

Situation	Value
<i>At the Global</i>	<ul style="list-style-type: none"> - In browser (non-strict): window - In browser (strict mode): undefined <pre>console.log(this); // window (non-strict), undefined (strict)</pre>
Inside an Object Method	<p>Refers to the object the method belongs to (the one before the . at call time).</p> <pre>const obj = { name: "Siva", greet() { console.log(this.name); } }; obj.greet(); // "Siva"</pre>
Inside the Standalone non-Arrow Function	<ul style="list-style-type: none"> - Non-strict: global object (window / global) - Strict: undefined <pre>function show() { console.log(this); } show(); // window (non-strict), undefined (strict)</pre>
Inside an Arrow Function(standalone)	<p>Lexically inherits this from its surrounding scope (global scope if top-level).</p> <pre>const arrow = () => console.log(this); arrow(); // In browser: window, in strict mode still window (lexical from global)</pre>
Inside an Arrow Function(as object method)	<p>Still inherits from the surrounding scope → usually not the object itself, but whatever this was outside the object.</p> <pre>const obj = { name: "Siva", greet: () => console.log(this.name) }; obj.greet(); // undefined (because `this` comes from global scope)</pre>
Inside an object created with the Constructor Function(new keyword)	<p>Refers to the newly created object instance.</p> <pre>function Person(name) { this.name = name; } const p1 = new Person("Maruthi"); console.log(p1.name); // "Maruthi"</pre>