

CURSO

FULL STACK DEVELOPER
NIVEL INICIAL

UNIDAD 9
Objetos y JSON

UNIDAD 9
Objetos y JSON



Full Stack Developer Inicial

Objetos y JSON

¿Qué es un objeto?

Un objeto es una colección de datos relacionados.

Posee variables y funciones, que se denominan propiedades y métodos.

Cada propiedad tiene un nombre y un valor.

Definir un objeto

```
let persona = { nombre:
```

```
  'Lucas',
```

```
  apellido: 'Martinez',
```

PROPIEDAD

```
  edad: 47,
```

```
  sexo: 'masculino',
```

NOMBRE

```
  dni: 23.305.379
```

VALOR

```
  casado: true,
```

```
  hijos: ['Ana Martinez', 'Marcos Martinez']
```

```
}
```

Acceder a propiedades

Para acceder a las propiedades de un objeto tenemos 2 opciones:

- Objeto.propiedad
- Objeto[propiedad]

Siguiendo el ejemplo anterior obtendríamos lo siguiente:

```
console.log(persona.nombre) ---> 'Lucas' console.log(persona[dni]) --->
```

```
23.305.379
```

Mutabilidad

Por defecto en JS los objetos son mutables, esto significa que además de obtener el valor de una propiedad de un objeto también podemos modificarlo:

```
persona.nombre = 'Fernando'
```

```
console.log(persona.nombre) ---> 'Fernando'
```

Referencia

Cuando definimos objetos en JS la variable que lo declara hace referencia al lugar de memoria donde se encuentra almacenado ese objeto.

Por lo tanto:

```
let persona1 = {nombre: 'Fernando'}; let persona2 = persona1;  
let persona3 = {nombre: 'Fernando'};
```

```
console.log(persona1 === persona2) ---> true console.log(persona1 ===  
persona3) ---> false
```

```
persona1.nombre = 'Lucas'; console.log(persona2.nombre) --->  
'Lucas'
```

Destructuring

En JS existe una expresión que nos permite desempaquetar valores de arrays u objetos en grupos de variables, permitiéndonos simplificar y crear código más legible.

Teniendo en cuenta el ejemplo anterior:

```
let { nombre, apellido, edad, sexo, dni, casado, hijos } = persona; console.log(nombre)
```

```
---> 'Lucas'
```

```
console.log(apellido) ---> 'Martinez' console.log(edad)
```

```
---> 47 console.log(sexo) ---> 'masculino'
```

```
console.log(dni) ---> 23.305.379 console.log(casado) --->
```

```
true
```

```
console.log(hijos) ---> ['Ana Martinez', 'Marcos Martinez']
```


¿Qué es JSON?

Corresponde a las siglas de Java Script Object Notation.

Es un formato ligero para almacenar y transportar datos.

Se usa a menudo cuando los datos se envían desde un servidor a una página web.

Es "autodescriptivo" y fácil de entender.

Reglas de sintaxis

Su sintaxis es muy similar a la de los objetos en **JS**, la principal diferencia es que los nombres de las propiedades también van entre comillas:

```
{  
  "empleados": [  
    {"nombre": "Juan", "apellido": "Perez"},  
    {"nombre": "Karina", "apellido": "Gonzalez"},  
    {"nombre": "Luz", "apellido": "López"},  
    {"nombre": "Marcos", "apellido": "Villalba"}  
  ]  
}
```

Convertir un JSON a un objeto JS

Un uso común de JSON es leer datos de un servidor web y mostrar los datos en una página web.

El formato JSON es sintácticamente idéntico al código para crear objetos JavaScript, gracias a esta similitud, es muy sencillo convertir datos JSON en objetos JavaScript nativos utilizando el método `JSON.parse()`

Convertir un JSON a un objeto JS

```
let data = `{ "empleados": [
    {"nombre": "Juan", "apellido": "Perez"},
    {"nombre": "Karina", "apellido": "Gonzalez"},
    {"nombre": "Luz", "apellido": "López"},
    {"nombre": "Marcos", "apellido": "Villalba"}
  ]
}`

let processedData = JSON.parse(data);

console.log(processedData.empleados[0]);

//{nombre: "Juan", apellido: "Perez"}
```

Convertir un objeto JS a JSON

De igual manera podemos convertir un objeto JS a formato JSON utilizando `JSON.stringify()`

```
let data = {  
  empleados: [  
    {nombre: "Juan", apellido: "Perez"},  
    {nombre: "Karina", apellido: "Gonzalez"},  
    {nombre: "Luz", apellido: "López"},  
    {nombre: "Marcos", apellido: "Villalba"}  
  ]  
}
```

```
let dataJson = JSON.stringify(data);
```



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES