Aspect	Explicit Binding	Function Borrowing
Definition	Manually setting the value of `this` for a function using `call()`, `apply()`, or `bind()`.	When one object borrows a method from another object to use it as its own.
Purpose	Control what `this` refers to regardless of how the function is invoked.	Reuse functions or methods between objects without redefining them.
How It Works	Use `call()`, `apply()`, or `bind()` to set `this` explicitly.	Borrow the function via `call()`, `apply()`, or by assigning the method to the new object.

Example	function greet() {
	console.log(`Hello, my name is ${this.name}`);
	}
	const person = { name: 'Alice' };
	greet.call(person); // Output: Hello, my name is Alice
		const person = {
		name: 'Alice',
		greet() {
		console.log(`Hello, my name is ${this.name}`);
		}
		};
		const anotherPerson = { name: 'Bob' };
		person.greet.call(anotherPerson); // Output: Hello, my name is Bob

Key Differences	General-purpose `this` binding vs. sharing methods between objects.	Explicit binding sets `this`; borrowing allows sharing functionality.