

Navigation

In this lesson, we learn all about navigation in HTML.
Let's begin!

WE'LL COVER THE FOLLOWING



- **Listing-03-09:** Adding quick links



`<nav>` in HTML5



You may have many links placed in a web page with the `<a>` tag. Most websites contain a set of navigation links that are the essential links within the site or page. With the help of the `<nav>` semantic element, you can place these links into a specific section. A page can have more than one set of navigation links. Quick links are often put into an `<aside>` section, as the page skeleton shown in the image below:

Using Figures

x6jr4kg.educative.run

Visual Studio Platform and Extensibility

by Istvan Novak

When to use macros, add-ins and packages?

After reading a short overview of the extensibility artifacts, a natural question arises: which one should I use for a certain extension task?

Macros are quite limited from a functionality point of view, because just a small part of the Visual Studio features can be accessed by using them and you cannot hide the macro source code. However, when this is not an issue and you want to provide an extension just to automate simple repetitive tasks, macros are your best friends. If macros constrain you because you want to add more than they offer you, add-ins and packages will be your saviors.

Few extensibility options are not available through automation model

Add-in development offers a richer set of tools than macros. Besides using the Visual Studio automation model you can extend the user interface easily and use the services provided by other add-ins and packages. Because your code is compiled into a .NET assembly you can use the same deployment and intellectual property-defending methods as for any .NET binary. However there are a few extensibility options that are not available through the automation object model or through standard Visual Studio IDE services. In this case you cannot use an add-in, you must create a VSPackage.

"...macros are your best friends."

text added using the <aside> tag

Full article published in CODE Magazine in April, 2008.

<aside> tags styled to sidebars

Listing-03-09: Adding quick links

```
<!DOCTYPE html>
<html>
<head>
  <title>Using Figures</title>
  <style>
    body {
      width: 720px;
      margin-left: 16px;
      font-family: Verdana, Arial, sans-serif;
    }

    p {
      text-align: justify;
      margin-top: 4px;
      margin-bottom: 4px;
    }

    header {
      background-color: deepskyblue;
      padding: 2px 16px;
    }
  </style>
</head>
<body>
  <div>
    <h2>Visual Studio Platform and Extensibility</h2>
    <p>by Istvan Novak</p>
    <h3>When to use macros, add-ins and packages?</h3>
    <p>After reading a short overview of the extensibility artifacts, a natural question arises: which one should I use for a certain extension task?</p>
    <p>Macros are quite limited from a functionality point of view, because just a small part of the Visual Studio features can be accessed by using them and you cannot hide the macro source code. However, when this is not an issue and you want to provide an extension just to automate simple repetitive tasks, macros are your best friends. If macros constrain you because you want to add more than they offer you, add-ins and packages will be your saviors.</p>
    <div>
      <p><i>Few extensibility options are not available through automation model</i></p>
      <p>Add-in development offers a richer set of tools than macros. Besides using the Visual Studio automation model you can extend the user interface easily and use the services provided by other add-ins and packages. Because your code is compiled into a .NET assembly you can use the same deployment and intellectual property-defending methods as for any .NET binary. However there are a few extensibility options that are not available through the automation object model or through standard Visual Studio IDE services. In this case you cannot use an add-in, you must create a VSPackage.</p>
    </div>
    <p><i>"...macros are your best friends."</i></p>
  </div>
  <div>
    Full article published in CODE Magazine in April, 2008.
  </div>
  <div>
    <aside> tags styled to sidebars
  </div>
</body>
</html>
```

```
h1 {
  color: white;
}

h2 {
  color: navy;
  border-bottom: 4px dotted cornflowerblue;
}

h3 {
  color: navy;
  margin-top: 1em;
  margin-bottom: 0px;
}

.byLine {
  color: white;
  font-style: italic;
}

.mainContent {
  background-color: aliceblue;
  padding: 4px 16px;
}

footer {
  background-color: cornflowerblue;
  padding: 1px 16px;
}

  footer > p {
    color: white;
    font-size: 0.8em;
  }

aside {
  float: left;
  margin: 0 16px 4px 0;
  padding: 8px 16px 0 16px;
  background-color: cornflowerblue;
  font-size: 0.9em;
}

aside>h3 {
  margin-top: 0;
  color: white;
}

ul {
  margin-left: 0px;
  padding-left: 0;
}

li {
  list-style: none;
  margin-left: 0px;
  margin-bottom: 2px;
}

li>a {
  text-decoration: none;
```

```

        color: white;
    }

    li>a:hover {
        color: lightgray;
    }

    .hilited {
        font-size: 0.9em;
        font-style: italic;
        color: dimgray;
    }

</style>

</head>
<body>
    <article>
        <header>
            <h1>HTML5 Semantic Elements</h1>
        </header>
        <div class="mainContent">
            <section>
                <h2>What Semantic Elements Are</h2>
                <aside>
                    <h3>Quick Links</h3>
                    <nav>
                        <ul>
                            <li><a href="#article">&lt;article&gt;</a></li>
                            <li><a href="#aside">&lt;aside&gt;</a></li>
                            <li><a href="#figure">&lt;figure&gt;</a></li>
                            <li><a href="#figcaption">&lt;figcaption&gt;</a></li>
                            <li><a href="#footer">&lt;footer&gt;</a></li>
                            <li><a href="#header">&lt;header&gt;</a></li>
                            <li><a href="#hgroup">&lt;hgroup&gt;</a></li>
                            <li><a href="#nav">&lt;nav&gt;</a></li>
                            <li><a href="#section">&lt;section&gt;</a></li>
                        </ul>
                    </nav>
                </aside>
                <p class="hilited">
                    To improve the structure of web pages, HTML5 adds
                    new semantic elements to the markup specification.
                    These elements give extra meaning to the content they enclose.
                </p>
            </section>
            <section>
                <h3 id="article">&lt;article&gt;</h3>
                <p>
                    This element defines an independent, self-contained content,
                    such as a blog entry, a newspaper article, a forum post,
                    a CV, an author biography, a story, etc.
                    &ndash;anything that you think of as an article.
                </p>
            </section>
            <section>
                <h3 id="aside">&lt;aside&gt;</h3>
                <p>
                    This element defines content that is separate from the other
                    (surrounding) content of the page&ndash;aside from the content
                    it is placed in. It is frequently used to create sidebars
                    related to an article.
                </p>
            </section>
        </div>
    </article>
</body>
</html>

```

```

</p>
</section>
<section>
  <h3 id="figure">&lt;figure&gt;</h3>
  <p>
    Represents a figure that is&ndash;in contrast to traditional
    images&ndash;a self-contained content, such as an illustration,
    diagram, photo, etc. The &lt;figure&gt; element is a wrapper
    for this content, including the &lt;img&gt; for the figure,
    as well as the caption nested into a &lt;figcaption&gt; element.
    The aim is to indicate the relation between the image and its
    associated caption.
  </p>
</section>
<section>
  <h3 id="figcaption">&lt;figcaption&gt;</h3>
  <p>
    This element defines a caption for the &lt;figure&gt; element.
  </p>
</section>
<section>
  <h3 id="footer">&lt;footer&gt;</h3>
  <p>
    Defines a footer for a document or section, so this element
    should contain information about its container element.
    It can be a set of important link, a copyright notice,
    terms of use, contact information, etc.
  </p>
</section>
<section>
  <h3 id="header">&lt;header&gt;</h3>
  <p>
    This element represents an enhanced heading for a document
    or a section. It should be a container for introductory
    content, and may contain logo, byline, set of navigational
    links, etc.
  </p>
</section>
<section>
  <h3 id="hgroup">&lt;hgroup&gt;</h3>
  <p>
    This element defines an enhanced heading that groups two
    or more heading elements without any additional content.
    Its purpose is to make a title and a subtitle (or subtitles)
    stand together.
  </p>
</section>
<section>
  <h3 id="nav">&lt;nav&gt;</h3>
  <p>
    Defines a major block of links on a page. These links
    may point to topics on the current page, or to other
    pages on the website. Not all links of a document must
    be in a &lt;nav&gt; element. A page may have multiple
    &lt;nav&gt; sections.
  </p>
</section>
<section>
  <h3 id="section">&lt;section&gt;</h3>
  <p>
    This element defines logical sections in a document.
    These sections can be headers, footers, chapters,

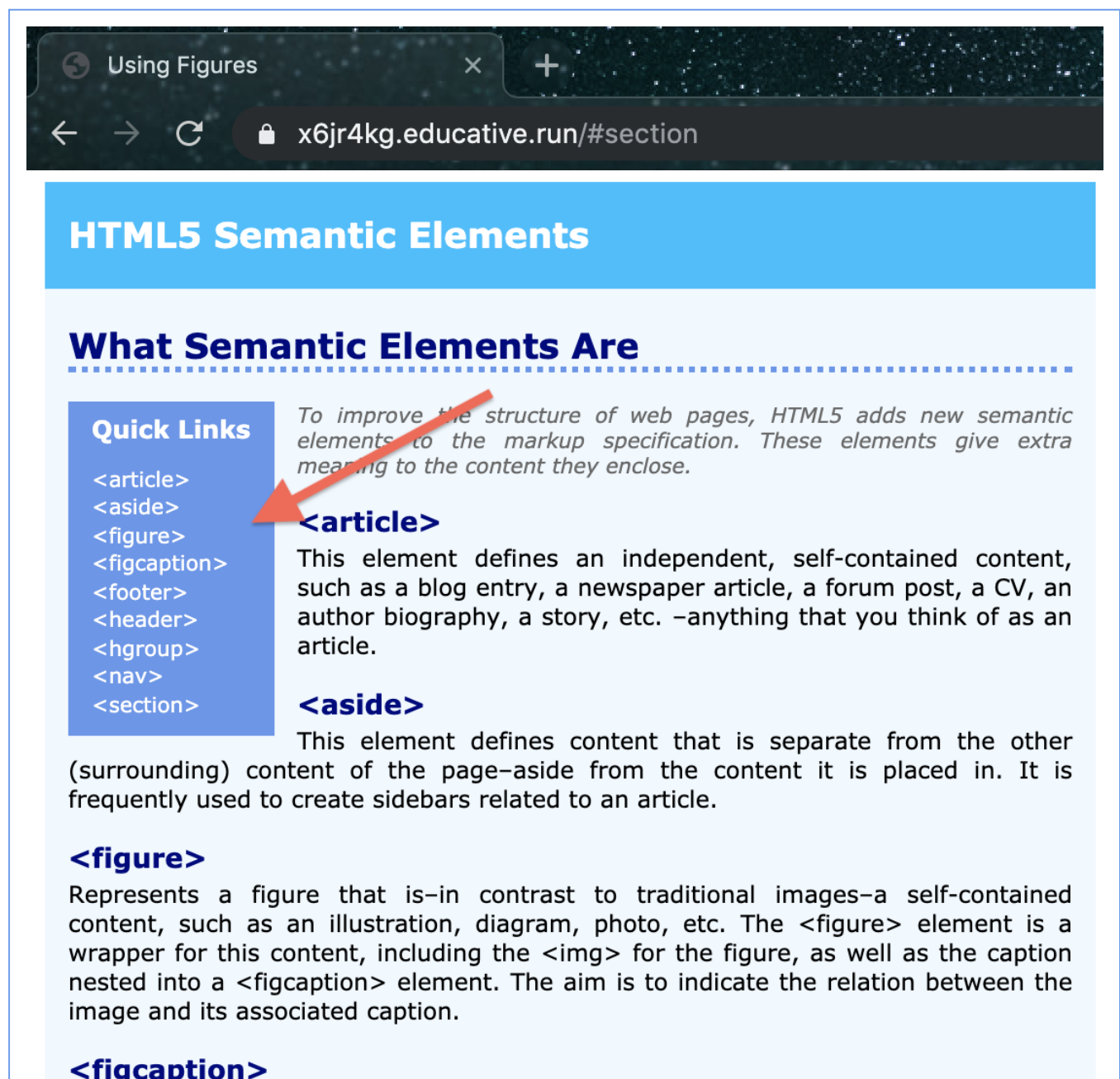
```

```

        sections of chapters, etc. Use <section>
        only if other semantic elements do not apply.
        As a rule of thumb, the content <section>
        holds always should begin with a heading
        (<h1>, ..., <h6>);
    </p>
</section>
</div>
</article>
<footer>
    <p>
        Full article published in CODE Magazine
        in April, 2008.
    </p>
</footer>
</body>
</html>

```

As you can see, the `<nav>` element contains only hyperlinks embedded into an unordered list. With simple styling, this markup is rendered as shown in the image below:



This element defines a caption for the <figure> element.

Adding navigation links to the page with <nav>

Achievement unlocked! 🎉

Congratulations! You've learned how to make use of the navigation tag in HTML.

All around amazing work!

Give yourself a round of applause! :)

In the *next lesson*, we'll uncover some more semantic elements.

