

ES6: NEXT GEN JS

Peoria JS and Web Professionals

July, 2018

TEMPLATE STRINGS

Template Strings Example

Note: Template strings vastly simplify creating output from JS. Remember `document.write("The value of myVar is " + myVar + " and...")`; As you (and your students know, this is highly subject to errors. Don't forget we have single quotes and double quotes.

Template strings are here to help. Of course, we also now have a third set of quotes (the back tick ```). These are used in template strings/ template literals. Within the ```, we specify the variables using `${myVar}` – this tells the rendering engine to interpolate the variable contents. You can also do calculations and call functions within these curly braces. There is more good news as well – you can continue your text on multiple lines without worrying about stitching the fragments together (in the past, we often used `\` to accomplish this).

Code snippet: Pay attention to line 22. People often forget to close the string.

```
10 ▼ const myClass = {
11     className: 'ES6',
12     instructor: 'Mark DuBois',
13     city: "Springfield, IL",
14     dates: "May 21 - 25"
15 };
16 const myMarkup = `
17 <div>
18 <h1>${myClass.className}</h1>
19 <h2>${myClass.instructor}</h2>
20 <p>I participated in the 2018 Working Connections ${myClass.className}
    class. It was held in ${myClass.city} from ${myClass.dates}. It is so much
    easier to write JS these days.</p>
21 </div>
22 `;
23
24 document.getElementById('result').innerHTML = myMarkup;
```

Reference file: **01ExampleTemplateStrings.html**

You can also nest template strings. How cool is that?

Code snippet: Pay attention to the nesting on lines 20 – 21.

```
11 ▼ const students = [
12     {fullName: 'Grace Fox'},
13     {fullName: 'Caleb Castillo'},
14     {fullName: 'Jeffery Myers'},
15     {fullName: 'Elizabeth Pena'},
16     {fullName: 'Joshua Knight'}
17 ];
18 const myMarkup = `
19 <ul>
20   ${students.map(student => `<li>${student.fullName}</li>`).join('')}
21 </ul>
22 `;
23 // since map returns an array, we need the join method to remove commas
24
25 document.getElementById('result').innerHTML = myMarkup;
```

Reference file: **01x2ExampleTemplateStringsNested.html**

What if you receive information which may (or may not) contain certain content. For example, a student in a class may already have a degree. In this example, try removing the degree and observe the effect on the output.

Code snippet:

```
11 // what if we don't know whether student already has degree - try removing
    degree and observe effects with ternary operator
12 const students =
13   {fullName: 'Grace Fox',
14     currentClass: 'ES6',
15     degree: 'BS'
16   }
17 ;
18 const myMarkup = `
19   <p>
20     ${students.fullName} is taking the ${students.currentClass} class.
21     ${students.degree ? `Degree: ${students.degree}` : ''}
22   </p>
23 `;
24
25 document.getElementById('result').innerHTML = myMarkup;
```

Reference file: **01x3ExampleTemplateStringsTernary.html**