

En JavaScript existen dos tipos principales de funciones: las funciones regulares y las funciones flecha. Las funciones regulares se definen con la palabra clave `function` y pueden tener un nombre, parámetros y un bloque de código. Estas funciones tienen su propio contexto de `this`, lo que significa que `this` hace referencia al objeto que las llama. También cuentan con el objeto especial `arguments`, que permite acceder a todos los parámetros enviados, pueden utilizarse como constructores con `new` y poseen la propiedad `prototype`.

Por otro lado, las funciones flecha se escriben de forma más corta utilizando la sintaxis `(param) => { ... }`, lo que las hace más concisas y fáciles de leer. Sin embargo, no crean su propio `this`, sino que heredan el `this` del lugar donde fueron definidas, no tienen acceso al objeto `arguments` (por lo que, si se necesita acceder a los parámetros, se debe usar la sintaxis de parámetros rest `...args`), no pueden emplearse como constructores y no tienen la propiedad `prototype`.

En resumen, las funciones regulares son más flexibles y pueden utilizarse en más contextos, mientras que las funciones flecha ofrecen una sintaxis más breve y clara, pero con ciertas limitaciones en cuanto a su uso de `this`, `arguments` y su rol como constructores.

## **Bibliografía**

Trinh, O. (2023, octubre 23). *Array function vs regular function*. Medium.

<https://medium.com/@olivier.trinh/array-function-vs-regular-function-n-258c6a848e4d>