**Universidad Autónoma de Santo Domingo**



**Estudiante:** Luis Miguel Sánchez Cabreja

**Mátricula:** 100613071

**Asignatura:** Lenguaje de Programación 3

**Sec.** 03

**Profesor:** Manuel Ramón Mayrele Checo

**Tema:** Cuadro de Tecnologías

| **Etiqueta** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| **<html>** | Define el principio y el final del documento HTML. | <html> Contenido del documento </html> |
| **<head>** | Contiene metadatos del documento HTML. | <head> <title>Título del documento</title> </head> |
| **<title>** | Define el título del documento. | <title> Título de la página </title> |
| **<body>** | Contiene el contenido visible del documento. | <body> Contenido visible de la página </body> |
| **<h1> - <h6>** | Define encabezados de diferentes niveles. | <h1>Encabezado de nivel 1</h1> |
| **<p>** | Define un párrafo de texto. | <p>Este es un párrafo de ejemplo.</p> |
| **<a>** | Define un enlace. | <a href="https://www.ejemplo.com">Enlace</a> |
| **<img>** | Inserta una imagen. | <img src="imagen.jpg" alt="Descripción"> |
| **<ul>** | Define una lista desordenada. | html <ul> <li>Elemento 1</li> <li>Elemento 2</li> </ul> |
| **<table>** | Define una tabla. | html <table> <tr> <td>Celda 1</td> <td>Celda 2</td> </tr> </table> |

**HTML**

**CSS**

| **Etiqueta** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| **color** | Define el color del texto. | color: blue; |
| **font-size** | Define el tamaño de la fuente. | font-size: 16px; |
| **font-family** | Define la familia de fuentes a utilizar. | font-family: Arial, sans-serif; |
| **background-color** | Define el color de fondo de un elemento. | background-color: #FFFF00; |
| **border** | Define el borde de un elemento. | border: 1px solid black; |
| **margin** | Define el margen externo de un elemento. | margin: 10px; |
| **padding** | Define el espacio alrededor del contenido de un elemento. | padding: 5px; |
| **text-align** | Define la alineación horizontal del texto dentro de un elemento. | text-align: center; |
| **display** | Define cómo se muestra un elemento. | display: block; o display: inline-block; |
| **width / height** | Define el ancho / alto de un elemento. | width: 100px; o height: 50px; |

**BOOTSTRAP**

| **Etiqueta** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| **.container** | Crea un contenedor con ancho fijo y centrado. | <div class="container">Contenido</div> |
| **.row** | Define una fila para organizar columnas. | <div class="row">...</div> |
| **.col** | Define una columna dentro de una fila. | <div class="col">Contenido</div> |
| **.btn** | Estiliza un elemento como un botón. | <button class="btn btn-primary">Botón</button> |
| **.btn-primary** | Estiliza un botón como principal. | <button class="btn btn-primary">Botón</button> |
| **.btn-secondary** | Estiliza un botón como secundario. | <button class="btn btn-secondary">Botón</button> |
| **.btn-danger** | Estiliza un botón como peligroso. | <button class="btn btn-danger">Botón</button> |
| **.alert** | Estiliza un elemento como una alerta. | <div class="alert alert-info">Mensaje</div> |
| **.badge** | Estiliza un elemento como un distintivo. | <span class="badge badge-primary">Nuevo</span> |
| **.navbar** | Estiliza un elemento como una barra de navegación. | <nav class="navbar navbar-expand-lg navbar-light bg-light">...</nav> |

**JAVASCRIPT**

| **Etiqueta** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| **Variables** | Almacenan datos que pueden cambiar durante la ejecución del programa. | let x = 5; o const PI = 3.14; |
| **Funciones** | Bloques de código reutilizable que realiza una tarea específica. | javascript function saludar(nombre) { return 'Hola, ' + nombre + '!'; } |
| **Condicionales** | Estructuras que permiten ejecutar diferentes bloques de código según una condición. | javascript if (condición) { // código si es verdadero } else { // código si es falso } |
| **Bucles** | Permiten ejecutar un bloque de código varias veces. | javascript for (let i = 0; i < 5; i++) { // código a repetir } |
| **Eventos** | Acciones que ocurren en el navegador y desencadenan código JavaScript. | javascript elemento.addEventListener('click', function() { // código a ejecutar }); |
| **Objetos** | Estructuras de datos que almacenan propiedades y métodos. | javascript let persona = { nombre: 'Juan', edad: 30, saludar: function() { return 'Hola, soy ' + this.nombre; } }; |
| **Arrays** | Estructuras de datos que almacenan una colección de elementos. | javascript let numeros = [1, 2, 3, 4, 5]; |
| **Métodos de Array** | Funciones integradas en los arrays para realizar operaciones como agregar, eliminar o modificar elementos. | javascript numeros.push(6); // Agregar elemento al final |
| **DOM (Document Object Model)** | Representación de la página web que JavaScript puede manipular. | javascript document.getElementById('miElemento'); |
| **Promesas** | Utilizadas para operaciones asíncronas y manejo de resultados futuros. | javascript new Promise(function(resolve, reject) { // código asíncrono }); |

**XML**

| **Etiqueta** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| **<element>** | Define un elemento XML. | <nombre>Valor</nombre> |
| **<attribute>** | Define un atributo XML. | <persona id="1">Juan</persona> |
| **<!-- comentario -->** | Inserta un comentario en XML. | <!-- Este es un comentario --> |
| **<?xml version="1.0"?>** | Declara la versión de XML utilizada. | <?xml version="1.0"?> |
| **<!DOCTYPE>** | Define el tipo de documento XML. | <!DOCTYPE ejemplo SYSTEM "archivo.dtd"> |
| **CDATA** | Indica que el contenido debe ser tratado como caracteres de datos. | <![CDATA[contenido]]> |
| **<!-- DTD -->** | Define la estructura permitida para un documento XML. | xml <!-- Definición de DTD --> <!DOCTYPE ejemplo [ <!ELEMENT ejemplo (subelemento1, subelemento2)> ]> |
| **<?xml-stylesheet?>** | Especifica la hoja de estilo para el documento XML. | <?xml-stylesheet type="text/xsl" href="estilo.xsl"?> |
| **<processing-instruction>** | Define una instrucción de procesamiento XML. | <?target instrucción?> |
| **<namespace>** | Define un espacio de nombres XML. | xml <namespace:elemento xmlns:namespace="URI">Valor</namespace:elemento> |

La integración de estas tecnologías es crucial para el desarrollo web moderno. HTML proporciona la estructura, CSS ofrece el estilo visual, JavaScript agrega funcionalidad dinámica, Bootstrap acelera el desarrollo y la respuesta del diseño, y XML se puede utilizar para el intercambio de datos estructurados. Juntas, estas tecnologías permiten la creación de experiencias web ricas y completas que satisfacen las necesidades de los usuarios y los desarrolladores. La sinergia entre estas tecnologías es clave para el éxito en el desarrollo y la entrega de aplicaciones web modernas y eficientes.