Template Literals

String concatenation is hard. Take this code for example.

```
function makeGreeting(name, email, id) {
  return (
    "Hello, " +
    name +
    ". We've emailed you at " +
    email +
    '. Your user id is "' +
    id +
    '".'
  );
}
```

All we're trying to do is take three variables (name, email, and id) and create a sentence using them. Sadly, in order to do that, it's a balancing act between using the right quotations, + signs, and escaping (\) the right characters. This is the exact problem that Template Literals (also called Template Strings) was created to solve.

With Template Literals, instead of using single (') or double quotes ("), you use backticks (``) (located to the left of the 1 key if you're using a QWERTY keyboard ☺). Anywhere inside of your backticks where you have an expression (a piece of code that results in a single value like a variable or function invocation), you can wrap that expression in ${expression goes here}.

So using Template Literals, we can take the confusing makeGreeting function above and simplify it to look like this.

```
function makeGreeting(name, email, id) {
  return `Hello, ${name}. We've emailed you at ${email}. Your user id is
"${id}".`;
}
```

Much better. No more worrying about using the right quotations, + signs, and escaping the right characters. Not only is it easier to write, but it's also much easier to read. Now instead of having a makeGreeting function, say we wanted a makeGreetingTemplate function that returned us an HTML string that we could throw into the DOM. Without template strings, we'd have something like this.

```
function makeGreetingTemplate(name, email, id) {
  return (
    "<div>" +
    "<h1>Hello, " +
    name +
```

```
    ".</h1>" +
    "<p>We've emailed you at " +
    email +
    ". " +
    'Your user id is "' +
    id +
    '".</p>' +
    "</div>"
  );
}
```

Perfect, except for the fact that not only is it terribly hard to write, it's even harder to read. What's nice about ES6's Template Strings is they also support multi-line strings. That means, using Template Strings, we can rewrite makeGreetingTemplate to look like this.

```
function makeGreetingTemplate(name, email, id) {
  return `
    <div>
      <h1>Hello, ${name}</h1>
      <p>
        We've email you at ${email}.
        Your user id is "${id}".
      </p>
    </div>
  `;
}
```

I consider that an absolute win.