



REPORTE: TUTORIAL DE JAVASCRIPT

ESTUDIANTE: PÉREZ MELLADO JENNYFER

GRUPO: 211

MAESTRO: IVAN ISAY GUERRA LOPEZ

MATERIA: DESARROLLO DE APPS MÓVILES

FECHA: 08 - 09 - 2025

Puntos clave del video tutorial visto:

- Sobre Javascript:
 - Es un lenguaje de programación que funciona dentro del navegador.
 - Fue creado en el año 1995.
 - Se creo con la idea de extender la funcionalidad de la web más allá de páginas informativas, integrando interactividad, animaciones, etc.
 - Permite extender la funcionalidad y que el usuario sea capaz de interactuar con estos elementos.
 - Se encuentra en constante evolución
 - Soporta múltiples paradigmas.
- ¿Qué puedes hacer con Java script?
 - Se puede detectar a través de que dispositivo se esta conectando una persona, lo que permite que la pantalla se adapte a dichos requerimientos.
 - Se puede agregar retroalimentación después de realizar ciertas acciones para indicar si lo que esta haciendo es correcto o incorrecto y mostrar las opciones funcionalmente acertadas.
- Creación de código (Comandos y otras especificaciones):
 - console.log: Comando con el cual se muestra en la pantalla de la consola lo que le indiquemos al código, ya sea texto, números, etc; la palabra console, nos permite interactuar con la consola y el log, indica que se debe mostrar el siguiente texto que se escriba (En el caso del video, el típico "Hello World")
 - Para trabajar con Javascript dentro de nuestro programa Visual Studio Code, se crean dos archivos, el nombre no es relevante, sin embargo, uno debe terminar como ".html" indicando que es un archivo HTML y otro como ".js" indicando que es un archivo Javascript; esto es, un estándar.
 - <script src=""></script>: Este comando dentro del archivo HTML va a permitir que llamemos, de cierta forma al contenido o los elementos dentro del archivo Javascript, por lo que dentro de las comillas se debe poner el nombre del archivo .js, de esta forma, ambos archivos estarán vinculados para que JS actúe dentro de la página HTML.
 - documento.write: Para escribir texto en la página.
 - console.error: Para que se cree un error, se puede especificar un texto que indique la naturaleza de dicho error.
 - variable.length: Muestra el "tamaño", el número de elementos que tiene por ejemplo un array.
- Tipos de datos en Javascript:

- String: Una cadena de texto que puede generarse al escribir un conjunto de caracteres entre comillas simples ' ' o dobles " ".
- Números: Para el manejo de variables numéricas del tipo entero, decimal y/o negativos.
- Lógicos: Una variable la cual solo tiene dos valores true = verdadero y false = falso.
- Array: Listas que pueden almacenar un conjunto de información/variables, se usa mediante corchetes y comas [,]; si quieres mostrar o usar un elemento específico de un array, usas el nombre del array y entre corchetes su subíndice, cada elemento de este, será número del 0 hasta donde llegue el array, por lo que en un array con 7 elementos, el primero está en [0] y el último está en [6],
- Object: Representación de un objeto en la vida real, una variable con características varias que pertenecen a una misma entidad, se utiliza mediante llaves, de la siguiente forma


```
{
        "nombre característica": "Característica,
        "Color de cabello": "Negro"
      }
```

➤ Manejo de variables:

- Para guardar información en variables, de forma inicial, basta con colocar el nombre de la variable seguido de un = y la información que tendrá; de esta forma indicamos que el valor de esa variable es por ejemplo una palabra "" o un número #.
- var: Es una palabra clave que le indica al programa que estamos creando una variable.
- let: Otra palabra clave parecida a var que se usa para crear variables.
- Al usar variables, en cualquier momento dentro del código su valor puede cambiar, esto solo usando nuevamente el nombre de la variable, y delante del igual otorgándole un nuevo valor.
- const: Palabra clave contraria, en esta se indica que nuestro dato será constante y por lo tanto una vez dado su valor inicial este no puede ser modificado en otros momentos del código.
- El nombre de una variable no puede iniciar con un número o un carácter distinto a una letra.

➤ Operadores:

- Para procesar con operaciones simples los datos dentro del programa, se utilizan los símbolos de "+, -, *, /", para sumar, restar, multiplicar y dividir respectivamente, claro, estos símbolos se utilizan en medio de los datos que quieren ser procesados, por ejemplo 2 + 3, o, 8 - 3 * 4.

- El operador de suma (+), también sirve en datos no numéricos, es decir datos con caracteres, su forma es la misma, pero en ese caso ya no se le llama sumar, sino que se le dice concatenar.
- De esta forma, también se pueden comparar datos mediante el uso de los símbolos "> y <", que indican o en este caso preguntan si tal dato es mayor o menor al siguiente, por lo que el resultado de dicha operación será un booleano con el valor de true o false; así mismo se puede preguntar si son iguales, si tienen el mismo valor utilizando "=", al igual que se puede saber si son distintos utilizando ahora este símbolo "!="; por ultimo también se puede utilizar una combinación de estos símbolos, particularmente "< y >" y "=", de la forma ">=" o "<=", siendo esto un mayor/menor igual que...?.
- Control de flujo:
 - Como el programa va a comportarse a partir de un resultado
 - if: Si una condición se cumple entonces se realizarán las acciones dentro del bloque de if.
 - else: Va seguido de "if", y si la condición no se cumple, se mostrarán las acciones dentro de este bloque en lugar de las acciones dentro de "if".
 - else if: En caso de tener mas de dos condiciones que es posible, se cumplan en el mismo proceso, mediante el "else if", solo se realizaran las acciones de uno de los bloques, ya sea "if" o "else if".
 - switch: A partir de una variable especifica, se realizan acciones especificas dependiendo el valor de este, es decir, si se tiene una variable numérica y queremos que el programa diga el número en letras, podemos tener algo como esto:


```
switch(numero(nombre de la variable)) {
  case 1(valor posible variable):
    console.log("uno");
    break;
  case 2:
    console.log("dos");
    break;
  default:
    console.log("Número desconocido");
    break;
```

De esta forma dependiendo el valor de la variable el programa mostrara una cosa u otra, en caso de que el valor sea 1, en la consola veremos escrito "uno"; la palabra "break", le indica al programa de que si se cumplió el caso 1, entonces no siga con los demás; por otro lado "default", es un valor predeterminado, en caso de que el valor especifico que tomo

la variable no se encuentre dentro de los casos previamente hechos, entonces se realizaran las acciones dentro del bloque de “default”, por ejemplo, si nuestra variable vale 3, el programa mostrara “Número desconocido”.

➤ Bucles:

- Ciclos o repeticiones de acciones por un número determinado de veces dado por una condición.
- while: Mientras cierta condición se cumpla, pasara lo siguiente..., es un ciclo que, al tener una condición inicial, es posible que jamás se ejecute dentro del programa.
- variable++ o variable--, mientras se ejecute un ciclo, una variable sumara o restara su valor en 1, cada vez que se ejecute.
- for: Se forma mediante tres partes, el valor inicial de una variable, la condición que mientras se cumpla se ejecutará, y la forma en que la variable cambia mediante alguna operación, por lo tanto, for será puede ejecutado un número determinado de veces y al menos una vez.

➤ Funciones:

- Bloques de código separados del principal que realizaran ciertas acciones, se manda a llamar dentro del código principal y hasta ese momento se ejecutaran en el programa, al contrario si nunca son llamadas, no serán ejecutadas, también pueden usarse varias veces en distintos segmentos del código.
- Para crear una función se usa la palabra “function”, se le da un nombre y seguido de esto unos paréntesis en los cuales podemos especificar datos o variables que el código principal debe otorgarle a la función, esto en caso de ser necesario, de lo contrario los paréntesis se dejan vacíos.
- Para que la función sea llamada y ejecutada en el programa, en el código principal se escribe el nombre de la función acompañado de sus paréntesis y entonces la función se ejecutara dentro del programa, de lo contrario no aparecerá en ningún momento.
- En caso de que la función necesite un dato, además de especificarlo en el bloque de la función en los paréntesis, en el momento de llamar a la función en el bloque principal, además del nombre de la función, dentro de los paréntesis, debemos darle entonces el dato que necesita, ya sea un número, un texto o una variable, etc.

Este tutorial me ayudo a repasar conceptos y herramientas que ya conocía, junto a otras que no tanto, además de enseñarme de forma más específica como funcionan o por decirlo de alguna forma, como se llaman estas herramientas en el lenguaje de Javascript.