Chapter 51: Custom Elements

Parameter	Details
name	The name of the new custom element.
options.extends	The name of the native element being extended, if any.
options.prototype	The custom prototype to use for the custom element, if any.

Section 51.1: Extending Native Elements

It's possible to extent native elements, but their descendants don't get to have their own tag names. Instead, the `is` attribute is used to specify which subclass an element is supposed to use. For example, here's an extension of the **`<img>`** element which logs a message to the console when it's loaded.

```
const prototype = Object.create(HTMLImageElement.prototype);
prototype.createdCallback = function() {
  this.addEventListener('load', event => {
      console.log("Image loaded successfully.");
  });
};

document.registerElement('ex-image', { extends: 'img', prototype: prototype });
```

```
<img is="ex-image" src="http://cdn.sstatic.net/Sites/stackoverflow/img/apple-touch-icon.png" />
```

Section 51.2: Registering New Elements

Defines an **`<initially-hidden>`** custom element which hides its contents until a specified number of seconds have elapsed.

```
const InitiallyHiddenElement = document.registerElement('initially-hidden', class extends
HTMLElement {
  createdCallback() {
    this.revealTimeoutId = null;
  }

  attachedCallback() {
    const seconds = Number(this.getAttribute('for'));
    this.style.display = 'none';
    this.revealTimeoutId = setTimeout(() => {
      this.style.display = 'block';
    }, seconds * 1000);
  }

  detachedCallback() {
    if (this.revealTimeoutId) {
      clearTimeout(this.revealTimeoutId);
      this.revealTimeoutId = null;
    }
  }
});
```

```
<initially-hidden for="2">Hello</initially-hidden>
<initially-hidden for="5">World</initially-hidden>
```