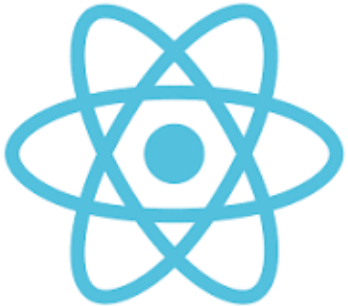
<codoa codo/>

Agencia de Aprendizaje a lo largo de la vida

# ESPECIALIZACIÓN REACT



BA

<codoa codo/>


# **Clase 03.** REACT JS

## *REPASO JS 2*

## *+ JSX*

BA

# JSON

# POPULARIDAD DE JSON

Es un formato de texto sencillo utilizado para el intercambio de datos entre distintos sistemas.

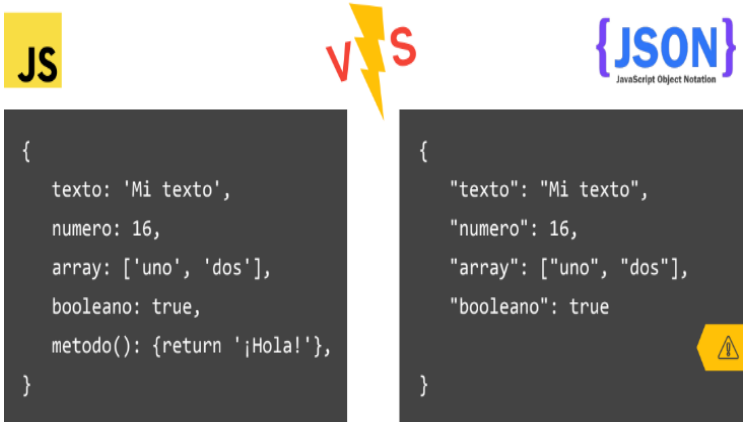
En la web, la mayoría de las peticiones y sus respuestas viajan como texto plano, es decir, texto sin codificaciones especiales.

JSON al ser una **cadena de texto** simple, es un **formato ideal para transmitir información** entre sitios y aplicaciones web.

Especialmente si tenemos en cuenta que Javascript está presente en todos los navegadores modernos.


La otra ventaja de JSON es que cualquier lenguaje de programación puede interpretarlo con facilidad. De hecho la mayoría de los lenguajes web trabaja nativamente con JSON.

# ESTRUCTURA DE UN JSON




**JSON no soporta métodos**

Como su nombre lo indica, JSON es muy similar a un objeto. La diferencias entre ellos son:

Objeto

- Admite comillas simples y dobles
- Las claves del objeto van sin comillas
- Podemos escribir métodos
- Se recomienda poner una coma en la última propiedad

JSON

- Sólo se pueden usar comillas dobles
- Las claves van entre comillas
- No admite métodos, sólo propiedades y valores
- No podemos poner una coma en el último elemento

# DESTRUCTURING

## SIN USAR DESTRUCTURING

```
let seleccion = ["Messi", "De Paul", "Martinez", "Di Maria","Alvarez"];
// extraemos datos de un array SIN DESTRUCTURACIÓN
let elDiez= seleccion[0]
console.log(elDiez)   'Messi'
```

Para extraer datos de un array, es necesario crear una variable y asignarle un elemento del array usando el operador de índice.

```
// OBJETO
let auto = {
    marca:"SsangYong",
    modelo:"Musso",
    anio:1996
}
// extraemos datos de un OBJETO SIN DESTRUCTURACIÓN
    let marcaAuto= auto.marca

    console.log(marcaAuto)   'SsangYong'
```

Para extraer datos de un objeto, es necesario que crear una variable y asignarle una propiedad específica de ese objeto.

## CON DESTRUCTURING

```
let colores = ["Rojo","Amarillo", "Verde"]
let [color1,color2,color3] = colores
console.log(color1)    'Rojo'

let seleccion = ["Messi", "De Paul", "Martinez", "Di Maria","Alvarez"]
let[elDiez,elDibu] = seleccion;
console.log(elDibu)    'De Paul'
```

Partiendo de un array previamente definido, se transfiere cada dato a las variables que definimos nosotros.

Javascript le asignará a cada variable el dato extraído de la estructura que elijamos, respetando el orden original.

Si queremos saltar un valor, podemos dejar vacío el nombre de la variable que corresponde con esa posición.

## CON DESTRUCTURING

```
let auto = {
    marca:"SsangYong",
    modelo:"Musso",
    anio:1996
}

let {marca,modelo:MiModelo} = auto

console.log(MiModelo)   'Musso'
```

Para desestructurar un objeto literal, creamos una variable (podemos usar var, let o const), y entre llaves, declaramos el o los nombres de las propiedades que queremos extraer.
A esa estructura la igualamos al objeto del cual queremos extraer los datos.

Partiendo de un objeto previamente definido, se transfiere cada propiedad o método a una o más variables que definamos.
Javascript le asignará a cada variable el valor de la propiedad que hayamos elegido.

Es posible que en algún caso necesitemos cambiarle el nombre a la variable que estamos creando.
En ese caso a continuación de la propiedad que estamos extrayendo colocamos dos puntos : seguidos del nuevo nombre.

# JSX

# QUE ES JSX

React usa JSX para declarar qué debe renderizar. JSX es una extensión de JavaScript que permite escribir un código más cercano visualmente a HTML, que mejora la legibilidad del código y hace que sea más fácil de entender.

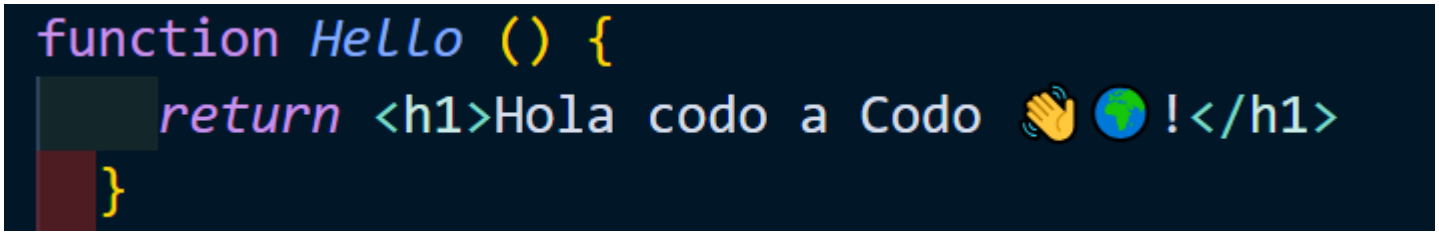
Sin JSX, deberíamos usar React.createElement para crear los elementos de la interfaz manualmente de esta forma:

```
import { createElement } from 'react'

function Hello () { // un componente es una función! 👀
  return React.createElement(
    'h1', // elemento a renderizar
    null, // atributos del elemento
    'Hola Mundo 👋🌎!' // contenido del elemento
  )
}
```

# QUE ES JSX

Esto es muy tedioso y poco legible. Por eso, React usa JSX para declarar qué debe renderizar.
Por eso usamos JSX de esta forma:




Ambos códigos son equivalentes.

## ¿Cómo se transforma el JSX?

El JSX se transforma en código JavaScript compatible en el navegador usando un transpilador o compilador.
Hay casos especiales en los que un transpilador no es necesario. Por ejemplo, Deno tiene soporte nativo para la sintaxis JSX y no es necesario transformar el código para hacerlo compatible.