# **Resumen Conceptual sobre Objetos Literales en JavaScript**

 Definición de Objetos Literales

Un objeto literal en JavaScript es una estructura de datos que permite almacenar pares clave-valor. Las claves son conocidas como propiedades y los valores pueden ser cualquier tipo de dato, incluyendo otros objetos.  
➡️ Ejemplo: `const persona = { nombre: 'Tony', apellido: 'Stark', edad: 45 };` crea un objeto con propiedades clave-valor.

 Acceso a las Propiedades del Objeto

Las propiedades de un objeto pueden ser accedidas utilizando la notación de punto (`objeto.propiedad`) o con notación de corchetes (`objeto['propiedad']`).  
➡️ Ejemplo: `persona.nombre` devuelve 'Tony'.

 Añadir Métodos y Objetos Anidados

Un objeto en JavaScript puede contener otros objetos o incluso funciones (métodos). Esto permite crear estructuras de datos complejas dentro de un mismo objeto.  
➡️ Ejemplo: `const persona = { nombre: 'Tony', direccion: { ciudad: 'New York', zipcode: 10001 } };`.

 Copia de Objetos y Referencias

Al copiar un objeto utilizando la asignación simple (`const obj2 = obj1`), se copia la referencia del objeto original. Esto significa que cualquier modificación en el nuevo objeto también afectará al original.  
➡️ Ejemplo: `persona2 = persona` copia la referencia de `persona`, por lo que los cambios en `persona2` afectan también a `persona`.

 Clonar un Objeto con el Operador Spread

Para crear una copia profunda de un objeto (un clon), se puede utilizar el operador spread (`...`). Esto permite extraer todas las propiedades del objeto original y asignarlas a un nuevo objeto, evitando modificar el original.  
➡️ Ejemplo: `const persona2 = { ...persona };` crea un clon de `persona`.

 Buenas Prácticas al Usar Objetos

Es importante evitar caracteres especiales en las claves de los objetos, como tildes o eñes, para asegurar que el código sea compatible en diferentes entornos y navegadores.  
➡️ Ejemplo: Usar `ciudad: 'New York'` en lugar de `ciudád: 'New York'`.

 Respaldo del Archivo

Es recomendable realizar copias de seguridad del código para evitar pérdidas de información. Guardar los archivos en carpetas organizadas permite mantener un registro de los ejercicios y avances.  
➡️ Ejemplo: Copiar `index.js` en la carpeta `bases` y renombrarlo como `03-objetos-literal.js`.