JavaScript avanzado

Sesión 1

¿Tienen alguna consulta o duda sobre el curso?

Universidad Tecnológica del Perú

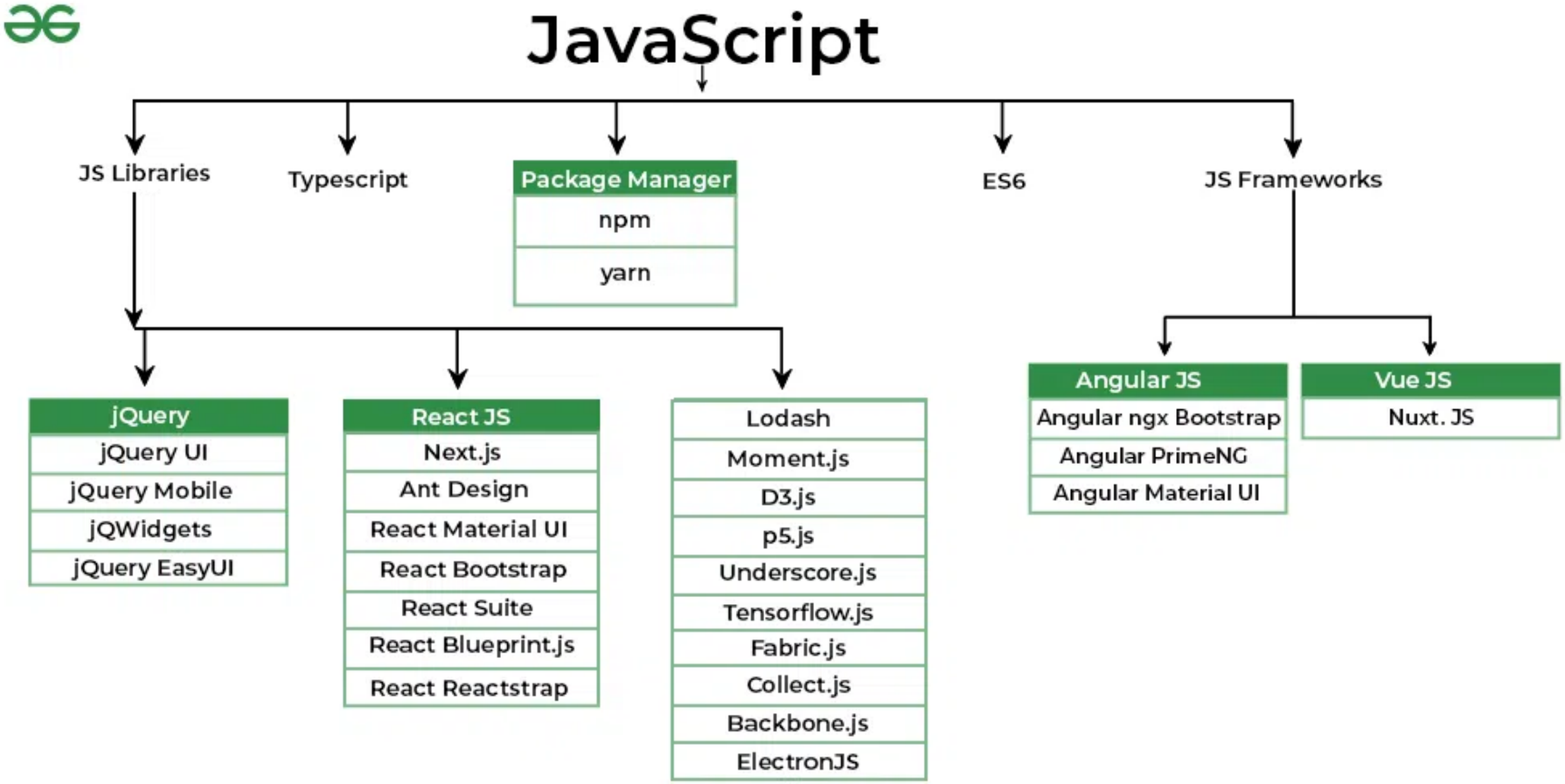
Logro de la sesión

Al finalizar la sesión el estudiante construye aplicaciones con JavaScript empleando diversos tipos de datos y estructuras de control para solucionar problemas algorítmicos.

Universidad Tecnológica del Perú

¿Qué es JavaScript?
¿Qué usos tiene?

Universidad Tecnológica del Perú



¿Cuál es la importancia del lenguaje JavaScript en el contexto actual?

Universidad Tecnológica del Perú

Contenido

JavaScript:

- Conceptos
- Tipos de datos
- Variables y constantes
- Sintaxis
  - Estructuras de control (condicional, repetitivo, try-catch, etc.)
  - Funciones, ejecución de eventos (onclick, onchange, etc.)
  - Salidas (document, alert, console, etc.)

Universidad Tecnológica del Perú

JavaScript

Concepto

- Es un lenguaje de programación del tipo intérprete (va traduciendo las sentencias una tras otra y no todas).
- Su principal fortaleza es en la programación web, pero se puede encontrar en distintos tipos de proyectos y aplicaciones.

Universidad Tecnológica del Perú

Tipos de datos en JavaScript

Números

Strings

Booleanos

Funciones

Objetos

Símbolos

Undefined

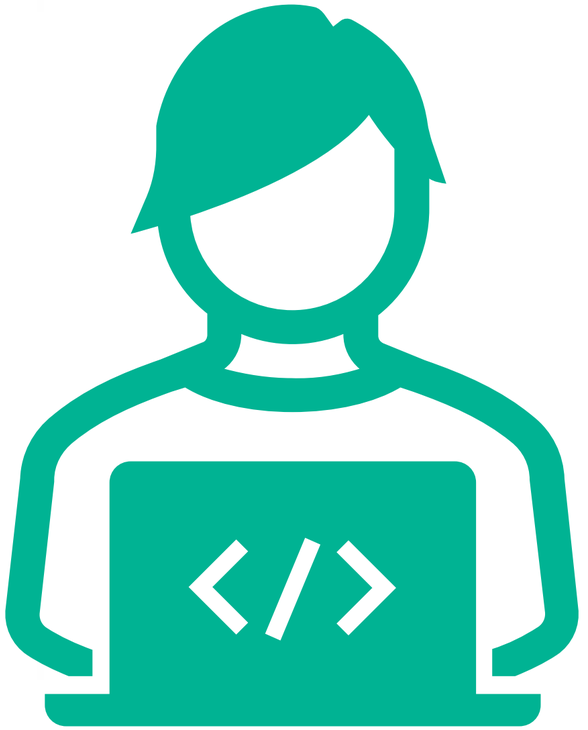
null

Array

Universidad Tecnológica del Perú

Configurando

- Configurar el entorno de trabajo para realizar las prácticas de laboratorio.



Universidad Tecnológica del Perú

Variables y constantes

- En JavaScript, tienes tres palabras clave principales para declarar variables que son: **var**, **let** y **const**
- Cada una tiene características y ámbitos diferentes.

- Además, en JavaScript se puede declarar una variable sin usar **var**, **let** o **const**. Sin embargo, no se recomienda esta práctica, ya que puede llevar a un comportamiento inesperado y errores difíciles de depurar.

Universidad Tecnológica del Perú

¿Qué es ámbito, y *hoisting* en JavaScript?

Universidad Tecnológica del Perú

Usando const

- **Ámbito**: De bloque.
- **Reasignación**: No se puede reasignar ni redeclarar.
- **Hoisting**: Se declara en la parte superior del bloque, pero debe ser inicializada en la declaración.
- **Consideraciones**: Ideal para valores que no deben cambiar a lo largo del código, como constantes o valores de configuración.

Usando let

- **Ámbito**: Funcional o global.
- **Reasignación**: Se puede reasignar, pero no volver a declarar.
- **Hoisting**: Se declara en la parte superior del bloque, pero no se inicializa hasta que se llega a su declaración.
- **Consideraciones**: Es la opción más común para declarar variables que pueden cambiar de valor dentro de un bloque.

Usando var

- **Ámbito**: Funcional o global.
- **Reasignación**: Se puede reasignar y volver a declarar.
- **Hoisting**: Se declara en la parte superior del ámbito, incluso si se declara más abajo.
- **Consideraciones**: Por su comportamiento, puede llevar a errores inesperados en código complejo. Se recomienda evitar su uso en JavaScript moderno.

Ejemplos de uso

```
// Uso de const, let y var
const PI = 3.14159; // Constante
let nombre = "Juan"; // Variable que puede cambiar
var edad = 30; // Evitar usar var en código moderno

// Dentro de un bloque
{
  let saludo = "Hola"; // Alcance de bloque
  console.log(saludo);
}

// Fuera del bloque, saludo no está definido
```

Universidad Tecnológica del Perú

Sintaxis de JavaScript

- Es **el conjunto de reglas** que define cómo se escribe y estructura el código en este lenguaje de programación.
- Incluye el uso de **palabras clave**, **operadores** y **signos** de puntuación para crear instrucciones que la computadora puede ejecutar.
- Entender y aplicar correctamente la sintaxis es crucial para el desarrollo de aplicaciones con este lenguaje.

Universidad Tecnológica del Perú

Estructuras de control

Condicional

- Las estructuras condicionales permiten ejecutar bloques de sentencias de código para una determinada condición, si esta no se cumple se puede optar por no realizar nada o llevar a cabo otro bloque de sentencias, de manera que no pueden ejecutarse ambos bloques, solo uno de ellos.

Estructuras de control condicionales

```
let diaSemana = 2;

switch (diaSemana) {
  case 1:
    console.log("Es lunes.");
    break;
  case 2:
    console.log("Es martes.");
    break;
  default:
    console.log("Día inválido.");
}
```

```
let edad = 18;

if (edad >= 18) {
  console.log("Eres mayor de edad.");
} else {
  console.log("Eres menor de edad.");
}
```

Universidad Tecnológica del Perú

Estructuras de control

Repetitivo

- Las estructuras repetitivas permiten ejecutar bloques de sentencias múltiples veces, aun si no conocemos el número de veces que debe repetirse el código. Se puede usar para ello una determinada condición, de manera que el bloque se detendrá cuando no se cumpla la condición.

Universidad Tecnológica del Perú

Estructuras de control repetitivas

```
let i = 0;
do {
  console.log("i =", i);
  i++;
} while (i < 5);
```

```
for (let i = 0; i < 5; i++) {
  console.log("El valor de i es:", i);
}
```

```
let contador = 0;
while (contador < 3) {
  console.log("Contador:", contador);
  contador++;
}
```

Universidad Tecnológica del Perú

Estructuras de control

Bloque try-catch

- Se utiliza para manejar errores.
- El código dentro de try se ejecuta normalmente, y si ocurre un error, se ejecuta el código dentro de catch.

```
try {
  let resultado = 10 / 0; // Genera un error
  console.log(resultado);
} catch (error) {
  console.error("Se produjo un error:", error);
}
```

Universidad Tecnológica del Perú

Eventos

- **onclick**: Se ejecuta cuando un usuario hace clic en un elemento como botones, enlaces, imágenes, etc.
- **onchange**: Se ejecuta cuando el valor de un elemento cambia como campos de texto, listas desplegables u otros.
- **onmouseout**: Se ejecuta cuando el cursor del ratón se aleja de un elemento. (Se emplea con **onmouseover** para generar efectos **hover**)
- **onsubmit**: Se ejecuta cuando se envía un formulario.

Eventos

- **onload**: Se ejecuta cuando se carga una página.
- **onunload**: Se ejecuta cuando se descarga una página.
- **onkeydown**: Se ejecuta cuando se presiona una tecla.
- **onkeyup**: Se ejecuta cuando se suelta una tecla.
- **ondblclick**: Se ejecuta cuando se hace doble clic en un elemento.

```
<!-- Ejemplo -->
<form onsubmit="validarFormulario()">
  <input type="submit" value="Enviar">
</form>
<script>
  function validarFormulario() {
    // Código para validar los datos del formulario
    if (/* condiciones de validación*/) {
      return true; // Permite enviar el formulario
    } else {
      alert("Por favor, complete todos los campos");
      return false; // Evita enviar el formulario
    }
  }
</script>
```

Universidad Tecnológica del Perú

Salidas

Document

- El objeto document representa la página web cargada en el navegador y sirve como punto de entrada al contenido de la página, que es el árbol DOM (Document Object Model)

Alert

- Permite desplegar una ventana flotante con el mensaje que deseemos, este mensaje puede ser cualquier tipo de variable, pero se mostrará convertido a texto.

Salidas

Console

- La consola de JavaScript proporciona acceso al objeto de ventana y a la consola de depuración del navegador.
- Los detalles de cómo funciona la consola varían de navegador en navegador
- Registrar mensajes en la consola es una forma básica de diagnosticar y solucionar problemas menores en el código.

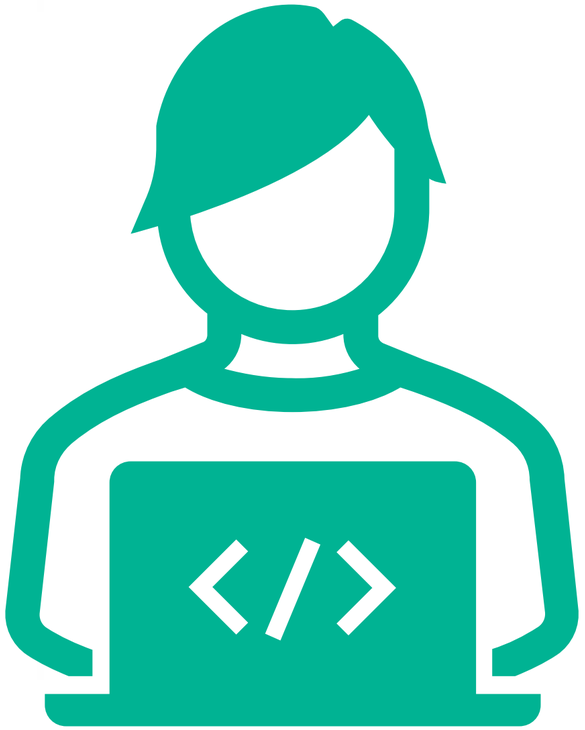
Universidad Tecnológica del Perú

Consola de JavaScript

Universidad Tecnológica del Perú

Programando

- Resolvemos ejercicios de algoritmia empleando JavaScript.



Universidad Tecnológica del Perú

¿Tienen alguna consulta o duda?

Universidad Tecnológica del Perú

Actividad

Resolver la actividad planteada en la plataforma.

Universidad Tecnológica del Perú

Cierre

¿Qué hemos aprendido hoy?

Elaboramos nuestras conclusiones sobre el tema tratado

Universidad Tecnológica del Perú


UTP
Universidad
Tecnológica
del Perú