The Learning House

427 S 4th Street #300

Louisville KY 40202

JavaScript in HTML

.NET Cohort

Coding Bootcamp

SOFTWARE GUILD

Lesson Goals

1. Understand the <script> element
2. Learn how to reference external JavaScript files

The <script> Element

To put JavaScript directly on an HTML page, we use the <script> tag:

```
<script type="text/javascript">
    function sayHi() {
        alert('hello world');
    }
</script>
```

Many developers put type="text/javascript" but it is the default, so technically it is not necessary.

JavaScript code is interpreted from top to bottom, so keep in mind when using JavaScript that if there are dependencies on other script files (like jQuery), you need to reference those files first.

We can also put our JavaScript in a separate file and provide a path to the file:

```
<script type="text/javascript" src="~/Scripts/jquery.validate.js"></script>
```

Script Placement

Traditionally, tags were placed between the <head> tags on a web page, which kept linked files together. However, because script files must be downloaded and parsed immediately when hit, many developers have taken to placing script files at the end of the body tag so the user receives visual feedback more quickly.

In MVC 4, it's typical to create a Scripts section in the layout page at the bottom of the body and link in scripts there, in the views.

```
        @Scripts.Render("~/bundles/jquery")
        @RenderSection("scripts", required: false)
    </body>
</html>
```

```
@section scripts
{
    <script type="text/javascript">
        function sayHi() {
            alert('hello world');
        }
    </script>
}
```

SOFTWARE GUILD

Best Practice

It's considered best practice to include as much JavaScript as possible in external files. Doing so grants some benefits:

1. Maintainability: Putting JavaScript in files in a scripts folder helps future developers find it there, rather than having to hunt through pages. This also makes it easily reusable.
2. Caching: Browsers cache externally-linked files, so your external file will only be downloaded once, improving performance.

<noscript>

There is a very small percentage of users who turn off JavaScript in their browser. The <noscript> element exists so that we can display only HTML to those users. So, if you have a site that is heavily dependent on JavaScript, you can notify the user that it will not work well (or at all, sometimes).

```
<noscript>This page requires JavaScript enabled</noscript>
```

If the browser supports scripting, the contents of the tag will not be shown.

Conclusion

Not much to it! Where possible, try to keep your scripts at the page bottom.