| **Laboratorio 3** |
| --- |
| **Lanzamiento de la Actividad**   * **Abra** el proyecto SD-06-1: ConstructorsLab-Individual en VS code siguiendo las instrucciones mencionadas en la Guía de Programación Local TIFC. * **Examine** el fragmento de código de ejemplo proporcionado. |
| **Fragmento de código de ejemplo**   | function Player(p1, p2) {  this.name = p1  this.age = p2  }  const playerOne = new Player("joe", 40)  console.log(playerOne.name + " is " + playerOne.age + " years old.") | | --- |   Este programa de JavaScript se utiliza para construir nuevos objetos de acuerdo a una plantilla definida. Hará lo siguiente:   * Definir una función **constructora**, llamada Player (Jugador), que se utilizará como una plantilla de objeto.   + La función se utilizará como un constructor cuando se la llame con el operador "new". * Llamar a la función constructora para crear un **nuevo** objeto Player:   + Este **nuevo objeto Player** se construye y se asigna a la variable **playerOne**:     - **playerOne.name** se le asigna el valor de cadena **"joe".**     - **playerOne.age** se le asigna el valor de número **40**. * Imprimir una cadena que consista en **playerOne.name** y **playerOne.age** en la consola:   + La cadena de salida resultante sería: joe tiene 40 años ("joe is 40 years old"). |
| **Tareas**  Los constructores se utilizan para definir una plantilla para la creación de objetos. El operador "new" creará un nuevo objeto que se ajusta a la plantilla.  **Hoy explorarán las funciones constructoras y aprenderán a crear sus propios objetos definidos por el usuario.**   * Para cada una de estas tareas, escriba su código directamente en el archivo de tareas: ¡los scripts de evaluación evaluarán lo que ha escrito!   **Tarea 1**: Para esta tarea, se te ha presentado un constructor para un objeto Mail (Correo).   * Modifique el código de modo que el asunto sea la palabra "hello" (hola) y el mensaje sea la palabra "world" (mundo).   + La salida ya **ha sido definida**. Solo necesita modificar el código en la línea 8.   **Tarea 2**: Para esta tarea, se te ha presentado un constructor para un objeto Mail (Correo).   * Modifique el código para que se le solicite al usuario ingresar su propio asunto y mensaje, en ese orden.   + La salida ya **ha sido definida**. Solo necesitas modificar el código en la línea 8 y agregar los avisos para que el usuario ingrese el asunto y el mensaje.   **Tarea 3**: Para esta tarea, se te ha presentado un constructor para un objeto Mail (Correo). **Necesitará definir su propia salida.**   * Modifique el código para que se le solicite al usuario ingresar su propio asunto y mensaje, en ese orden. * Amplíe el constructor para incluir un método printMail() que muestre lo siguiente en la consola:   + <subj>: <msg>     - Por ejemplo, si las dos entradas son "hello" y "world", entonces la salida debería ser "hello: world".   **Tarea 4**: Para esta tarea**, deberá escribir su propio constructor**.   * Cree una función **constructora** llamada **Journey** (Viaje) que se utilizará como una plantilla para crear objetos que describen un viaje.   + Un nuevo objeto Journey debería contener dos propiedades: **start** (inicio) y **end** (fin). * Luego, **solicite** al usuario **dos** piezas de datos:   + from: El inicio de su viaje, o desde dónde vienen.   + to: El final de su viaje, o hacia dónde se dirigen. * Asegúrese de que se pueda crear un **nuevo objeto** según la línea de código proporcionada.   + La salida ya **ha sido definida**. Si encuentra un error, es posible que necesite modificar más su constructor. |
| **Tareas Adicionales**  Si ha completado las tareas anteriores, intente las siguientes tareas adicionales como un desafío adicional para obtener un bonus.  **Tarea 5**: Cree un constructor para un objeto FriendsList (Lista de Amigos) que almacenará una lista de nombres en un **array**.   * Su programa debería **solicitar** al usuario un número, luego solicitar **ese número** de veces que se enumeren los nombres uno por uno. * Luego, su programa debería imprimir el array directamente en la consola.   + La salida debería lucir así: [ 'name1', 'name2' ] ([ 'nombre1', 'nombre2' ])   **Tarea 6**: ¿Puede crear un constructor para un objeto que describa una lista de compras? Use esta tarea para **experimentar** con constructores.   * ¿Qué extensión debe tener la lista?   + No todos los usuarios querrán el mismo número de artículos. * ¿Cómo tratar las cantidades múltiples de un mismo artículo, como dos botellas de leche o una docena de huevos? * ¿Cómo recopilaría estos datos del usuario? * ¿Cómo almacenaría estos datos en un objeto? * ¿Cómo sería la función constructora de este objeto?   **Tarea 7**: ¿Puede crear un constructor para un objeto que pueda describir un coche? Use esta tarea para **experimentar** con constructores.   * Piense qué datos debe tener el objeto, cómo recopilaría esos datos del usuario y cómo los almacenaría.   + ¿Marca, modelo y año?   + ¿Color?   + ¿Número de puertas?   + ¿Kilometraje?   + ¿Motor de combustión o eléctrico?   + ¿Algo más? |
| **Puntuación**  El instructor puntuará las tareas de codificación al final de la sesión. |
| **Consejos**   * Un **constructor**, de forma similar a otras funciones, tiene **parámetros** para tomar **argumentos** como **entrada**. Estos argumentos se utilizan en el cuerpo de la función del constructor. * La palabra clave "this" actúa como un **marcador de posición**. Cuando el operador "new" se llama en un constructor, construye un objeto a partir del cuerpo del constructor y la palabra clave "this" se referirá a ese nuevo objeto. * Cuando definimos un método dentro de un objeto, no necesitamos darle un nombre a esa función. Lo mismo es cierto para los constructores. Las funciones que no tienen un nombre se llaman **funciones anónimas**. |