Template Literals

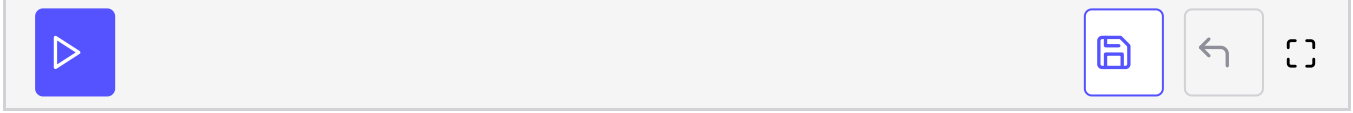
We'll cover JavaScript's new way of creating a string, the backticks (`). We'll see how these backticks make strings easier to create dynamically and how they make working with strings more intuitive.

These are fun and straightforward.

Using the new special character `, right above the tab key on the left side of the keyboard, we now have what are called template literals. We now have a way to visually format text exactly the way we want it. The way it's printed will be exactly how it looks.

Strings created with backticks (`) retain their formatting exactly as written. Here's an example.

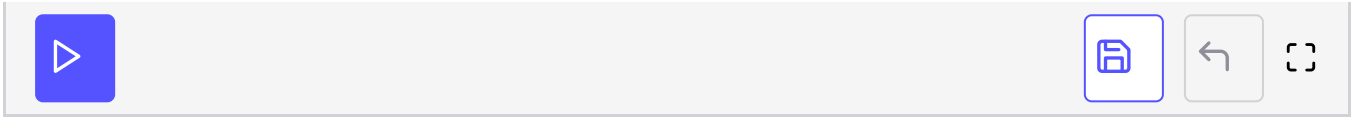
```
const str = `Hello there!
This new type of string keeps linebreaks in the string.
It also keeps all other types of whitespace.
A       tab is preserved correctly.
Multiple

linebreaks

are all

preserved

correctly.        Sweet.`;

console.log(str);
```



We also have a way of inserting variables into this string and having them automatically parsed. We can insert the characters `${}` into a template literal string. Anything inside the brackets will be evaluated as JavaScript and inserted into the final string.

```
const firstName = 'John';
```

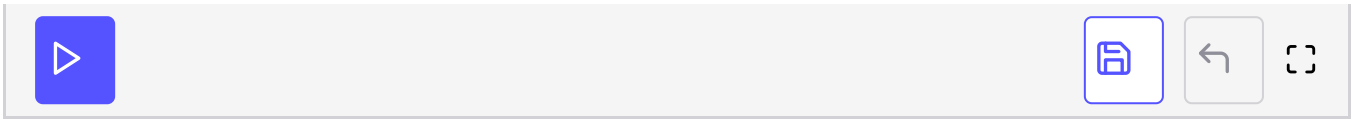
```
const lastName = 'Smith';

var oldWay = 'Hi! My name is ' + firstName + ' ' + lastName + '!';

const newWay = `Hi! My name is ${firstName} ${lastName}!`;

console.log(oldWay); // -> Hi! My name is John Smith!
console.log(newWay); // -> Hi! My name is John Smith!
```



The new way of creating this string is much easier to read and write. The two blocks of `${}` in the middle are easier for our minds to think about than using `+` symbols throughout the string.

That's really all there is to it. It makes our lives a little easier. Here's a complex example of the old way vs. the new way.

```
const numbers = [44, 29, 93];

var oldWay = "Today's lottery numbers are:\n\t- "
    + numbers[0]
    + "\n\t- " + numbers[1]
    + "\n\t- " + numbers[2];

const newWay = `Today's lottery numbers are:
        - ${numbers[0]}
        - ${numbers[1]}
        - ${numbers[2]}`;

console.log(oldWay);
console.log(newWay);
```



That's it for template literals.