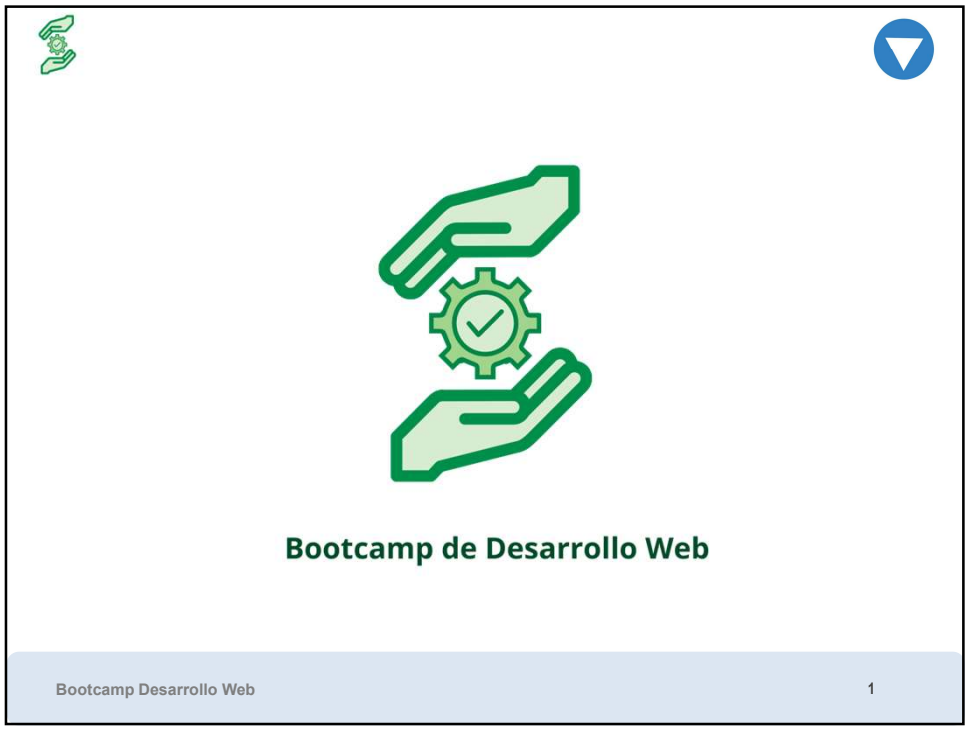
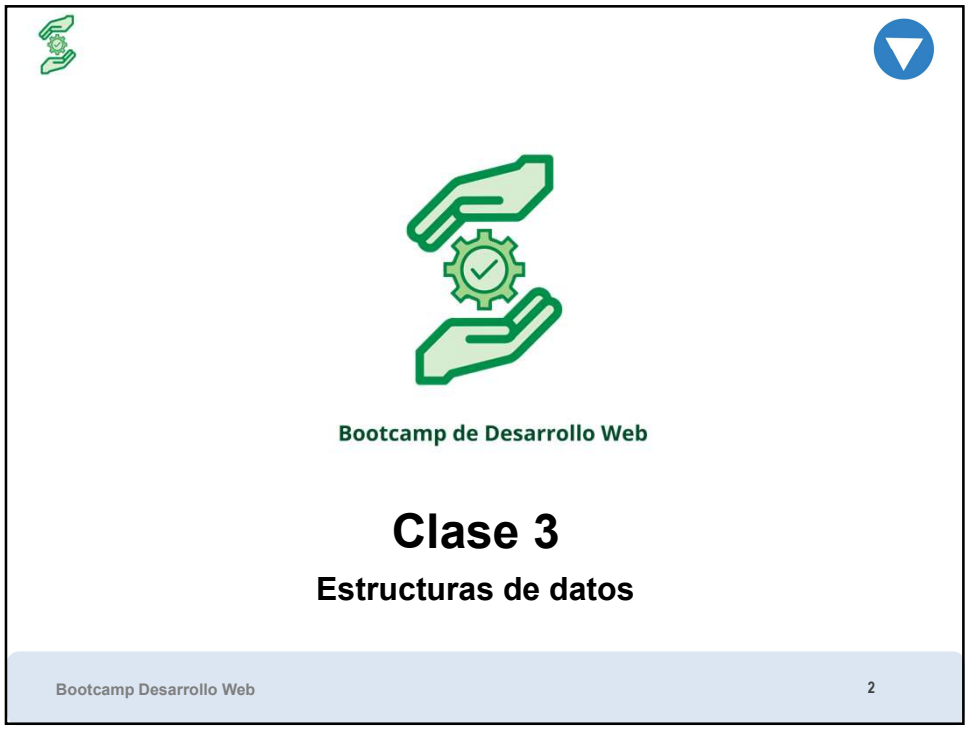

Bootcamp de Desarrollo Web
Bootcamp Desarrollo Web
1

1


Bootcamp de Desarrollo Web
Clase 3
Estructuras de datos
Bootcamp Desarrollo Web
2

2


1 - ¿Qué son las Estructuras de Datos?

Las estructuras de datos son formas organizadas y de almacenar, organizar y manipular conjuntos de datos de manera eficiente en programas informáticos.

Estas estructuras están diseñadas para gestionar y estructurar datos de tal manera que se puedan realizar operaciones como **inserción**, **búsqueda**, **modificación** y **eliminación** de forma efectiva.


Importancia de las Estructuras de Datos

Son fundamentales en programación, ya que permiten a los programadores manejar conjuntos de datos de manera eficiente y facilitan el desarrollo de algoritmos.

La elección de la estructura de datos adecuada es esencial, ya que diferentes estructuras son más eficientes para diferentes tipos de operaciones.

Algunas estructuras son excelentes para la búsqueda rápida de datos, mientras que otras son ideales para la inserción o eliminación eficiente de elementos.

Arrays (Arreglos)

Estructuras lineales que almacenan elementos del mismo tipo de manera contigua en la memoria.

Propiedades
- Índices
- Tamaño

5

Definición de un Array

En este ejemplo, se define un array Nombres que almacena cadenas y tiene una capacidad para 3 elementos.

```
Algoritmo Ejemplo_Creacion_Array
    Definir nombres Como Caracter;
    Dimension nombres[3];

    // Asignación de valores al array
    nombres[1] ← "Alice"
    nombres[2] ← "Bob"
    nombres[3] ← "Charlie"

FinAlgoritmo
```


6

Acceso a un Array

Para accesar al valor del elemento del array, se indica la posición del elemento al que se desea acceder de la siguiente manera:

```
Algoritmo Ejemplo_Creacion_Array
    Definir nombres Como Caracter;
    Dimension nombres[3];

    // Asignación de valores al array
    nombres[1] ← "Alice"
    nombres[2] ← "Bob"
    nombres[3] ← "Charlie"

    Escribir  nombres[2];

FinAlgoritmo
```


Manipular un Array

Se modifica el valor en la posición 2 del array asignándole el valor de “Diana” en lugar de “Bob”.

```
Algoritmo Ejemplo_Creacion_Array
    Definir nombres Como Caracter;
    Dimension nombres[3];

    // Asignación de valores al array
    nombres[1] ← "Alice"
    nombres[2] ← "Bob"
    nombres[3] ← "Charlie"

    Escribir  nombres[2];

    nombres[2] = "Diana";

    Escribir nombres[2];

FinAlgoritmo
```

Ejercicio

Supongamos que estás desarrollando un juego y necesitas almacenar los puntajes de los jugadores para realizar algunas operaciones básicas.

Requerimientos:

- Crear un array para almacenar los puntajes de 3 jugadores.
- Asignar puntajes aleatorios (pueden ser números enteros pequeños) a cada jugador.
- Mostrar por pantalla todos los puntajes asignados.
- Cambiar el puntaje del segundo jugador a un nuevo valor.
- Mostrar por pantalla la lista actualizada de puntajes.

Objetivo

Este problema implica la creación de un array para almacenar los puntajes, el acceso para mostrar los puntajes asignados y la manipulación para cambiar un puntaje específico. Es un ejemplo breve que ilustra la creación y manipulación básica de arrays en un contexto de juego.

Bootcamp de Desarrollo Web

Clase 3

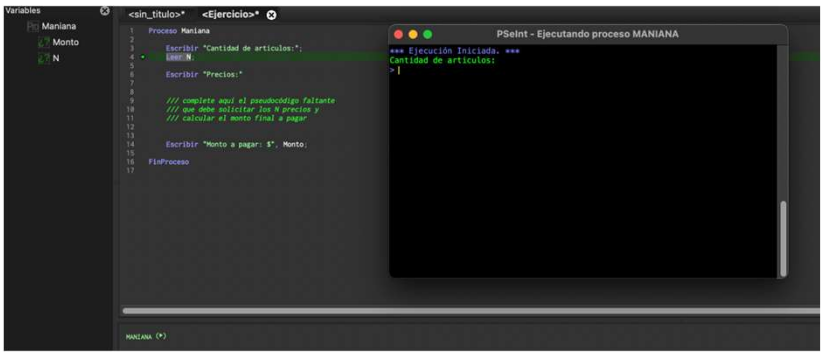
Debugging y manejo de errores

1- Consola de desarrollador

La consola de desarrollo, también conocida como consola del navegador o consola del desarrollador, es una herramienta integral que se encuentra en la mayoría de los navegadores web y entornos de programación modernos.

Esta consola proporciona un entorno de depuración y permite a los programadores inspeccionar, probar y depurar código directamente desde el navegador o el entorno de programación.

Consola de PseInt



2 – Depuración de código (Debugging)

El **debugging**, o **depuración** en español, se refiere al proceso de identificar, encontrar y corregir errores o fallos en un programa de computadora.

Estos errores pueden ser de diferentes tipos, como errores de sintaxis, lógicos o de comportamiento inesperado que afectan el funcionamiento normal de un software.

Bug

Un **"bug"** es un término comúnmente utilizado en el ámbito de la informática y la programación para referirse a un defecto, error o problema en un programa de software que causa un comportamiento no deseado o incorrecto.

Un bug puede manifestarse de diversas maneras y puede estar relacionado con errores de codificación, lógica incorrecta, malentendidos en los requisitos del software, entre otros factores.

Tipos de errores

- **Error de Sintaxis:** Se produce cuando el código escrito no sigue las reglas gramaticales del lenguaje de programación específico.
- **Error Lógico:** Ocuando el programa se ejecuta, pero produce resultados inesperados debido a una falla en la lógica implementada.
- **Error de comportamiento:** Un error de comportamiento se produce cuando el programa se ejecuta de acuerdo con la lógica implementada, pero no produce los resultados esperados.

Objetivos del debugging

Identificar errores: El objetivo principal es encontrar y comprender los errores que están causando el mal funcionamiento o el comportamiento no deseado en un programa.

Localizar la causa raíz: Una vez que se detecta un error, se busca su causa raíz para entender por qué está ocurriendo y cómo se puede corregir.

Corregir errores: Una vez identificado el problema, se procede a realizar las correcciones necesarias en el código para solucionar el error.

Métodos y técnicas comunes de debugging

1.Uso de puntos de interrupción (breakpoints): Se establecen puntos específicos en el código donde la ejecución se detiene para examinar el estado del programa en ese momento.

2.Inspección de variables: Se observa el valor de las variables en diferentes puntos del programa para comprender cómo cambian y si están contribuyendo al error.

3.Ejecución paso a paso: Permite ejecutar el código línea por línea para entender su flujo y detectar posibles problemas en cada paso.

Métodos y técnicas comunes de debugging

4.Registro de errores (logs): La impresión de mensajes o logs en puntos clave del programa ayuda a seguir el flujo de ejecución y a detectar posibles problemas.

5.Herramientas de depuración: Muchos entornos de desarrollo proporcionan herramientas específicas para depurar código, como consolas de desarrollo, herramientas de inspección, perfiles de rendimiento, etc.

Importancia del debugging

- Mejora la calidad del software al corregir errores que podrían afectar el funcionamiento.
- Ayuda a encontrar y resolver problemas de manera más eficiente, ahorrando tiempo en el desarrollo.
- Permite a los desarrolladores comprender mejor el código y el funcionamiento interno de un programa.

Bootcamp de Desarrollo Web

Clase 3

Fin