Sidebars

In this lesson, we'll learn how to use the aside tag in HTML.
Let's begin!

WE'LL COVER THE FOLLOWING

Your web page may have content that is only tangentially related to the main flow of the content and can be positioned anywhere near the related content, such as a sidebar. HTML5 provides the `<aside>` semantic tag to define this "tangentially related" section.

You can put anything in `<aside>` that you would put into a sidebar, including text, an image, a set of navigation links, a header, a footer, and so on.

Listing-03-08 shows you the skeleton of a web page that contains two small sidebars.

Listing-03-08: Using the `<aside>` tag

```
<!DOCTYPE html>
```

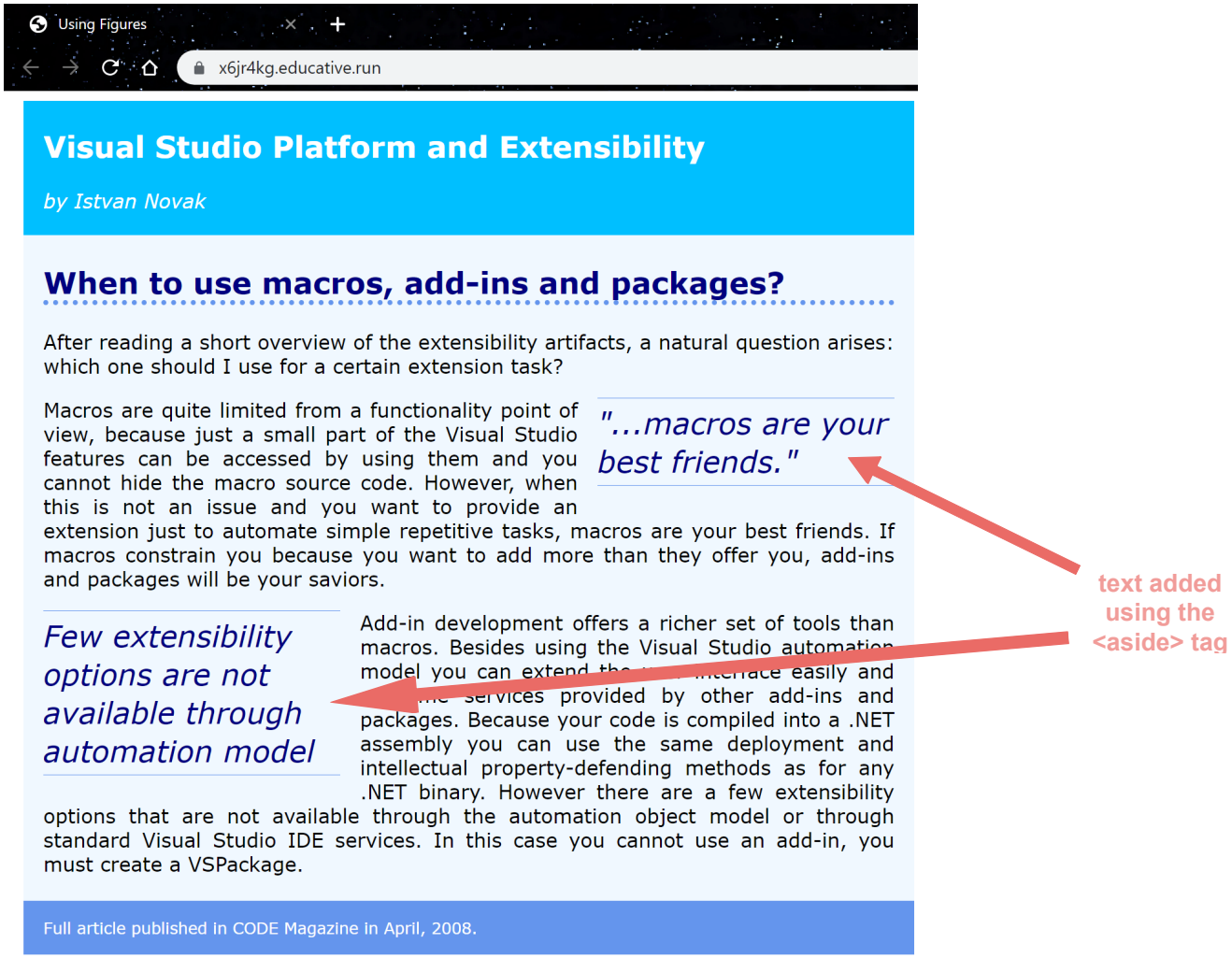
```
<html>
<head>
  <title>Using Figures</title>
  <style>
    body {
      width: 720px;
      margin-left: 16px;
      font-family: Verdana, Arial, sans-serif;
    }

    p {
      text-align: justify;
    }

    header {
      background-color: deepskyblue;
      padding: 2px 16px;
    }

    h1 {
      color: white;
    }

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

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

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

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

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

    aside {
      max-width: 240px;
      border-top: thin cornflowerblue solid;
      border-bottom: thin cornflowerblue solid;
      font-size: 1.5em;
      line-height: 130%;
      font-style: italic;
      padding-top: 4px;
      padding-bottom: 4px;
      margin-bottom: 8px;
      color: navy;
    }

    .leftBar {
```

```
      float: left;
      margin-right: 16px;
    }

    .rightBar {
      float: right;
      margin-left: 16px;
    }
  </style>

</head>
<body>
  <article>
    <header>
      <h1>Visual Studio Platform and Extensibility</h1>
      <p class="byLine">by Istvan Novak</p>
    </header>
    <div class="mainContent">
      <section>
        <h2>When to use macros, add-ins and packages?</h2>
        <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>
        <aside class="rightBar">
          &amp;quot;...macros are your best friends.&amp;quot;
        </aside>
        <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>
        <aside class="leftBar">
          Few extensibility options are not available
          through automation model
        </aside>
        <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
          consume 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>
      </section>
    </div>
  </article>
  <footer>
    <p>
```

```
      Full article published in CODE Magazine
      in April, 2008.
    </p>
  </footer>
</body>
</html>
```

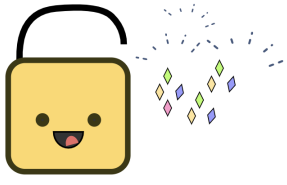
This exercise shows that the sidebars belong to the section describing the main content.

With a few style rules (one for the `<aside>` tag, and two others for the `leftBar` and `rightBar` classes) you can transform the `<aside>` sections into real sidebars, as shown in the figure below:



<aside> tags styled to sidebars

Achievement unlocked!



Congratulations! You've learned to use the `<aside>` tag in HTML!

Amazing work! Give yourself a round of applause!

In the *next lesson*, we'll learn all about navigation!

Stay tuned! :)