

# Conceptos clave

- Template literals
- Template tags
- Custom Elements
- HTML Templates
- Shadow DOM
- Light DOM
- Polyfill
- Lit-element / Lit-html

# Template literals

# Template literals

Multiline strings | Interpolation | Template tags

# Template tags

# Template tags

```
const interpolated = interpolate`I paid ${10}€`

function interpolate(literals, ...expressions) {
  let string = ``
  for (const [i, val] of expressions.entries()) {
    string += literals[i] + val
  }
  string += literals[literals.length - 1]
  return string
}
```

# Custom Elements

# Custom elements

Extender HTML(HTMLElement) con tus propios tags

```
<my-component datos="datos"></my-component>

class MyComponent extends HTMLElement {
  connectedCallback() {
    this.innerHTML = `<h1>Hello world</h1>`;
  }
}

customElements.define('my-component', MyComponent);
```

# HTML Templates



# HTML Templates

```
<template id="template">
  <script>
    const button = document.getElementById('click-me');
    button.addEventListener('click', event => alert(event))
  </script>
  <style>
    #click-me {
      background: tomato;
      .....
      padding: .5rem 1rem;
    }
  </style>
  <button id="click-me">Log click event</button>
</template>
```

# Shadow dom

# Shadow dom

- Versión encapsulada del DOM.
- Simplifica CSS: el DOM dentro del ámbito significa que puedes usar simples selectores de CSS, nombres de id/clase más genéricos, y no preocuparte por conflictos de nombres.
- <https://developers.google.com/web/fundamentals/web-components/shadowdom>
- `shadowRoot`, `attachShadow({ mode: 'open' })`

# Light Dom

# Light Dom

Para diferenciar del "shadow DOM", llamamos "light DOM" al DOM normal (en el árbol DOM principal)

```
<my-element>  
  <p>I won't render</p>  
</my-element>
```

# Slots

# Slots

La manera de componer unos web components con otros

# Polyfill



# Polyfill

A polyfill, or polyfiller, is a piece of code (or plugin) that provides the technology that you, the developer, expect the browser to provide natively. Flattening the API landscape if you will. Oct 8, 2010

# Lit-element

- Try lit-element
- Lit-element
- El coñazo de hacerlo desde cero
- Ejemplo completo de hacerlo desde cero
- Sin lit-element, no hay bindings.... @click, etc..
- Lit-element sigue las [best practices](#)

# Lit-html

# Lit-html

- Lit-html concepts

# Extra: el problema con las CSS

- CSS custom properties
- Constructible stylesheets
- CSS Modules
- :host
- *With the exception of **inherited CSS properties**, shadow roots are also isolated from styles defined outside the shadow root, whether in the main page or an outer shadow root.*
- CSS Inheritance and shadow dom

# Extra: Life-cycle callbacks

- `connectedCallback`
- lit-element lifecycle versus estandar web components
- `First updated`
- `Mozilla, custom elements lifecycle`