JavaScript Function Decision Cheat Sheet

Utility / Normal Logic

Use Normal Function or Arrow Function.
Example:
function add(a,b){ return a+b; }
const addA = (a,b) => a+b;

Short Inline Use

Use Arrow Function or Anonymous Function.
Example:
setTimeout(() => console.log('Done!'), 1000);

Array Operations

Use Higher-Order Functions (map, filter, reduce).
Example:
[1,2,3].map(n => n*2);

Async / API Calls

Use Async/Await (modern) or Callbacks (legacy).
Example:
async function getData(){
const res = await fetch(url);
return await res.json(); }

Private State

Use Closure or Factory Function.
Example:
function secretCounter(){ let c=0; return ()=> ++c; }

Reusable Objects

Use Factory Function or Class/Constructor.
Example:
class User { constructor(n){ this.name=n;} greet(){ return 'Hi '+this.name; } }

Algorithms / Trees

Use Recursive Functions.
Example:
function fact(n){ return n===1?1:n*fact(n-1); }

Infinite Values

Use Generator Functions.
Example:
function* gen(){ let i=1; while(true) yield i++; }

Private Scope

Use IIFE.
Example:
(function(){ console.log('Runs instantly!'); })();