observe

- SVG framework function: observe
  - Usage
  - Example
  - Notes


```
title: 'Iconify SVG Framework Function: observe'
functions:
  scan: './scan.md'
  stopObserving: './stop-observing.md'
  pauseObserver: './pause-observer.md'
  resumeObserver: './resume-observer.md'
  renderSVG: './render-svg.md'
  renderHTML: './render-html.md'
  loadIcons: './load-icons.md'
```

SVG framework function: observe

This tutorial is part of Iconify SVG Framework functions tutorial.

Function `[func]observe()` tells the SVG framework to watch an HTML element for icon placeholders, so the SVG framework could find them and replace with `[tag]svg`.

SVG framework automatically watches `[prop]document.body`, so no need to call this function for `[prop]documet.body`.

When to use this function:

- If you have removed observer for `[prop]document.body` using `[func]stopObserving(document.body)` (you should not do this, use `[func]pauseObserver()` instead to pause observer instead of removing observer).
- If you need to observe an element that is not part of `[prop]document.body`, such as Shadow DOM.

Usage

Function has the following parameter:

- `[prop]root`, `[type]HTMLElement`. Element to observe.

Example

```
// Create element.
const node = document.createElement('div');

// Tell SVG framework to watch element.
Iconify.observe(node);

// Add icon 1 second later - it will be replaced with <svg>
setTimeout(() => {
	node.innerHTML =
		'Home icon: <span class="iconify" data-icon="mdi:home"></span>';
}, 1000);
```

Another example:

```
// Create element, start watching it.
function createNode() {
	let clicked = false;

	// Create element, set content.
	const node = document.createElement('a');
	node.innerHTML =
		'<span class="iconify" data-icon="carbon:circle-dash"></span>';
```

```
// Change icon on click.
node.addEventListener('click', () => {
  clicked = !clicked;
  node
    .querySelector('.iconify')
    .setAttribute(
      'data-icon',
      clicked ? 'carbon:circle-filled' : 'carbon:circle-dash'
    );
});

// Observe changes.
// This is not needed if you are adding element to document.body because document.body is already
// automatically watched. Use this if element is part of Shadow DOM or some other custom elements tree.
Iconify.observe(node);

return node;
}

// Stop watching element.
// Always stop observing element if you no longer need it!
// SVG framework doesn't know if you still need an element.
function removeNode(node) {
Iconify.stopObserving(node);
}
```

Notes

- Icon placeholders are not replaced immediately. It might happen on the next tick or when icon data has been retrieved from API. Do not expect HTML to include `[tag]svg` tags right after you change element's content. If you want to find an icon in your root element, search by class name `[str]iconify` or by attribute `[attr]data-icon`, not by tag. If you want to have full control over rendering, see `[func]renderSVG()`, `[func]renderHTML()` and `[func]loadIcons()`.
- When you start observing a custom element, it is scanned for placeholders. No need to call `[func]scan()`, it is done automatically.

□□□ #2
□ □□□ □□□ 2026-10-31 14:20:35 UTC

□ □□□ □□□ 2026-10-31 14:20:36 UTC