JavaScript Avanzado

Sesión 4

¿Tienen alguna consulta o duda sobre la clase previa?

Universidad Tecnológica del Perú

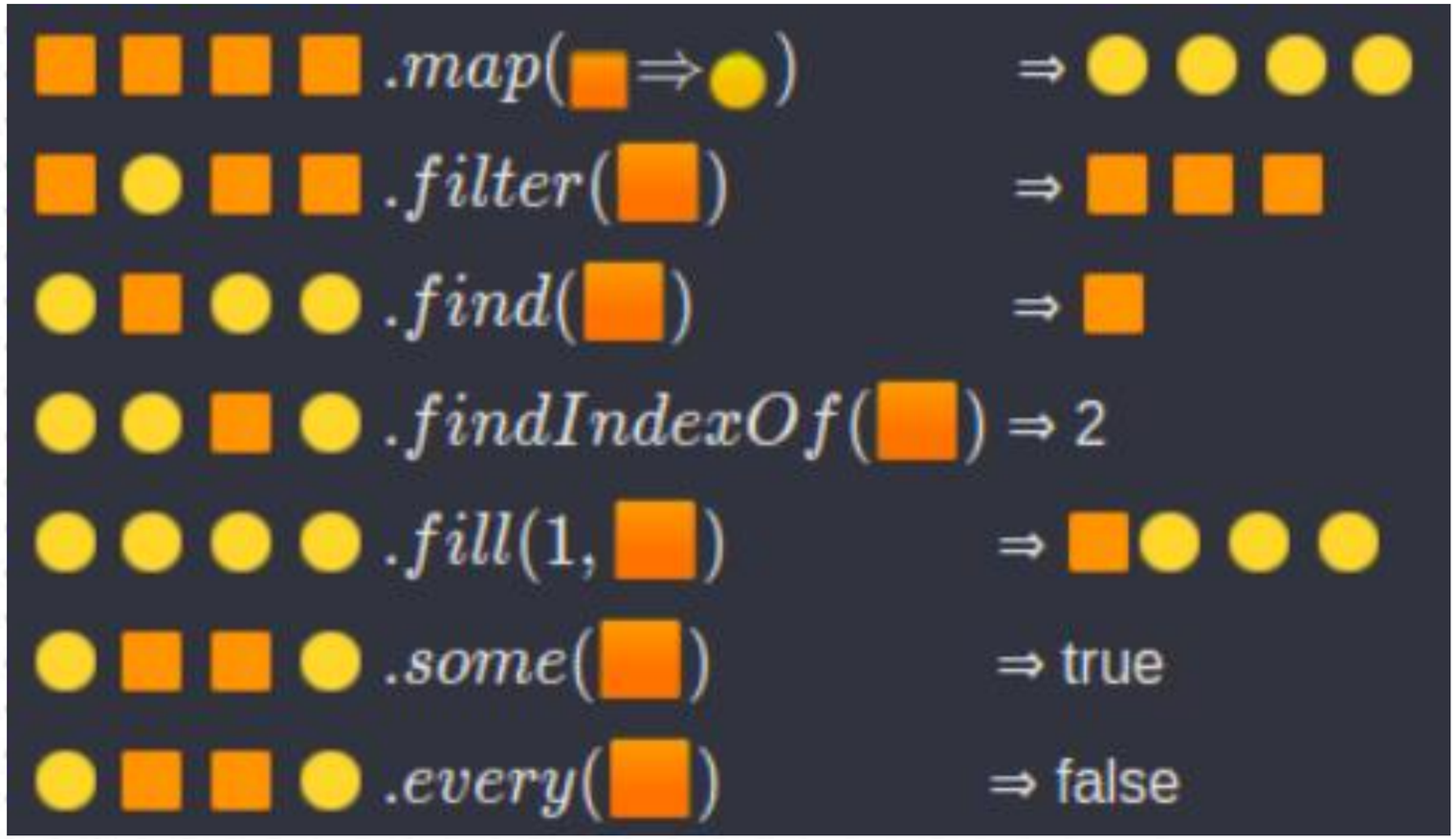
Logro de la sesión

Al finalizar la sesión el estudiante aplica el manejo de cadenas en el lenguaje JavaScript mediante la construcción de aplicativos web.

Universidad Tecnológica del Perú

¿Qué son los arreglos en programación?
¿Qué usos tienen?

Universidad Tecnológica del Perú



¿Cuál es la importancia de los objetos y arreglos en la programación?

Universidad Tecnológica del Perú

Contenido

Objetos y arreglos

- Arreglos (lineales, bidimensionales)
- Propiedades y funciones con arreglos
- Iteraciones
- Clases y objetos en JS
- Notación JSON
- Colecciones, mapas y el operador flecha.

Universidad Tecnológica del Perú

Arreglos en JavaScript

- Los **arrays** son colecciones de datos, múltiples datos almacenados en una sola variable, para definir un arreglo se debe colocar sus elementos dentro de corchetes y separados por comas.

Universidad Tecnológica del Perú

Los **arrays** se puede crear arreglos de dos formas:

```
//Primera forma:
let listA = ["Argos","Pelusa","Shirley"];



//Segunda forma:
let listB = new Array("Argos","Pelusa","Shirley");
```

Universidad Tecnológica del Perú

Arreglos bidimensionales en JavaScript

- Un arreglo **unidimensional** es una colección ordenada de elementos, donde cada elemento tiene un índice único que comienza desde 0.

- Un arreglo **bidimensional** es como una tabla, con filas y columnas. Cada elemento está identificado por dos índices: uno para la fila y otro para la columna.

Universidad Tecnológica del Perú

```
// Arreglo unidimensional:
let lista = [“Lunes",“Martes",“Miércoles"];



// Arreglo bidimensional:
let matriz = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
];
```

Universidad Tecnológica del Perú

Propiedades y funciones

- La propiedad principal de un array es **length**, que devuelve la cantidad de elementos de un array.

```
let lista = new Array(1,2,3,4,5,6);
console.log(lista.length); //6
```

Universidad Tecnológica del Perú

Funciones

join()

toString()

pop()

push()

shift()

unshift()

length()

Métodos REDUCE, FILTER y MAP

- **REDUCE** permite reducir todos los elementos del arreglo a un solo elemento. Como una concatenación de elementos o suma de valores
- **FILTER** permite reducir crear un arreglo con los elementos que cumplan con la condición indicada.
- **MAP** permite crear un arreglo a partir de los elementos del arreglo inicial.

Universidad Tecnológica del Perú

Clases y objetos en JS

- Una clase es un "molde" o "plantilla" que se utiliza para crear objetos con propiedades y métodos similares
- Se utiliza la palabra clave **class** para definir una clase en JavaScript.

```
class Persona {
    constructor(nombre, edad) {
        this.nombre = nombre;
        this.edad = edad;
    }

    saludar() {
        console.log(`${this.nombre}
            ${this.edad} años.`);
    }
}
```

Universidad Tecnológica del Perú

Componentes

- **Constructor**: El método constructor es un método especial para crear e inicializar un objeto creado con una clase.
- **Métodos**: Dentro de una clase, puedes definir métodos que representen comportamientos.

Instancias

- Para crear una instancia de una clase, se utiliza la palabra clave **new** seguida del nombre de la clase.

Universidad Tecnológica del Perú

Objetos

- En JavaScript, un objeto es una colección de pares clave/valor.
- Los objetos son fundamentales para almacenar y manipular datos.
- Se accede a las propiedades de un objeto utilizando la notación de punto o la notación de corchetes.

Objetos

```
//Notación Literal

let persona = {
    nombre: 'Jhon',
    edad: 30,
    saludar: function() {
        console.log(`Soy ${this.nombre}.`);
    }
};

persona.saludar();
```

```
//Constructor de Objetos

let persona = new Object();
persona.nombre = 'Jhon';
persona.edad = 30;
persona.saludar = function() {
    console.log(`Soy ${this.nombre}.`);
};


persona.saludar();
```

Universidad Tecnológica del Perú

Objetos

```
let persona = {
    nombre: 'Jhon',
    edad: 30
};

console.log(persona.nombre); // "Jhon"
console.log(persona['edad']); // 30

persona.nombre = 'Jane';
persona['edad'] = 25;

// { nombre: "Jane", edad: 25 }
console.log(persona);
```

Universidad Tecnológica del Perú

Modularidad en JavaScript

- Al trabajar con proyectos JavaScript de mayor envergadura, es fundamental organizar el código en módulos o archivos separados para mejorar la legibilidad, la reutilización y la mantenibilidad.

- Esto implica la capacidad de definir clases en un archivo y luego utilizarlas en otros.

Módulos ES6 (import/export)

```
// clase1.js
export class MiClase {
    // ...
}

// clase2.js
import { MiClase } from './clase1.js';
let miObjeto = new MiClase();
```

Notación JSON

- **JSON** (JavaScript Object Notation) es un formato de texto ligero y de fácil lectura que se utiliza para intercambiar datos entre sistemas.
- Está basado en la sintaxis de los objetos en JavaScript, lo que lo hace muy popular en el desarrollo web.
  - **Objetos**: Representados por llaves {}, contienen pares clave-valor separados por comas. Las claves deben ser cadenas de texto y los valores pueden ser de cualquier tipo de dato soportado por JSON.
  - **Arreglos**: Representados por corchetes [], son colecciones ordenadas de valores. Los valores pueden ser de cualquier tipo de dato soportado por JSON.

Notación JSON

```
{
    "nombre": "Juan",
    "edad": 30,
    "ciudad": "Madrid",
    "hobbies": ["leer", "programar", "viajar"],
    "contactos": {
      "email": "juan@ejemplo.com",
      "telefono": "123456789"
    }
}
```

Universidad Tecnológica del Perú

Colecciones y mapas en JS

- **Sets**: Son colecciones de valores únicos, es decir, no permiten duplicados. Son útiles cuando necesitas almacenar elementos y asegurarte de que cada uno es único.
- **Mapas** (Map) son estructuras de datos que permiten almacenar pares clave/valor, donde las claves pueden ser de cualquier tipo (incluyendo objetos y funciones).
  - Se diferencian de los objetos en que las claves no están limitadas a ser cadenas o símbolos.
  - Los métodos más comunes incluyen set(), get(), has(), delete(), y clear().

Universidad Tecnológica del Perú

Operador flecha =>

El **operador flecha (=>)** es una forma concisa de escribir funciones en JavaScript.

- Fue introducido en **ECMAScript 2015** (**ES6**) y proporciona una sintaxis más corta para escribir expresiones de función.
- No tienen su propio this, por lo que heredan el this del contexto en el que fueron definidas.
- Son ideales para funciones cortas y anónimas.

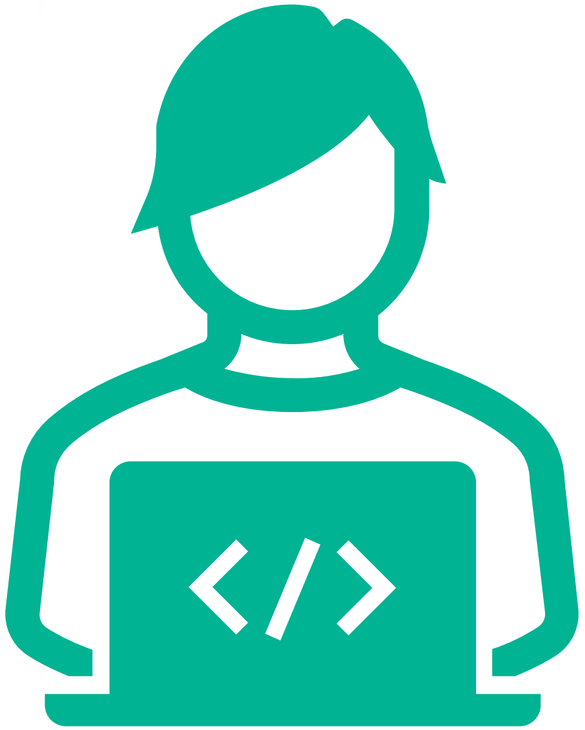
Ejemplo

```
// Función tradicional
function sumar(a, b) {
    return a + b;
}

// Función flecha equivalente
const sumar = (a, b) => a + b;

console.log(sumar(3, 5)); // 8
```

Universidad Tecnológica del Perú

Programando

- Construir algoritmos empleando estructuras de datos y el operador flecha.



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