += 1;</code>	Declara aumento con valor. similar a <code>X = X + 1;</code>
<code>X *= 1;</code>	Declarar aumento proporcional al valor, similar a <code>X = x * 3;</code>

Palabras reservadas para JavaScript				
break	delete	function	return	typeof
case	do	if	switch	var
catch	else	in	this	void
continue	false	instanceof	throw	while
debugger	finally	new	true	with
default	for	null	try	class
const	enum	export	extends	import
super	implements	private	public	yield
interface	package	protected	static	arguments
eval	int	synchronized	long	final

Librerías

Motor de juego

p5.js

<https://p5js.org/es/libraries/>

Sprite p5.js

<https://molleindustria.github.io/p5.play/docs/classes/Sprite.html>

Motor físico

Matter.js

<https://brm.io/matter-js/>

Motor grafico

Sweetalert

<https://sweetalert2.github.io/>
<https://sweetalert.js.org/guides/>

Activación de teclas

keyCode

<https://keycode.info/>

¿Cómo declarar los comandos de la sintaxis básica?

Funciones

```
function nombre_de_función (parámetro) {  
  //bloque de instrucciones  
}  
  
var nombre_de_variable = function (parámetro) {  
  //bloque de instrucciones  
}
```

Array o matrices (vectores)

```
var x = []; // Crear un vector (array)  
  
x [0]; // El primer elemento de un vector (matriz) o  
array  
  
x.length; //Todos los elementos del vector (matriz) o  
array  
  
x[x.length-1]; //El último elemento del vector (matriz) o  
array  
  
x.push(y); //Agregar elementos a los vectores (array)
```

Condicional

Sencilla

```
if (expresión) {  
  //bloque de instrucciones  
}
```

Con dos opciones

```
if (expresión1) {  
    //bloque de instrucciones  
} else {expresión2}
```

Con un ciclo (anidadas) de if

```
if (expresión1) {  
    if (expresión2) {  
        //bloque de instrucciones 2  
    }  
    //bloque de instrucciones 1  
} else {expresión3}
```

Con ciclos (anidadas) de if-else

```
if (expresión1) {  
    //Ejecute bloque de instrucción 1  
} else if (expresión2) {  
    //Ejecute bloque de instrucción 2  
} else if (expresión3) {  
    //Ejecute bloque de instrucción 3  
} else {  
    //Si todas son falsas, ejecuta 4  
}
```

Función de switch

```
var n;  
switch (n) {  
case 1: //Ejecuta bloque de comando del caso 1  
break;  
case 2: //Ejecuta bloque de comando del caso  
break;  
case...  
}
```

Función de bucle For

```
for (elemento inicial; elemento de comparación; elemento de incremento)  
{  
//bloque de instrucción  
}
```