|  |  |  |  |
| --- | --- | --- | --- |
| JAVASCRIPT | | PHP | |
| INICIO | | | |
| (En HTML) => <script> -código- </script> | | Tanto en HTML como en PHP => **<?php** -código- **?>**  **<? echo** -código- **?> 🡺 <?=** -código- **?>** | |
| ACCESO | | | |
| <script src=”ruta/archivo.js”></script> | | <form **action=”ruta/archivo.php”**></form> | |
| DECLARACIÓN DE VARIABLES | | | |
| var / let variable = valor; | | **$**variable = valor; | |
| TIPO DE VARIABLE | | | |
| typeof(variable); | | **gettype($variable);** | |
| ARRAYS | | | |
| let nombre\_array = [val1, val2, val3, ….]; | | $nombre\_array **= array(**val1, val2, val3, ….**);** | |
| let primerValor = nombre\_array[0]; // = “val1”; | | $primerValor = $nombre\_array**[0];** // = “val1”; | |
| DICCIONARIOS | | | |
| let diccionario = { “clave1”:valor1, “clave2”:valor2,…}; | | $diccionario = **array( “clave1”=>valor1, “clave2”=>valor2,…);** | |
| BUCLES | | | |
| for (let index = 0; index < valorMáximo; index++){ -código- } | | for ($index = 0; $index < valorMáximo; $index++){ -código- } | |
| for (let index = valorMáximo; index > 0; index--){ -código- } | | for ($index = valorMáximo; $index > 0; $index --){ -código- } | |
| array.foreach( (elemento) => { -código- }); | | **foreach** ($array **as** $valor) { -código- } | |
| while (condición) { -código- } | | = | |
| let cadena = “\*\*\*\*\*\*\*”; | **MÉTODOS PARA STRINGS** | | **$cadena = “\*\*\*\*\*\*\*”;** |
| let longitud = cadena.length; | | $longitud = **strlen**($cadena); | |
| let aMinúsculas = cadena.toLowerCase(); | | $aMinusculas = **strtolower**($cadena); | |
| let aMayusculas = cadena.toUpperCase(); | | $aMayusculas = **strtoupper**($cadena); | |
| let letra = cadena.charAt(indice); | | $letra = **$cadena[indice]**; | |
| let arrayPalabras/Letras = cadena.split(separador); | | $arrayPalabras = **explode(separador, $cadena);**  $arrayLetras = **str\_split($cadena);** | |
| let porcion = cadena.substring(posición inicio, posición final); | | $porcion = **substr($cadena, posición inicio, posición final);** | |
| let nuevaCadena = cadena.replace(expresión, nuevoValor); | | $nuevaCadena **=** **str\_replace($cadena**, **expresión, nuevoValor)**; | |
| let array = [val1, val2, val3]; | **MÉTODOS PARA ARRAYS** | | **$array = [val1, val2, val3];** |
| let longitud = array.length; | | $longitud = **count**($array**);** | |
| let creaCadena = array.join(separador); | | $creaCadena = **implode(separador,** $array**);** | |
| let eliminaUltimoElemento = array.pop(); | | $eliminaUltimoElemento = **array\_pop(**$array**);** | |
| let eliminaPrimerElemento = array.shift(); | | $eliminaPrimerElemento = **array\_shift(**$array**);** | |
| array.push(nuevoElemento); // Añade elemento al final | | **array\_push(**$array**, elemento);** | |
| array.unshift(nuevoElemento); // Añade elemento al principio | | **array\_unshift(**$array**, elemento);** | |
| array.reverse(); // Coloca los elementos al inverso | | $arrayInvertido = **array\_reverse(**$array**)**; | |
| FUNCIONES | | | |
| function miFuncion( parametro1, parametro2, … ) { -codigo- } | | | |

|  |  |
| --- | --- |
| JAVASCRIPT - DOM | |
| OBTENER ELEMENTOS | |
| document | Objeto javascript que hace referencia al contenido total de nuestra página. |
| document.getElementsByTagName(nombreEtiqueta) | Obtiene todos los elementos de la página por etiqueta/tag. |
| document.getElementsByName(nombreAtributo) | Obtiene todos los elementos de la página por el atributo NAME. |
| document.getElementById(nombreAtributo) | Obtiene todos los elementos de la página por el atributo ID. |
| ACCESO A ATRIBUTOS Y ESTILOS | |
| <a id="enlace" href="http://www...com">Enlace</a>  let enlace = document.getElementById("enlace");  alert(enlace.href); *// muestra http://www...com* | <img id="imagen" **style="margin:0;** border:0;" src="logo.png" />  let imagen = document.getElementById("imagen");  alert(**imagen.style.margin**); |
| PÁGINA | |
| window.location *= “http://www.paginanueva.com”;* | Cambia la dirección de la página (en la misma ventana). |
| window.open*(“pagina.html”,”nombrepagina”,”width=500,height=200”);* | Abre la dirección en una nueva ventana. |
| CREAR ELEMENTOS | |
| let parrafo = document.createElement("p"); | Creación de un **nodo de tipo** **Element/Tag** que represente al elemento. |
| let atributo = document.createAttribute("id"); | Creación de un **nodo de** **tipo Atributo.** |
| atributo.value = “nombre\_id”; | Se define el contenido del atributo. |
| let contenido = document.createTextNode("Hola Mundo!"); | Creación de un **nodo de** **tipo Texto** que represente el contenido del elemento. |
| parrafo.appendChild(contenido); | Añadir el nodo Text como nodo hijo del nodo Element. |
| document.body.appendChild(parrafo); | Añadir el nodo Element a la página, en forma de nodo hijo del nodo correspondiente al lugar en el que se quiere insertar el elemento. |
| ELIMINACIÓN DE ELEMENTOS | |
| var parrafo = document.getElementById("provisional");  parrafo.parentNode.removeChild(parrafo);  <p id="provisional">...</p> | La forma más segura y rápida de acceder al nodo padre de un elemento es mediante la propiedad **nodoHijo.parentNode**. |

|  |  |
| --- | --- |
| JAVASCRIPT - EVENTOS | |
| ELEMENTOS | |
| ONBLUR | El elemento pierde el focus / la selección. |
| ONFOCUS | El elemento recibe el focus / es seleccionado. |
| ONCHANGE | Cuando el elemento detecta un cambio. |
| ONCLICK | Clicar con el ratón. |
| ONDBLCLICK | Clicar dos veces con el ratón. |
| ONSELECT | Al seleccionar un texto. |
| TECLAS | |
| ONKEYDOWN | Mantener pulsada una tecla. |
| ONKEYPRESS | Pulsar una tecla. |
| ONKEYUP | Soltar una tecla pulsada. |
| PÁGINA | |
| ONLOAD | La página se ha cargado completamente. |
| ONUNLOAD | Al abandonar la página / cerrar el navegador. |
| ONRESIZE | Al modificar el tamaño del navegador. |
| RATÓN / MOUSE | |
| ONMOUSEDOWN | Mantener pulsado un botón del ratón. |
| ONMOUSEUP | Al soltar el botón pulsado del ratón. |
| ONMOUSEMOVE | Al mover el ratón. |
| ONMOUSEOUT | El ratón sale del elemento. |
| ONMOUSEOVER | El ratón pasa por encima de un elemento. |
| FORMULARIO | |
| ONRESET | Borra los datos del formulario. |
| ONSUBMIT | Cuando se envía el formulario. |

|  |
| --- |
| **JAVASCRIPT** |
| **THIS => Hace referencia al elemento que ha provocado el evento.**  <div id="contenidos" style="width:150px; height:60px; border:thin solid silver"  onmouseover="document.getElementById('contenidos').style.borderColor='black';"  onmouseout="document.getElementById('contenidos').style.borderColor='silver';>  Sección de contenidos...  </div>  <div id="contenidos" style="width:150px; height:60px; border:thin solid silver"  onmouseover="this.style.borderColor='black';"  onmouseout="this.style.borderColor='silver';">  Sección de contenidos...  </div>  **En este caso THIS (este mismo elemento) hace referencia al propio div con id=”contenidos”, que es quien ejecuta los eventos onmouseover y onmouseout.**  **Quiere decir que al propio div se le cambiara el color de borde cuando pase por encima el ratón y cuando deje de pasar por encima.** |

|  |  |  |  |
| --- | --- | --- | --- |
| JAVASCRIPT - FORMULARIOS | | | |
| ATRIBUTOS | | | |
| NAME | | Nombre del elemento. (Read Only) | |
| ID | | Identificador del elemento. | |
| TYPE | | Indica el tipo de elemento. | |
| VALUE | | Indica el valor que tiene el elemento. | |
| TIPOS - <INPUT> | | | |
| TEXT | Tipo texto. | | |
| NUMBER | Tipo número. | | |
| EMAIL | Tipo email. Verifica que el formato esté bien escrito. | | |
| PASSWORD | Tipo contraseña. Enmascara el texto introducido con símbolos. | | |
| RADIO | Tipo “selecciona una opción”. | | |
| CHECKBOX | Tipo “marca la casilla”. | | |
| URL | Tipo URL. | | |
| HIDDEN | Esconde el campo, pero sigue estando en el formulario. | | |
| FILE 🡺ACCEPT=”.jpg, .png, ….” | Tipo archivo. Verifica que el formato esté bien escrito. | | |
| BUTTON | Tipo botón. Puedes darle una función con el evento onclick. | | |
| RESET | Limpia el formulario. | | |
| SUBMIT | Envía el formulario. | | |
| RESTRICCIONES | | | |
| REQUIRED | | | Indica que el elemento no debe estar vacío. |
| MIN / MAX | | | Indica el número mínimo y máximo que se puede introducir. |
| PATTERN | | | Indica el patrón que se debe seguir para validar el campo. |
| MINLENGTH / MAXLENGTH | | | Indica la longitud mínima y máxima del texto introducido. |
| CHECKVALIDITY() | | | Valida si el valor introducido cumple con las restricciones. |
| SETCUSTOMVALIDITY(“mensaje”) | | | Mensaje personalizado que se mostrará en el elemento a validar. |
| let validación = elemento.VALIDTY | | | Retorna el estado de validación del elemento. |
| validación.valueMissing | | | Retorna TRUE si no hay valor. (REQUIRED) |
| validación.patternMismatch | | | Retorna TRUE si el valor no coincide con el patrón. |
| validación.tooLong / tooShort | | | Retorna TRUE si el valor es más largo/corto que máximo/mínimo especificado. (MAXLENGTH / MINLENGTH) |
| validación.rangeOverflow / rangeUnderFlow | | | Retorna TRUE si el valor es mayor/menor al máximo/mínimo especificado. (MAX/MIN) |
| validación.typeMismatch | | | Retorna TRUE si el valor no coincide con la sintaxis del tipo especificado. (TYPE = “EMAIL/URL”) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OBTENER VALORES | | | | |
| TEXTO / TEXTAREA | let valor = document.**getElementById**(“texto”)**.value;** | | | |
| RADIOBUTTON | let elementos = document.**getElementsByName**(“name”);  for (var i=0; i < elementos.length; i++) {  let valor = **elementos[i].value**;  let seleccionado = **elementos[i].checked**;  } | | | |
| CHECKBOX | let valor = document.getElementById(“id”)**.value**;  let seleccionado = document.getElementById(“id”)**.checked**; | | | |
| SELECT | let lista = document.**getElementById("idSelect");**  letindiceSeleccionado = lista.**selectedIndex**;  let opcionSeleccionada = lista.**options[indiceSeleccionado]**;  let textoSeleccionado = opcionSeleccionada.**text**;  let valorSeleccionado = opcionSeleccionada.**value**; | | | |
| Validaciones personalizadas: | | | | |
| <form ONSUBMIT=”RETURN VALIDACION()”> | | | Se utiliza una función para validar todos los campos. | |
| Si se cumplen las siguientes validaciones se retornará FALSE: | | | | |
| let valor = document.getElementById("campo").value; | | | | |
| valor.length == 0 (texto) | | Obliga a que el texto introducido no esté vacío. | | |
| isNaN(valor) (número) | | Obliga al usuario a introducir un valor numérico. | | |
| valor.selectedIndex == 0 (select) | | Obliga a seleccionar una opción de una lista. | | |
| let elemento = document.getElementById("campo"); | | | | |
| elemento.checked | | Verifica si el checkbox ha sido seleccionado. | | |
| EXPRESIONES REGULARES  Se validan mediante el método “test()” 🡪 expresionRegular.test(valor) | | | | |
| /^\w+([.-\_+]?\w+)\*@\w+([.-]?\w+)\*(\.\w{2,10})+$/ | | | | Valida un email. |
| /^[a-zA-ZáéíóúÁÉÍÓÚüÜñÑ'-]+\s?[a-zA-ZáéíóúÁÉÍÓÚüÜñÑ'-]\*$/ | | | | Valida un nombre. |
| /^(\d{8}[A-HJ-NP-TV-Z]|[XYZ]\d{7}[A-Z])$/; | | | | Valida tanto DNI como NIE. |
| /^(0[1-9]|[1-2]\d|3[01])(\/)(0[1-9]|1[012])\2(\d{4})$/; | | | | Valida fecha. |
| /^(\+\d{1,3}\s?)?(\d{3,4}[-\s]?){2}\d{3,4}$/ | | | | Valida teléfono. |

|  |
| --- |
| **JAVASCRIPT – TRY / CATCH** |
| Try {  Aquí intenta ejecutar el código.  ej: conectar a una base de datos / abrir un archivo / …  } Catch (error) {  Aquí intercepta el error.  Se puede retornar el mensaje de error que se quiere mostrar o ejecutar otro código.  } Finally {  Aquí siempre se ejecutará un código al finalizar.  } |

|  |  |  |
| --- | --- | --- |
| **PHP** | | |
| **echo / print** | Para imprimir datos por pantalla. | |
| Incluir un archivo externo a la web: | | |
| <?php **include(**“archivo.html”**)**; | Produce un **Warning** si no encuentra el archivo. | |
| <?php **require(**“archivo.html”**)**; | Produce un **Error** si no encuentra el archivo. | |
| **FORMULARIOS <form>** | | |
| **ACTION** | Acceder al archivo .php. | |
| **METHOD** | Método de envío de datos. | |
| **FORMULARIOS - METHOD** | | |
| **GET** | Envía los datos mostrándolos en la URL.  http://localhost/tutorial/get.php**?name=Mac&age=21** | |
| **POST** | Envía los datos SIN mostrarlos en la URL. | |
| **RECOGER DATOS** (se utiliza el atributo NAME)  <INPUT TYPE="SUBMIT" **NAME="enviar“** VALUE="procesar"> | | |
| $nombre = **$\_GET[“enviar”];** | Si el form es METHOD=”GET”. | |
| $nombre = **$\_POST[“enviar”];** | Si el form es METHOD=”POST”. | |
|  | | |
| **CAMPOS CHECKBOX**  <INPUT **TYPE="checkbox"** **NAME="extras[]"** VALUE="garaje" CHECKED>Garaje  <INPUT **TYPE="checkbox"** **NAME="extras[]"** VALUE="piscina">Piscina  <INPUT **TYPE="checkbox"** **NAME="extras[]"** VALUE="jardin">Jardín | | $extras = **$\_POST[‘extras’]**;  foreach**($extras as $extra)** {  print(“$extra<BR>\n”) }; |
| **SESIONES** | | |
| **session\_start();** | Inicializa una sesión. | |
| **session\_register(variable); $\_SESSION[‘nombre’] = valor;** | Registra / almacena una variable en la sesión. | |
| **session\_unregister(variable); unset( $\_SESSION[‘nombre’]);** | Elimina una variable de la sesión. | |
| **session\_is\_registered(variable); If(isset($\_SESSION[‘nombre’]))** | Comprueba si una variable está registrada. | |
| **session\_destroy();** | Cierra una sesión. | |
| **COOKIES** | | |
| **$cookie\_name** = "remember\_me"; | Nombre de la cookie. | |
| **$cookie\_value** = 1; | Valor de la cookie. | |
| **$cookie\_expire\_time** = time() + (24\*3600); | Tiempo de expiración de la cookie. (ej: 1 dia) | |
| **setcookie($cookie\_name, $cookie\_value, $cookie\_expire\_time);** | Crear la cookie. | |

|  |  |
| --- | --- |
| **FETCH**  **(Javascript: async - await)** | |
| **async** function serverFetch(){} | Se crea la función de tipo async. |
| const respuesta = **await fetch(url);** | Se hace la petición al servidor. |
| **If (!result.ok) throw** **{**ok:false, msg:”mensaje de error”**};** | Pregunta si ha salido algo mal, si es así lanza un error. |
| let data = **await respuesta.json();** | Convierte el resultado en json. |
| **FETCH**  **(PHP)** | |
| **header(“Content-Type: application/json”);** | Dice que este archivo devuelve un archivo de tipo JSON. |
| **echo json\_encode(**$variable**);** | Variable que retorna al fetch en formato JSON.  La $variable puede ser: array, string, int… |