Figures in the Page

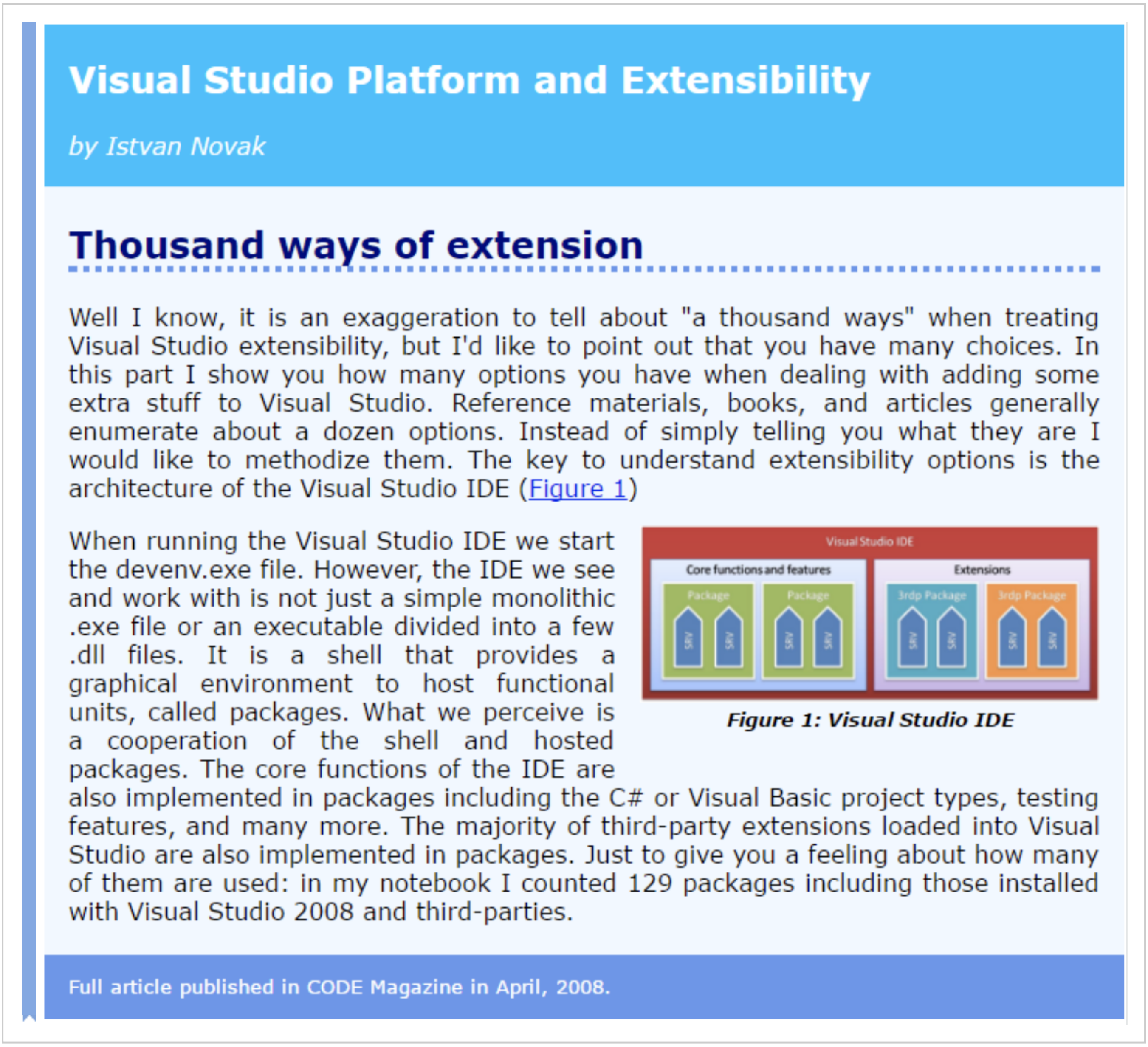
Let's study the semantics of figures and how to add them in our HTML page.

WE'LL COVER THE FOLLOWING

In Chapter 3, you have already met with images and the usage of `<img>` tag. You may ask yourself, why does HTML5 have another tag for figures?

The reason behind this fact is that figures have another semantic than images. A figure is a ***self-contained content*** that is related to the main flow of a document, but it can be placed anywhere in the page, and can even be removed without affecting the main flow of the document.

For example, in the image below, you can see such a page. The figure in this article can be moved anywhere on the page without breaking the main flow of the document.



Visual Studio Platform and Extensibility

by Istvan Novak

Thousand ways of extension

Well I know, it is an exaggeration to tell about "a thousand ways" when treating Visual Studio extensibility, but I'd like to point out that you have many choices. In this part I show you how many options you have when dealing with adding some extra stuff to Visual Studio. Reference materials, books, and articles generally enumerate about a dozen options. Instead of simply telling you what they are I would like to methodize them. The key to understand extensibility options is the architecture of the Visual Studio IDE (Figure 1)

When running the Visual Studio IDE we start the devenv.exe file. However, the IDE we see and work with is not just a simple monolithic .exe file or an executable divided into a few .dll files. It is a shell that provides a graphical environment to host functional units, called packages. What we perceive is a cooperation of the shell and hosted packages. The core functions of the IDE are also implemented in packages including the C# or Visual Basic project types, testing features, and many more. The majority of third-party extensions loaded into Visual Studio are also implemented in packages. Just to give you a feeling about how many of them are used: in my notebook I counted 129 packages including those installed with Visual Studio 2008 and third-parties.

Figure 1: Visual Studio IDE

Full article published in CODE Magazine in April, 2008.

A web page with a figure

HTML5 provides the `<figure>` and `<figcaption>` tags to define such a figure. Listing-03-07 given below shows the skeleton of the page in the image above, highlighting the tags that are related to the figure.

Listing-03-07: Making use of the `<figure>` 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;
    }

    figure {
      float: right;
      margin: 0 0 16px 16px;
    }

    figcaption {
      font-size: 0.8em;
      font-style: italic;
      font-weight: bold;
      text-align: center;
    }

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

      footer > p {
        color: white;
        font-size: 0.8em;
      }
  </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>Thousand ways of extension</h2>
        <p>
          Well I know, it is an exaggeration to tell about
          &quot;a thousand ways&quot; when treating Visual Studio
          extensibility, but I'd like to point out that you
          have many choices. In this part I show you how many
          options you have when dealing with adding some extra
          stuff to Visual Studio. Reference materials, books,
          and articles generally enumerate about a dozen options.
          Instead of simply telling you what they are I would
          like to methodize them. The key to understand
          extensibility options is the architecture of
          the Visual Studio IDE
          (<a href="#figure1">Figure 1</a>)
        </p>
        <figure id="figure1">
          <a href="Figure1.gif" target="_blank">
            <img src="/Figure1.gif" width="307" height="118" />
          </a>
          <figcaption>
            Figure 1: Visual Studio IDE
          </figcaption>
        </figure>
        <p>
          When running the Visual Studio IDE we start the devenv.exe
          file. However, the IDE we see and work with is not
          just a simple monolithic .exe file or an executable
          divided into a few .dll files. It is a shell that
          provides a graphical environment to host functional
          units, called packages. What we perceive is a cooperation
          of the shell and hosted packages. The core functions
          of the IDE are also implemented in packages including
          the C# or Visual Basic project types, testing features,
          and many more. The majority of third-party extensions
          loaded into Visual Studio are also implemented in packages.
          Just to give you a feeling about how many of them are used:
          in my notebook I counted 129 packages including those
          installed with Visual Studio 2008 and third-parties.
        </p>
      </section>
    </div>
  </article>
  <footer>
    <p>
      Full article published in CODE Magazine
      in April, 2008.
    </p>
  </footer>
</body>
</html>
```

As you can see, the definition of this figure is pretty simple. The `<figure>` tag defines the whole construct, marking the nested markup as a self-contained content that is related to the document, but it's still independently positioned. The corresponding caption is defined with `<figcaption>`, and without a doubt, it is a part of the figure.

The image of the figure is defined by the `<a>` and `<img>` tags, but it could be anything else that represents the illustration.

As the image above shows, the illustration is positioned aside from the main document flow. This can be quite easily achieved with CSS styling, as you will learn later.

The `<figure>` is defined right before the second paragraph, so that the text wraps it, due to this simple CSS rule:

style.css

```
figure {
  float: right;
  margin: 0 0 16px 16px;
}
```

In the *next lesson*, we will study sidebars in HTML.