# **Resumen Conceptual sobre Funciones en JavaScript**

 Funciones Básicas

Una función en JavaScript se utiliza para encapsular un bloque de código que puede ser reutilizado. Las funciones pueden recibir parámetros y devolver valores. Esto permite crear comportamientos y lógicas específicas que se pueden ejecutar en diferentes contextos.  
➡️ Ejemplo: `function saludar(nombre) { return `Hola, ${nombre}`; }` retorna un saludo personalizado.

 Declaración con const

Para evitar problemas de reasignación accidental, es recomendable declarar funciones con `const`, lo que previene modificar la referencia de la función.  
➡️ Ejemplo: `const saludar = function(nombre) { return `Hola, ${nombre}`; };`.

 Funciones de Flecha

Las funciones de flecha (`=>`) ofrecen una sintaxis compacta para escribir funciones y permiten una lectura más clara. Se recomienda usarlas en funciones simples, especialmente en React.  
➡️ Ejemplo: `const saludar = (nombre) => `Hola, ${nombre}`;` es equivalente a una función regular.

 Funciones con Return Implícito

Cuando una función de flecha solo realiza un `return`, se puede omitir el uso de llaves y la palabra `return`, lo que simplifica el código.  
➡️ Ejemplo: `const saludar = (nombre) => `Hola, ${nombre}`;` es equivalente a usar llaves y `return`.

 Retorno de Objetos en Funciones de Flecha

Para retornar objetos directamente en funciones de flecha, el objeto debe ser envuelto entre paréntesis, evitando que JavaScript interprete las llaves como el inicio del cuerpo de la función.  
➡️ Ejemplo: `const getUser = () => ({ uid: 'ABC123', username: 'el\_papi1501' });`.

 Buenas Prácticas al Usar Funciones

Es recomendable evitar redefinir funciones o cambiar sus referencias. Al declarar funciones con `const`, se garantiza que la función se mantenga intacta durante el ciclo de vida del código.  
➡️ Ejemplo: Usar `const` para todas las declaraciones de funciones y evitar reasignaciones.

 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 `05-funciones.js`.