

Watch: 📺 **Strings más legibles con template literals**

08:42



Tienes el [código de este vídeo en GitHub](#).

Los template literals nos permiten interpolar valores en una cadena de texto de forma más legible:

```
"Thank you " + newUser.firstName + " for registering!";
```

```
`Thank you ${newUser.firstName} for registering!`;
```

Además, también nos permiten hacer strings multilínea.

Habrás visto también funciones que, en lugar de paréntesis y parámetros, usan un template string. Es habitual verlo en librerías populares como [Styled Components](#) o [graphql-tag](#):

```
const query = gql`
  {
    user(id: 5) {
      firstName
      lastName
    }
  }
`
```

Es útil porque al permitir saltos de línea, el código dentro del string sigue siendo legible, pero si vamos a [la implementación](#) podemos ver que la función no solo recibe una string como parámetro.

Los tagged templates son funciones que van a recibir los fragmentos de string en un array como primer parámetro, seguido de todos los valores que hayan sido interpolados.

En nuestro proyecto, podemos usar los tagged templates, siguiendo el ejemplo que nos propone [este artículo de SitePoint](#), para sanear el input antes de pintarlo como HTML. **Importante:** la validación que hacemos es demasiado simple para ser segura, no usar este snippet en producción:

```
function sanitize(strings, ...values) {
  var output = "";
  for (var index = 0; index < values.length; index++) {
    var valueString = values[index].toString();

    if (valueString.indexOf(">") !== -1) {
      valueString = "-";
    }

    output += strings[index] + valueString;
  }

  output += strings[index];
  return output;
}

function handleFormSuccess(form, newUser) {
  var thanksBlock = document.getElementById("thanks");
  var title = thanksBlock.querySelector("h3");
  var content = thanksBlock.querySelector("p");
```

```
title.innerHTML = sanitize`Thank you ${newUser.firstName}
for registering!`;
content.innerHTML = sanitize`We sent a confirmation email to
<strong>${newUser.email}</strong>`;

form.classList.add("hidden");
thanksBlock.classList.remove("hidden");
}
```

Enlaces relacionados:

- [GraphQL Tag](#)
- [SitePoint: Understanding ES6 template string](#)