Html: codificación ensamble de la página web, imágenes, videos, etc.…

CSS: dar una imagen y modelo de la página web

JS: Hacer la página dinámica

WA, web assembly, un nuevo modelo de programación destinada a agrupar los tres anteriores en uno

JS: Framewords

Crear proyectos eficientes, rápidos y completas, angolan, view,

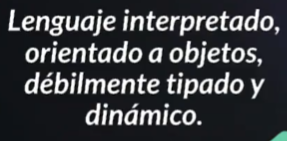
React native: aplicaciones nativas, Android, ios

Electron: desarrollo de apps para escritorio, ios o Android

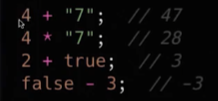
Back-end/IOT: Back end es la parte de programación de una app y el front end es la parte visual y de diseño de esta misma. NodeJS es un entorno de ejecución directamente relacionado con el back end.

¿Qué es JavaScript?

Es un lenguaje de programación que permite el dinamismo entre otras páginas web y en ella misma para crear una app web.

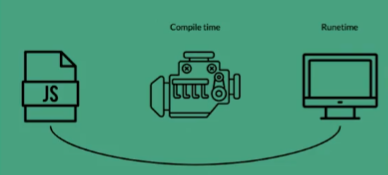


Débilmente tipado String Boolean (true=1 false=0)



Dinámico Runetime

Pasa directamente de un código de programación a un navegador para mostrar la página web. Lee línea por línea.

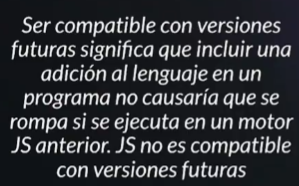


Estático

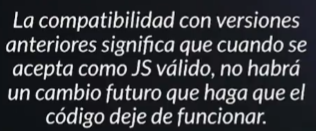
Compiladores, etapa en la cual se pasa un código “humano” a bycode y después al navegador mostrando la página web. V8

Bycode, código binario, lenguaje que entiende la máquina o navegadores

Forwards



JS es un Backwards Compatible, se hace estándar

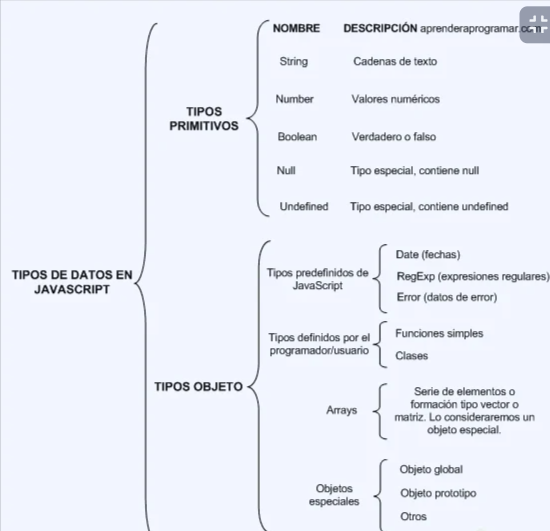
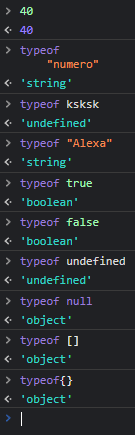


JS compilador BABEL, traduce una nueva versión de JS a una vieja y estándar

Elementos de un lenguaje de programación

Data que guardamos en una memoria, números, string, textos, palabras o frases …

Tareas son funciones que haremos con esa data.



¿Qué es una variable en Js?

Var es una palabra reservada que es una variable que se reserva y se guarda. En var se guarda datos para poder utilizarlos después.

// generar comentarios en el código //declarar //inicializar

¿Qué son las funciones en JS?

Son un conjunto de sentencias para generar acciones con las variables guardadas.

//declarativa

Function funciona como un var al guarda una función

//expresión



Cuando hablamos de funciones en JavaScript, tenemos dos tipos de funciones: Funciones Declarativas (function declaration / function statement) y Expresiones de función (function expression / funciones anónimas).

**Funciones Declarativas:**

En las funciones declarativas, utilizamos la palabra reservada function al inicio para poder declarar la función:

**function** **saludar**(nombre) {

console.log(`Hola ${nombre}`);

}

saludar('Diego');

#### Expresión de función:

En la expresión de función, la declaración se inicia con la palabra reservada var, donde se generará una variable que guardará una función anónima.

**var** nombre = **function**(nombre {

console.log(`Hola ${nombre}`)

}

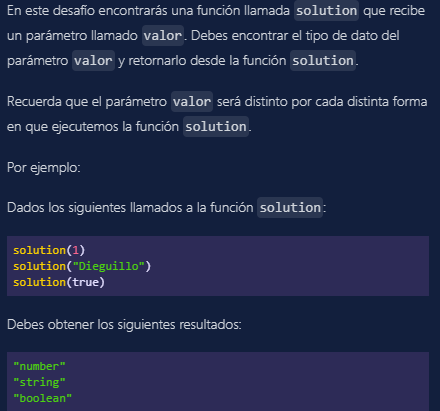
nombre(‘Diego’);

En la expresión de función, la función podría o no llevar nombre, aunque es más común que se hagan anónimas.

### Diferencias:

A las funciones declarativas se les aplica hoisting, y a la expresión de función, no. Ya que el hoisting solo se aplica en las palabras reservadas var y function.

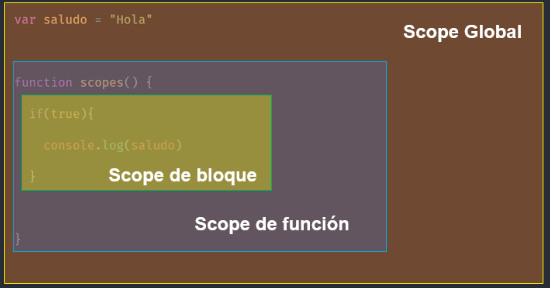
Lo que quiere decir que, con las funciones declarativas, podemos mandar llamar la función antes de que ésta sea declarada, y con la expresión de función, no, tendríamos que declararla primero, y después mandarla llamar.



Scope , es cada uno de los entornos donde las variables tienen alcance dentro del código de JS.

En este hay guardada muchas variables que js va buscando a medida que las necesites.

El scope local puede ser una función de un bloque, un bloque es toda porción de código que está encerrada en llaves {}, pueden ser: if, else, while y for,



El scope global son las variables globales que se encuentran declaradas fuera de los bloques de código o funciones es el entonco donde las variables pueden ser accedidas desde cualquier lugar de nuestro programa.

var miNombre = "Alexa"; function nombre () {var miApellido = "G"; console.log (miNombre + " " + miApellido);} nombre ();

Hoisting es cuando las variables y las funciones se manda a llamar antes de ser declararas.

Console.log es una función que nos ofrece el navegador nos ayuda a imprimir en la consola de Chrome los resultados.

Coerción

Implícitas es cuando el lenguaje nos ayuda y cambia de un tipo de valor a otro.

Explicitas es la forma en la obligamos a que un valor de un tipo cambia a otro tipo.

Consiste en transformar de un tipo de dato a otro de un variable.

Var … = String(…);

Number…

Valores truthy y falsy

El cero, null, NaN, undefined, false, (“”), seran un valor falso.

Un espacio, 1 en adelante, cualquier letra un [], {}, funtcion, true, serán valores true

Operaciones: Asignación, comparación y aritméticos

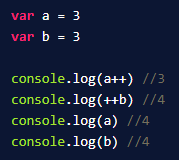
Var es asignación

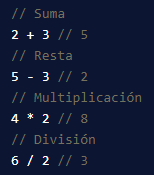
== comparación

=== se cerciora de que sea totalmente igual strin string o número número, tipo y valor sean los mismos.

&& si a es vdd o b es vdd se cumple // si a o b es vdd se cumple

++ a --a --a ++a



Condicionales: if, Else, else if

var edad = 18;

 if (edad === 18) {

    console.log ("Puedes votar, será tu primera votacion");

 } else if (edad > 18){

    console.log ("Puedes votar de nuevo");

 } else {console.log ("Aun no puedes votar");

}

condition ? true : false;

var número = 2;

var resultado = numero === 2 ? "Si soy un dos" : "No soy un dos";

Switch validar casos

var numero = 1;

switch(numero) {

    case 1:

      console.log ("Soy uno");

      break;

    case 2:

      console.log ("soy un dos");

      break;

    case 3:

      console.log ("Soy un tres");

      break;

    default:

       console.log ("No soy nada");

}

export function solution(article) {

  if (article == 'computadora') {

    return 'Con mi computadora puedo programar usando JavaScript';

  } else if (article == 'celular') {

    return 'En mi celular puedo aprender usando la app de Platzi'

  } else if (article == 'cable') {

    return '¡Hay un cable en mi bota!'

  } else  {

    return 'Artículo no encontrado'

  }

}

Array es una estructura de datos, tipo objetos, es un valor que guarda mas valores adentro, numero, objetos, string etc

[se guardan ciertas cosas]

Index (0,1,2,3)

Loops ciclos

For Es para crear una tarea que se reinicie sin necesidad de hacerlo manualmente

.i

While

Objetos

New: generar un nuevo objeto de otro objeto

Filter() crea un nuevo array con los elementos del array que se pasó como parámetro y que cumplan con la condición definida.

Map() crea un nuevo array con los resultados de la llamada a la función indicada aplicados a cada uno de sus elementos.

.find() encontrar algo de nuestro articulo

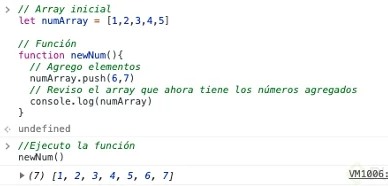
.forEach filtros para encontrar ciertos artículos

Some verdadero o falso para artículos que cumplan esa validación. Nos manda si es vdd que existe algún elemento que cumpla.

-var articulosbaratos = articulos.some(function(articulo){ return articulo.costo <= 700; });

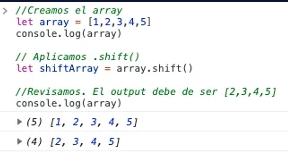
true-

El método .push nos permite agregar uno o más elementos al final de un array





El método .shift() es para eliminar el primer elemento de un array, el elemento 0.



El método .pop elimina el ultimo elemento de un array

