

# HTML attributes

- When the browser loads the page, it parses the HTML text and generates DOM objects from it. For element nodes most standard HTML attributes automatically become properties of DOM objects.
- For instance, if the tag is **<body id="page">**, then the DOM object has **body.id="page"**
- HTML attributes are case insensitive (id is same as ID) and their value are always strings

# Non-standard attributes, dataset

- Sometimes we wish to pass custom data from HTML to JavaScript
- The data-\* attributes allow us to store extra info on HTML elements

```
<div id="studentOne" data-name="John Smith" data-age="23"></div>
```

```
var student = document.getElementById('studentOne');  
console.log('name = ', student.dataset.name);    // "John Smith"  
console.log('age = ', student.dataset.age);      // "23"
```

# HTML <script> tag

- The HTML <script> tag is used to embed or reference Javascript code
- Question: Should <script> be in <head> tag OR at the bottom of the <body> OR anywhere in the HTML?
- Most javascript developer wait for the **DOMContentLoaded** or **load** event before modifying the DOM
- The modern browsers support the **async** and **defer** attributes on the <script> tag
- defer scripts are only executed after the entire document has been loaded
- <http://www.growingwiththeweb.com/2014/02/async-vs-defer-attributes.html>

# Example

```
<!DOCTYPE html>
<html>
<head>
  <script src="index.js" defer></script>
</head>
<body>
<div>
  <header>Javascript events</header>
  <main>
    <h4>Add numbers:</h4>
    <p>
      <input id="numOne"> + <input id="numTwo">
    </p>
    <p id="result"></p>
  </main>
</div>
</body>
</html>
```

```
// index.js

var numOne = document.getElementById('numOne');
var numTwo = document.getElementById('numTwo');
var result = document.getElementById('result');

numOne.addEventListener('input', add);
numTwo.addEventListener('input', add);

function add() {
  var one = parseInt(numOne.value) || 0;
  var two = parseInt(numTwo.value) || 0;
  result.innerHTML = `result: ${one+two}`;
}
```