**Sesión 11: Condicionales, bucles y métodos con JavaScript**

* 1. **Condicionales**

Las «instrucciones condicionales» se usan para realizar las diferentes acciones según una condición.

En JavaScript, podemos encontrar las siguientes declaraciones condicionales:

| Instrucción | Descripción | Sintaxis |
| --- | --- | --- |
| *if* | Se usa para ejecutar un bloque código si la condición es verdadera. | if (condición) {  bloque de código a ejecutar  si la condición es verdadera  } |
| *if…else* | Se usa para ejecutar un bloque de código si la condición es verdadera u otro bloque de código si la condición es falsa. | if(condición){  bloque de código a ejecutar  si la condición    es verdadera  }  else {  bloque de código a ejecutar  si la condición    es falsa  } |
| *if…else if…else* | Se usa para seleccionar uno de los muchos bloques de código a ejecutar. | if(condición1){     bloque de código a ejecutar     si la condición1 es verdadera  }  else if(condición2){     bloque de código a ejecutar     si la condición2 es verdadera  }  else {     bloque de código a ejecutar     si condición1 y condición2     no son verdaderas.  } |
| *Switch* | Se usa para seleccionar uno de los varios bloques de código a ejecutar (igual). | switch([evaluación | valor]) {     case 1:       ejecuta este bloque       si el valor o la evaluación       son iguales a 1.       break;     case 2:       ejecuta este bloque       si el valor o la evaluación       son iguales a 2.       break;     default:       ejecuta este bloque si no ingresa a ninguno       de los "case" anteriores.  } |

* **Estructuras condicionales**

Comparan una variable contra otro(s) valor(es), para que, en base al resultado de esta comparación, se siga un curso de acción dentro del programa. Cabe mencionar que la comparación se puede hacer contra otra variable o contra una constante, según se necesite. Existen dos tipos básicos, las simples y las múltiples.

|  | Simples | Dobles | Múltiples |
| --- | --- | --- | --- |
| Definición | También conocidos como “Tomas de decisión”. | Permiten elegir entre dos opciones o alternativas posibles en función del cumplimiento o no de una determinada condición | Son tomas de decisión especializada que permiten comparar una variable, contra distintos posibles resultados, ejecutando para cada caso una serie de instrucciones específicas |
| Sintaxis | Si <condición> entonces  Acción(es)  Fin si | Si <condición> entonces         Acción(es)      Sino         Acción(es)  Fin si | Si <condición> entonces         Acción(es)        Sino         Si <condición> entonces                     Acción(es)               Sino                .                .  Varias condiciones  Fin si  Fin si |

* 1. **Bucles**

Los bucles ofrecen una forma rápida y sencilla de hacer algo repetidamente; es decir, repiten una acción varias veces.

En esta sesión, revisaremos las declaraciones for y While.

* 1. **Ciclo FOR**

Un ciclo for se repite hasta que una condición especificada se evalúe como false. El bucle for de JavaScript es similar al bucle for de Java y C.

La declaración for tiene el siguiente aspecto:

for ([expresiónInicial]; [expresiónCondicional]; [expresiónDeActualización])

instrucción

* 1. **Ciclo While**

Una declaración while ejecuta sus instrucciones siempre que una condición especificada se evalúe como true. Una instrucción while tiene el siguiente aspecto:

while (condición)

expresión

Si la condición se vuelve false, la instrucción dentro del bucle se deja de ejecutar y el control pasa a la instrucción que sigue al bucle.

* 1. **Métodos**

Se trata de una porción de código que permite realizar tareas asociadas a un objeto, básicamente un método es una función que pertenece a un objeto o clase. Los métodos permiten brindar a los objetos funcionalidades y características comunes a su clase.

Todos los métodos son funciones, es decir, retornan un dato. Aunque si definimos un método que retorna un tipo “void”, significa que no retorna nada. Ej.: void imprimir()

Por ejemplo: cuando definimos métodos para un objeto “pared”

public void AgregarPuerta() {

this.puertas++;

}

public void QuitarPuerta() {

this.puertas--;

}

public void PintarColor() {

this.color= "Azul";

}

Void: significa que la función no devuelve nada.  
Public: indica que se puede invocar desde fuera de la clase.

* 1. **. Recursos complementarios**
* **Tutorial Javascript - Condicionales: If, Else if**

En este video conocerás sobre las condicionales y cómo usarlas.

**Fuente:**

Codynamico. (16 de diciembre de 2019). *Tutorial Javascript - Condicionales: If, Else if.* [Archivo de video]. Youtube. <https://youtu.be/54Oi-HaMZ7E>

* **¿QUÉ SON los CICLOS o BUCLES?**

En este video conocerás sobre los bucles y cómo usarlos.

**Fuente:**

Tecnología Binaria. (7 de noviembre de 2014). ¿*QUÉ SON los CICLOS o BUCLES? | Iniciándose en la Programación #10* [Archivo de video]. Youtube. <https://youtu.be/yNvkL7L0_Vo>

* **Métodos en Java**

En este video se explica en qué consisten los métodos Java.

**Fuente:**

edu4java. (24 de setiembre de 2014). *Métodos en Java* [Archivo de video]. Youtube. <https://www.youtube.com/watch?v=aOVEtIxIPVw>

**Bibliografía**

Hunter, A. (20 de mayo de 2020). *La Iteración del JavaScript Moderno*. <https://www.linkedin.com/pulse/la-iteraci%C3%B3n-del-javascript-moderno-alvison-hunter/?originalSubdomain=es>