**INFORME DE PRÁCTICA O TALLER N° \_\_\_**

|  |  |
| --- | --- |
| DATOS DE LA ASIGNATURA | |
| Nombre del estudiante / Integrantes:  Morales Cali Jhonatan Alexander  Mendez Minda Pablo  Vasquez Escobar Diego Mauricio  Carrera: Redes y telecomunicaciones  Docente:  Ing. Rosero Gómez Julio David  Asignatura: Desarrollo de aplicaciones de internet | **Nivel: Tercero**  **Fecha: 12 de julio de 2023**  **Calificación:**  **Periodo Académico: Abril 2023 – Septiembre 2023** |

**1. TITULO:**

Diseño y desarrollo de las páginas “Nosotros”, “Productos” y “Contactos” de un sitio web de tipo empresarial.

**2. OBJETIVO ALCANZADO:**

Realizar un sitio web que complemente la ventana “Home”, incluya las ventanas “Nosotros”, “Productos” y “Contactos”. El contenido de cada ventana debe estar acorde a la descripción de la página. La ventana “Nosotros” indicará a que se dedica la empresa. La ventana “Productos” contendrá los artículos que ofrece a la venta la institución. La venta “Contactos” contendrá la información de contacto como mapa del sitio, números de contacto, redes sociales, email, etc.

**3. DESCRIPCIÓN DE LAS ACTIVIDADES DESARROLLADAS DURANTE LA PRACTICA**

|  |  |
| --- | --- |
| ACTIVIDAD | ENTREGABLES DEL ESTUDIANTE |
| Desarrollo del tema de estudio mismo que proviene de la selección de una idea que tiene la empresa. | Informe académico-investigativo acerca de los elementos de un plan de investigación de la empresa Alfa S.A d |
| Crear una aplicación web que contenga una página de “Home” de una empresa dedicada a la venta de artículos tecnológicos. | Código de programación visual studio |
| La página de “Home” debe mostrar la información general de la empresa, use estilos para personalizar la página web. | Código de programación visual studio |
| Para la implementación de estilos implemente los frameworks Bootstrap y Tailwind, en conjunto con una hoja de estilos CSS | Código de programación visual studio |
| Para la visualización de la aplicación en un ambiente de producción implemente XAMPP como servidor web. | Código de programación visual studio |
| Realice constantes revisiones a la vista de la aplicación para mantener la armonía y una correcta visualización de los componentes usados. | Código de programación visual studio |

**4. MARCO TEÓRICO:**

Establecer teorías o conceptos Generales descritos durante la práctica como parte del proceso de aprendizaje.

**5. MATERIALES / EQUIPOS**

|  |  |  |
| --- | --- | --- |
| NRO | Nombre de Equipo / Material | Uso que se le dio en la práctica |
| 1 | Computador personal | Para realizar todo el taller |
| 2 | Visual studio code | Programa para la elaboración del Código |
| 3 | internet | Para ingresar a la página de W3shool |
| 4 | boostrap | Código de carrusel |

**6. CONCLUSIONES Y RECOMENDACIONES**

|  |  |
| --- | --- |
| **CONCLUSIONES** | **RECOMENDACIONES** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**8. EVALUACIÓN REFLEXIVA DE LA PRÁCTICA (Estudiante)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ESCALA DE EVALUACIÓN** | | | | | | | | | |
| **1** | **2** | | **3** | **4** | | | **5** | | |
| **Nunca** | **Pocas veces** | | **Algunas veces** | **Casi siempre** | | | **Siempre** | | |
| **DESCRIPCIÓN DE CRITERIOS A EVALUAR7** | | | | **Escala de evaluación8** | | | | | |
| **1** | **2** | **3** | | **4** | **5** |
| ¿Considera que la práctica le ayudó en su formación integral (actitudes, conocimientos y habilidades)? | | | |  |  |  | |  |  |
| ¿Lo aprendido en su proceso de formación profesional le sirvió para el desempeño de su Práctica? | | | |  |  |  | |  |  |
| ¿Considera que las actividades que realizó fueron las pertinentes para el cumplimiento del objetivo de su práctica? | | | |  |  |  | |  |  |
| ¿La realización de la práctica cumplió con sus expectativas personales y profesionales? | | | |  |  |  | |  |  |
| **DE MANERA BREVE, CLARA Y CONCISA, PRESENTE SUS CONSIDERACIONES SOBRE9:** | | | | | | | | | |
| Dificultades observadas en el desarrollo de su práctica | |  | | | | | | | |
| Logros obtenidos gracias a su práctica, aprovechables para su futura vida profesional | |  | | | | | | | |
| Recomendaciones para el desarrollo de módulos | |  | | | | | | | |
| **OBSERVACIONES FINALES:** | | | | | | | | | |
| **Firma del estudiante** | | | | | | | | | |

**9. EVIDENCIAS FOTOGRAFICAS / ANEXOS:**

***Grupo de personas en una playa

Descripción generada automáticamenteUn grupo de personas en el mar

Descripción generada automáticamente con confianza media***

**Link:** <https://www.youtube.com/watch?v=sEYjy3EbGbU&t=4s>

|  |  |
| --- | --- |
| REVISIÓN | APROBACIÓN |
| Firma:……………………………………….………  Fecha:…    Docente Carrera | **Firma:……………………………………….………**  Fecha:…    **Coordinador** |
|
|
|
|
|
|
|
|
|

**1. ¿Cuáles son los tipos de datos JavaScript?**

Sin duda alguna, los datos son uno de los pilares fundamentales dentro de los lenguajes de programación, ya que son ellos quienes transitan por el paradigma «entrada, proceso y salida» teniendo así datos de entrada - procesamiento de los datos - datos de salida. Es por ello que cada lenguaje de programación tiene estructuras o tipos de datos integrados que permiten almacenar y abstraer los datos usados en cada programa.

JavaScript tiene la característica de ser un lenguaje débilmente tipado con tipado dinámico, es decir, que todos los tipos de datos se pueden asignar o reasignar a las variables que los almacenan. Por ejemplo, una variable que almacena un dato numérico puede posteriormente almacenar un dato de tipo booleano.

* Undefined
* Boolean
* Number
* String
* BigInt
* Symbol
* Null
* Object
* Function

**Datos primitivos de JavaScript**

Existen seis tipos de datos primitivos, es decir, que no son un objeto. Hay que tener cuidado con este concepto porque JavaScript en muchas situaciones convierte automáticamente los datos primitivos en objetos equivalentes a excepción de null y undefined.

**Undefined:** representa una variable que no ha sido declarada o a la cual no se le ha asignado un valor.

**Boolean:** representa un valor lógico y puede tener dos valores, ya sean true o false.

**Number:** permite representar y manipular valores numéricos como «37» o «-9.25».

**String:** representa datos textuales (cadenas de caracteres).

**BigInt:** representa valores numéricos que son demasiado grandes para ser representados por el tipo de dato number.

**Symbol:** es un valor primitivo único e inmutable.

**Otros tipos de datos de JavaScript**

**Null:** representa la ausencia intencional de cualquier valor, un valor nulo o «vacío».

**Object:** representa una colección de datos definidos y entidades más complejas.

**Function**: es una forma abreviada para funciones, aunque cada constructor de funciones se deriva del constructor Object. Son objetos con la capacidad de ser ejecutables.

**2. ¿Qué son las variables no declaradas y no definidas?**

**Indefinido:** Ocurre cuando una variable ha sido declarada pero no se le ha asignado ningún valor. Undefined no es una palabra clave.

**No declarada:** ocurre cuando intentamos acceder a cualquier variable que no esté inicializada o declarada antes usando la palabra clave var o const . Si usamos el operador ‘typeof’ para obtener el valor de una variable no declarada, nos enfrentaremos al error de tiempo de ejecución con el valor devuelto como indefinido. El ámbito de las variables no declaradas es siempre global.

**3. ¿Qué significa la palabra clave 'this' en JavaScript?**

This en JavaScript es una palabra clave muy utilizada dentro de funciones y clases, pues tiene un valor flexible. This significa esto en español y, como su nombre indica, hace referencia al objeto en cuestión. Es decir, si estamos creando cualquier función, la palabra clave this se usará para representar o llamar al objeto que dicha función está modificando.

A continuación, te ponemos un ejemplo con una variable llamada ejemplo:

const ejemplo = {

color: azul,

func: function ( ) {

return this.color;

},

};

Esta variable tiene dos propiedades, function y color. Aquí, this hace referencia a la constante ejemplo, pues es el objeto al que pertenece en este contexto. Entonces, al hacer que la función nos devuelva la propiedad this.color, el programa entiende que lo que hará realmente es devolvernos la propiedad ejemplo.color.

console.log (ejemplo.func());

// resultado esperado: azul

Una de las formas más comunes en las que se utiliza la palabra clave “this” en JavaScript es en las clases. Al leer nuestro post sobre las clases en JavaScript, podrás ver que utilizamos “this” para acceder a las distintas propiedades de una clase. Allí, accedemos a ellas utilizando el comando objeto, propiedad. La diferencia es que, en vez de utilizar el nombre del objeto, utilizamos “this”, esto es muy útil cuando queremos cambiar el nombre del objeto o este no tiene un nombre específico.

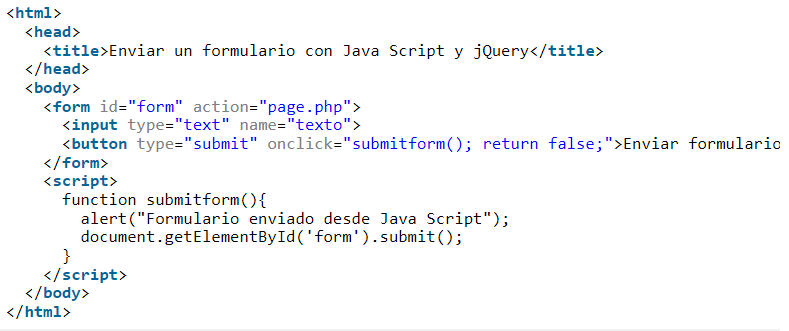
En este contexto, cuando nosotros estamos alterando el prototipo del objeto, la palabra clave “this” hace referencia a un array. Por ello, utilizamos la palabra clave “this” para poder acceder a sus propiedades. En la segunda línea de código de este método, puedes ver escrito “this.length”. En este caso, estamos usando “this” para acceder a la propiedad de longitud del array en cuestión. Además, hemos atribuido esta propiedad a la variable i.

Del mismo modo, estamos utilizando la palabra clave “this” como resultado de la función, pues la ponemos después de la palabra clave “return”. En este sentido, estamos diciendo que, después de la función, el programa nos devolverá el array que estamos manipulando.

**4. Explique cómo se puede enviar un formulario usando JavaScript.**

Java Script es un lenguaje muy poderoso, utilizado especialmente en el frontend para realizar páginas dinámicas que puedan brindar una mejor usabilidad a los usuarios, es por eso que en este breve tutorial vamos a ver cómo solucionar un problema muy frecuenta que se pueden encontrar los desarrolladores, el cual consiste en enviar un formulario utilizando Java Script y jQuery.

Vamos a empezar con el código para enviar el formulario utilizando el lenguaje de programación Java Script, primero crearemos un formulario HTML muy sencillos, seguido de la función que va a realizar el envío.



Del código mencionado es muy importante ver que el botón del formulario se encarga de hacer el envío aprovechando a través onclick, además es muy importante no olvidarse de incluir el return false ya que eso evita que el formulario se envie automáticamente sin ejecutar la función submitform de Java Script, otra forma de evitar eso es usar un input de type button como botón.

Generalmente, se envía un formulario cuando el usuario presiona un botón de envío. Sin embargo, a veces, se puede enviar el formulario mediante la programación con JavaScript.

JavaScript proporciona el objeto formulario que contiene el método submit(). Utilice el ‘id’ del formulario para conseguir el objeto formulario.

**5. ¿Qué son las cookies de JavaScript?**

Es un archivo con datos que se coloca en el navegador y que contienen información sobre el usuario que visita una determinada página web. En este artículo vemos cómo funcionan las cookies en Javascript, cuáles son sus principales parámetros y cómo crearlas, modificarlas o eliminarlas.

Las cookies.Js se definen por una serie de parámetros, los cuáles se implementan a través de código Js. Entre ellos está el «nombre-valor», el cual contiene la información a almacenar, una fecha de expiración que determina la fecha de validez de la cookie o el dominio y la ruta a la cual se envía. Pero veamos más a fondo estos conceptos y cómo se han de incluir en el código de la cookie.

**Document.cookie:** Mediante esta propiedad de pueden crear, modificar, eliminar y leer cookies en Javascript. Dentro de ella se incluyen diversos parámetros, como los que vemos a continuación.

**Name-Value:** Es un parámetro obligatorio a la hora de crear las cookies. «Name» se refiere al nombre que se adjudica a lascookies, mientras que «value» representa su valor. Por ejemplo, «name» podría ser «color\_favorito» y «value» podría ser «azul». Una de las principales diferencias a la hora de crear cookies en PHP. y en Javascript es que en Javascript se puede crear una cookie asignándole un valor vacío.

**Expire date:** Este parámetro establece una fecha de final de validez de la cookie. La fecha se ha de establecer en formato UTC.

Si se establece una fecha de validez anterior a la fecha actual, o se asigna un valor negativo en «max-age«, lo que se conseguirá es eliminar la cookie.

**Domain & Path:** Básicamente se trata de la URL para la cual la cookie es válida. En el caso de «Domain» se refiere al dominio, mientras que «Path» es el subdominio.

Debes tener en cuenta que la directiva same-origin policy no permite crear cookies para un dominio diferente al que crea la propia cookie. En el caso de los subdominios, se debe indicar para cuál se desea asignar la cookie. En caso de no asignar ninguna ruta, se creará automáticamente para la ruta de la página actual.

**SetCookie:** Con este script se crea una variable para una función con diferentes parámetros, principalmente el nombre de la cookie, su valor, y su período de validez. Como te hemos dicho anteriormente, si no se especifica la fecha se creará una sesion cookie, y si la fecha es anterior a la actual, la cookie se elimina.

**GetCookie:** Se trata de crear una función que devuelve el valor de una cookie determinada.