

Chapter 34: Data attributes

Section 34.1: Accessing data attributes

Using the dataset property

The new dataset property allows access (for both reading and writing) to all data attributes `data-*` on any element.

```
<p>Countries:</p>
<ul>
  <li id="C1" onclick="showDetails(this)" data-id="US" data-dial-code="1">USA</li>
  <li id="C2" onclick="showDetails(this)" data-id="CA" data-dial-code="1">Canada</li>
  <li id="C3" onclick="showDetails(this)" data-id="FF" data-dial-code="3">France</li>
</ul>
<button type="button" onclick="correctDetails()">Correct Country Details</button>
<script>
function showDetails(item) {
    var msg = item.innerHTML
        + "\r\nISO ID: " + item.dataset.id
        + "\r\nDial Code: " + item.dataset.dialCode;
    alert(msg);
}

function correctDetails(item) {
    var item = document.getElementById("C3");
    item.dataset.id = "FR";
    item.dataset.dialCode = "33";
}
</script>
```

Note: The dataset property is only supported in modern browsers and it's slightly slower than the `getAttribute` and `setAttribute` methods which are supported by all browsers.

Using the `getAttribute` & `setAttribute` methods

If you want to support the older browsers before HTML5, you can use the `getAttribute` and `setAttribute` methods which are used to access any attribute including the data attributes. The two functions in the example above can be written this way:

```
<script>
function showDetails(item) {
    var msg = item.innerHTML
        + "\r\nISO ID: " + item.getAttribute("data-id")
        + "\r\nDial Code: " + item.getAttribute("data-dial-code");
    alert(msg);
}

function correctDetails(item) {
    var item = document.getElementById("C3");
    item.setAttribute("id", "FR");
    item.setAttribute("data-dial-code", "33");
}
</script>
```