

# **Bootstrapping DITA**

## ***Customizing HTML output for modern web frameworks***

**Roger Sheen @infotexture**

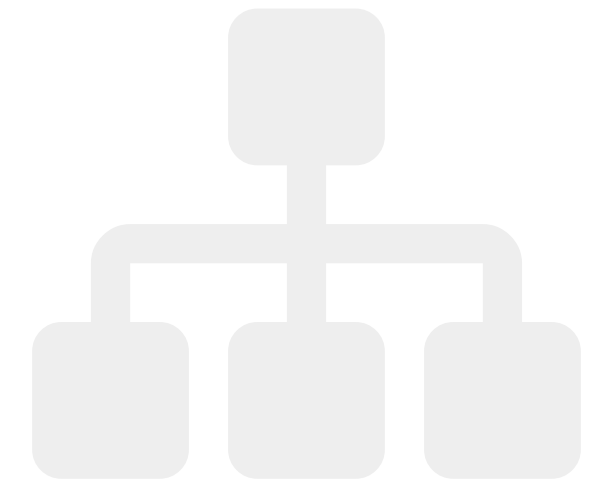


Web developers often use CSS frameworks, HTML5 boilerplate or component libraries like Bootstrap or Foundation to quickly build robust, responsive sites.



With custom HTML plug-ins, DITA-OT can produce HTML5 output that uses these templates so that generated documents can build on existing front-end solutions.

# Agenda



- DITA-OT website
- DITA-OT site plug-in
- Adding CSS & metadata
- Adding menu & classes
- Styling navigation

# DITA-OT website

- Published via [GitHub pages](#) to [dita-ot.org](http://dita-ot.org).
- The website is maintained in DITA, [Markdown](#) and HTML, versioned in Git and updated by pushing commits to [github.com/dita-ot/dita-ot.github.io](https://github.com/dita-ot/dita-ot.github.io).
- GitHub pages is powered by [Jekyll](#), an open source tool like DITA-OT that transforms files in one format with variables and templates, and generates output.

# DITA-OT site plug-in

The DITA-OT site plug-in is available on GitHub:

<https://github.com/dita-ot/org.dita-ot.html>

## Why not just use this?

- Contains code specific to the DITA-OT project
- It requires Jekyll
- Based on older HTML5 code

***So let's just start from scratch...***

# Adding CSS & metadata

Customizing the `chapterHead` template mode:

```
<xsl:template match="*" mode="chapterHead">
  <head>
    <!-- initial meta information -->
    <xsl:call-template name="generateCharset"/> <!-- Set the character set to UTF-8 -->
    <!-- Add <meta> elements from basic Bootstrap template -->
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <!-- Continue with DITA-OT defaults -->
    <!-- [...] -->
    <!-- Bootstrap core CSS -->
    <link href="/css/bootstrap.min.css" rel="stylesheet"/>
    <!-- Continue with DITA-OT defaults -->
  </head>
</xsl:template>
```



# Adding menu & classes

Customizing the `chapterBody` template mode:

```
<xsl:template match="*" mode="chapterBody">
  <body>
    <!-- Here there be defaults... -->
    <div class="navbar navbar-default navbar-static-top" role="navigation">
      <div class="container">
        <div class="navbar-header">
          <!-- Your menu here... -->
        </div>
      </div>
    </div>
    <div>
    </div>
    <div>
    </div>
    <main id="content" class="col-md-9 container" role="main">
      <xsl:apply-templates select="." mode="addContentToHtmlBodyElement"/>
    </main>
    <xsl:apply-templates select="." mode="addFooterToHtmlBodyElement"/>
  </body>
</xsl:template>
```

# Styling navigation

Customizing the `gen-user-sidetoc` template mode:

```
<xsl:template match="*" mode="gen-user-sidetoc">
  <xsl:if test="$nav-toc = ('partial', 'full')">
    <nav class="col-md-3" role="toc">
      <div class="well well-sm">
        <ul class="bs-docs-sidenav">
          <xsl:choose>
            <!-- Here there be defaults... -->
          </xsl:choose>
        </ul>
      </div>
    </nav>
  </xsl:if>
</xsl:template>
```