<template>: The Content Template element

The **HTML Content Template (`<template>`) element** is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.

Think of a template as a content fragment that is being stored for subsequent use in the document. While the parser does process the contents of the **`<template>`** element while loading the page, it does so only to ensure that those contents are valid; the element's contents are not rendered, however.

Content categories	Metadata content, flow content, phrasing content, script-supporting element
Permitted content	No restrictions
Tag omission	None, both the starting and ending tag are mandatory.
Permitted parents	`<body>`, `<frameset>`, `<head>`, `<dl>` and `<colgroup>` without a `span` attribute
Permitted ARIA roles	None
DOM interface	`HTMLTemplateElement`

Attributes

This element only includes the global attributes.

Example

First we start with the HTML portion of the example.

```html
<table id="producttable">
  <thead>
    <tr>
      <td>UPC_Code</td>
      <td>Product_Name</td>
    </tr>
  </thead>
  <tbody>
    <!-- existing data could optionally be included here -->
  </tbody>
</table>

<template id="productrow">
  <tr>
    <td class="record"></td>
    <td></td>
  </tr>
</template>
```

First, we have a table into which we will later insert content using JavaScript code. Then comes the template, which describes the structure of an HTML fragment representing a single table row.

Now that the table has been created and the template defined, we use JavaScript to insert rows into the table, with each row being constructed using the template as its basis.

```
// Test to see if the browser supports the HTML template element by c
// for the presence of the template element's content attribute.
if ('content' in document.createElement('template')) {

    // Instantiate the table with the existing HTML tbody
    // and the row with the template
    var t = document.querySelector('#productrow');

    // Clone the new row and insert it into the table
    var tb = document.querySelector("tbody");
    var clone = document.importNode(t.content, true);
    td = clone.querySelectorAll("td");
    td[0].textContent = "1235646565";
    td[1].textContent = "Stuff";

    tb.appendChild(clone);

    // Clone the new row and insert it into the table
    var clone2 = document.importNode(t.content, true);
    td = clone2.querySelectorAll("td");
    td[0].textContent = "0384928528";
    td[1].textContent = "Acme Kidney Beans 2";

    tb.appendChild(clone2);

} else {
  // Find another way to add the rows to the table because
  // the HTML template element is not supported.
}
```

The result is the original HTML table, with two new rows appended to it via JavaScript:

UPC_Code	Product_Name
1235646565	Stuff
0384928528	Acme Kidney Beans 2

Specifications

Specification	Status	Comment
HTML Living Standard The definition of 'template element' in that specification.	LS Living Standard	
HTML5 The definition of 'template element' in that specification.	REC Recommendation	Initial definition

Browser compatibility

Update compatibility data on GitHub

Basic support	
Chrome	26
Edge	13
Firefox	22
IE	No
Opera	15
Safari	8
WebView Android	Yes

Chrome Android	26
Edge Mobile	Yes
Firefox Android	22
Opera Android	?
Safari iOS	8
Samsung Internet Android	Yes

□ Full support

□ No support

□ Compatibility unknown

See also

- Web components: `<slot>` (and historical: `<shadow>`)
- Using templates and slots