Element: pseudo() method

- Syntax
  - Parameters
  - Return value
- Examples
  - Basic usage
    - HTML
    - CSS
    - JavaScript
    - Result
- Specifications
- Browser compatibility
- See also

.

The **`pseudo()`** method of the Element interface returns a CSSPseudoElement object representing the CSS pseudo-element of the specified type associated with the element.

Provided its `type` parameter contains a valid pseudo-element type, `pseudo()` will always return a `CSSPseudoElement` instance, even if that pseudo-element hasn't been generated on the calling element.

Syntax

```
pseudo(type)
```

Parameters

- `type`
  - : A string representing the type of pseudo-element to return a representation of. Valid values are:
    - ::after
    - ::before
    - ::marker

Return value

A CSSPseudoElement object instance, or `null` if `type` is not equal to a valid pseudo-element type.

Examples

Basic usage

In this example, we demonstrate basic usage of the `pseudo()` method.

HTML

We include a p element containing text, and an output element to log output from JavaScript.

```
<p>New York's hottest club is...</p>
<output></output>
```

CSS

We give the `<p>` element's ::after pseudo-element some content and apply some basic styles to both.

```
body {
  width: 80%;
  margin: 0 auto;
}
```

```
p {
  background-color: violet;
  padding: 20px;
```

```
}

p::after {
  content: "Crease";
  background-color: cadetblue;
  padding: 20px;
}
```

JavaScript

In our script, we grab references to our `<p>` and `<output>` elements, and retrieve a `CSSPseudoElement` representing the `<p>` element's `::after` pseudo-element via the `pseudo()` method. We then log some details of the pseudo-element to our `<output>` element. We also include some rudimentary error handling via a `try...catch` structure, to print an error message in non-supporting browsers.

```
const pElem = document.querySelector("p");
const output = document.querySelector("output");

try {
  const pseudoElem = pElem.pseudo("::after");
  output.textContent = `${pseudoElem.type} pseudo-element. Parent:
<${pseudoElem.parent.tagName.toLowerCase()}>`;
} catch (e) {
  output.textContent = `Your browser doesn't support CSSPseudoElement and/or the pseudo()
method: ${e}`;
}
```

Result

Specifications

Browser compatibility

See also

- CSSPseudoElement.pseudo()

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:34:32 UTC