

# Introduction to HTML 5 with Centricity

Presented by

Mike Baeske

Software Developer - Team Leader

Ortho NorthEast (ONE)

# Objectives

- ▶ Discover the structure of an HTML 5 web page including various HTML 5 tags to build the content.
- ▶ See how to style the website using CSS.
- ▶ Learn how to deploy HTML content into Centricity.

# Ortho NorthEast (ONE)

- ▶ An Orthopedic practice founded in 1962
- ▶ 11 Locations throughout northeast Indiana
- ▶ 88 Providers
- ▶ 37 Physicians
- ▶ 35 Advanced Practitioners
- ▶ 16 CRNAs

# HTML Editors

## Licensed

- ▶ Clinical Forge
- ▶ Microsoft Visual Studio

## In Development

- ▶ VFE
- ▶ Bounce

## Free

- ▶ Notepad
- ▶ Notepad++
- ▶ **Visual Studio Code**

# Elements of a Webpage

- ▶ Hypertext Markup Language (HTML)
  - ▶ What is displayed on the webpage
- ▶ Cascading Style Sheets (CSS)
  - ▶ How does the webpage look
- ▶ Script
  - ▶ What does the webpage do

# Hypertext Markup Language (HTML)

- ▶ Builds the structure of the web page
- ▶ Elements of a page are built using tags:  
`<html></html>`
- ▶ Most consist of a `<start tag>` content `<end tag>` structure
- ▶ Example:  
`<p>I wrote a paragraph</p>`

# HTML Fundamentals

- ▶ Tags can be nested in other tags:

```
<html>
```

```
  <body>
```

```
    <p>I am three tags deep.</p>
```

```
  </body>
```

```
</html>
```

# HTML Fundamentals

- ▶ Tags can have attributes that impact the behavior
- ▶ Style attribute can change the look of a tag inline:
  - ▶ `<p style="color:green">Green text</p>`

Green text

- ▶ Title attribute creates hover over text for a tag:

This is a header tag with hover over hints.

**Header text**



# Creating an HTML Webpage

- ▶ Start page with `<!DOCTYPE html>` to tell browsers it is HTML
- ▶ `<html></html>` should surround all other components
- ▶ `<head></head>` should contain setup object not being displayed
- ▶ `<body></body>` contains objects being rendered to the user
- ▶ Save the file with `.html` extension: `index.html`

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/css/default.css" />
    <script src="../../common/angularjs/1.3.13/angular.min.js"></script>
  </head>

  <body>
    <h3>Header text</h3>
    <p>Paragraph text</p>
  </body>
</html>
```

**Header text**

Paragraph text

# Key Elements

- ▶ `<input>`
  - ▶ `type = 'text', 'checkbox', 'radio', 'date'`
- ▶ `<textarea>` - multiline edit box
- ▶ `<img>` - displays an image
- ▶ `<button>` - executes coding (script)
- ▶ `<a>` - hyperlinks to external objects
- ▶ `<table>` - creates a table layout
  - ▶ `<tr>` - establishes a table row
  - ▶ `<th>` - table header
  - ▶ `<td>` - table cell

# Demonstration

# Cascading Style Sheets (CSS)

- ▶ Formats elements with a different look
- ▶ Can standardize the look of your forms by using the same style sheet
- ▶ Updating the look of all your content can easily be adjusted if one default style sheet is used
- ▶ CSS file connected with HTML by link tag:  
`<link rel="stylesheet" href="css/default.css" />`

# CSS Syntax

- ▶ Selector {property1:value1; property2:value2;...}
- ▶ Example:p {color: red;}

```
h3 {  
  color: green;  
  text-align: center;  
  text-decoration: underline;  
}
```

**Header text**

**Paragraph text**

# CSS Class Selector

- ▶ HTML Class attribute can be used to target specific elements

```
.hpiHeader {  
  background-color:lightblue;  
}
```

```
.peHeader {  
  background-color:lightgreen;  
}
```

```
.imagingHeader {  
  background-color:lightsalmon;  
}
```

```
.impressionHeader {  
  background-color:lightslategrey;  
}
```

```
.planHeader {  
  background-color:lightseagreen;  
}
```

```
<h2 class='hpiHeader'>HPI</h2>
```

```
<h2 class='peHeader'>PE</h2>
```

```
<h2 class='imagingHeader'>IMAGING STUDIES</h2>
```

```
<h2 class='impressionHeader'>IMPRESSION</h2>
```

```
<h2 class='planHeader'>PLAN</h2>
```

**HPI**

**PE**

**IMAGING STUDIES**

**IMPRESSION**

**PLAN**

# Demonstration

# Script

- ▶ Automated actions performed by a webpage are controlled by a script tag
- ▶ Different scripting languages can be used:
  - ▶ JavaScript
  - ▶ TypeScript
  - ▶ VBscript



# Javascript Example

```
▶ <script>  
▶     function popup(){  
▶         window.alert("Doctors love me, I am a  
pop up!!!")  
▶     }  
▶ </script>
```

# Centricity HTML Example Content

- ▶ Built using Javascript and AngularJS
- ▶ common folder contains libraries including, centricityemr:

```
<script src="../../common/angularjs/1.3.13/angular.min.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-module-1.1.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-service-1.1.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-controller-1.1.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-filter-1.1.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-directive-1.1.js"></script>  
<script src="../../common/centricityemr/1.1/centricityemr-exceptionHandler-1.1.js"></script>
```

# Integrating HTML Content in Centricity

- ▶ HTML Files must be placed on Jboss server:

C:\Program Files\Centricity Practice  
Solution\jboss\standalone\deployments

\<YOUR\_DATABASE\_NAME>.ear\CentricityPracticeWS.war\Enc  
ounterForms

- ▶ Sharing the EncounterForms folder will make your developer lives easier

dev-cps-beta > EncounterForms >

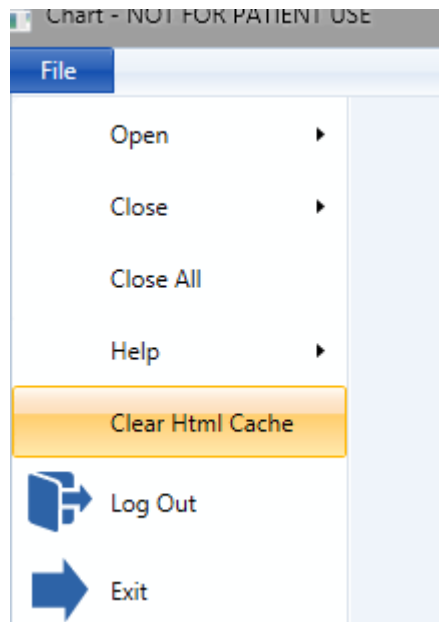
Name
CHUGDemoForm

# Centricity Integration

- ▶ SHOW\_HTML\_FORM() MEL symbol can pull form once folder structure is on Jboss server
- ▶ {SHOW\_HTML\_FORM("//localhost/EncounterForms/CHUGDemoForm/index.html", "CHUGDemoForm")}
- ▶ Opens form as a pop up page
- ▶ Clinical Kit required for embedded version:  
[HTMLEFS][1][CHUG Demo Form][Enterprise][CHUG Demo Form Notes][CHUG Demo Form]  
[//localhost/EncounterForms/CHUGDemoForm/index.html][true]
- ▶ MEL symbols for adding forms can be used after import

# Refreshing Content

- ▶ Cache needs to be cleared to see new version of HTML content, if already open



# Online Resources

- ▶ [W3schools.com](https://www.w3schools.com)
- ▶ [Stackoverflow.com](https://stackoverflow.com)
- ▶ <https://docs.angularjs.org/api>
- ▶ <https://code.visualstudio.com/>

The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. The shapes are primarily triangles and polygons, creating a dynamic, layered effect. The overall composition is clean and modern.

# Questions?

Mike Baeske  
mbaeske@orthone.com