

You can still use div

[Mike Robinson](#) Tuesday, July 13th, 2010

“Sorry, can you say that again?”, I hear you ask. Certainly: you can still use `<div>`! Despite HTML5 bringing us new elements like `<article>`, `<section>`, and `<aside>`, the `<div>` element still has its place. Let the HTML5 Doctor tell you why.

Status: Unchanged

In HTML 4, the `<div>` element was defined to be a generic element for structuring a page. Although you can allude to the nature of its content by assigning `id` and `class` attributes with meaningful names, a `<div>` has almost no semantic meaning. [The HTML5 definition](#) is basically the same as in HTML 4:

The `div` element has no special meaning at all. It represents its children. It can be used with the `class`, `lang`, and `title` attributes to mark up semantics common to a group of consecutive elements.

[W3C Specification](#)

`<div>` is literally a container for flow content*, a collection of (hopefully) more semantically marked-up content that may need to be grouped together. It lies at the opposite end of the semantic spectrum from the new HTML5 structural elements.

* For those who haven't encountered this term before, [flow content](#) elements are the same as HTML 4's block-level content elements.

`<div>` vs semantic elements

The new semantic elements (`<article>`, `<section>`, and friends) justifiably capture a lot of `<div>`'s territory, but `<div>` still has a place in the HTML5 world. You should use `<div>` when there is no other more semantically

appropriate element that suits your purpose. Its most common use will likely be for stylistic purposes — i.e., wrapping some semantically marked-up content in a CSS-styled container.

Ask yourself questions about your content: Is it part of the site’s navigation? Is the content secondary to its surrounding content? If this content were printed on a page with nothing else, would it make sense? You will need to evaluate your content carefully, thinking about what it is and pairing it with an appropriate element. I recommend taking a look at [The Amazing HTML5 Doctor Easily Confused HTML5 Element Flowchart of Enlightenment](#) to help you choose the right element for your flow content.

Example Uses

Below are a few examples of how you can still use `<div>` appropriately in your HTML5 sites.

Site Wrapper

While you can [use the `<body>` element](#) as a “natural” wrapper for site content, many people like to use `<div>` as top-level container for styling the entire site. This is an appropriate use of `<div>`, as a site wrapper has no meaning or purpose other than to aid styling.

```
<body>
  <div id="wrapper">
    <header>
      <h1>My Happy Blog</h1>
      <nav>
        <!-- ... -->
      </nav>
    </header>

    <!-- ... rest of site content ... -->
  </div>
</body>
```

Intro paragraph

If you want to style a particular block of content, like the intro to an article, `<div>` is a perfectly suitable element:

```
<article>
  <h1>Big announcement</h1>
  <div class="intro">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
    minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex
    ea commodo consequat.</p>
    <p>Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur.</p>
  </div>
  <p>Sed massa metus, molestie nec facilisis non, lobortis ac orci.
  Suspendisse porttitor laoreet mi, at laoreet dolor rhoncus non. Sed ut
  massa quis mi placerat iaculis non sit amet odio. Cras tempus urna vitae
  felis rutrum porta.</p>
  <p>Maecenas auctor lacus eget mauris tincidunt consectetur. Donec
  molestie malesuada ligula, sed feugiat massa consequat sit amet.
  Pellentesque orci metus, ultricies sit amet adipiscing eget, gravida
  fringilla dui. Vestibulum accumsan dui diam, eget venenatis urna. Sed eu
  lobortis justo. Aliquam erat volutpat. Aliquam erat volutpat.</p>
</article>
```

The first two paragraphs aren't really that different from the rest of the article, but by wrapping them in a `<div>`, they can be styled to capture the reader's attention and draw them into the article. Of course, if you only want to style the first paragraph, you could use a CSS selector like `h1 + p {}` or `article p:first-child {}`. But if you want to style an article's first paragraph only *sometimes*, or you need the style to work in older versions of Internet Explorer, or (as in this case) you want style more than one paragraph, a wrapper `<div>` is the way to go.

Other uses for `<div>`

As previously stated, use of `<div>` will decline in favour of the new semantic elements. If you're using lots of `<div>`s and only a few of the

semantic elements in your HTML5 sites, then you're really not taking advantage of what HTML5 has to offer.

That said, you may need to rely on `<div>` as you transition into the world of HTML5. For example, if many of your site's visitors use Internet Explorer with JavaScript disabled, then you won't be able to style the new elements for these users. A piece of advice we gave back in our article on [how to use HTML5 in your client work right now](#) was to use `<div>` with classes named after the new semantic elements, getting you to think about how you would use the new semantic elements without actually using them.

```
<div class="article">
  <div class="header">
    <h1>My blog post</h1>
    <p>An example using divs with classes instead of the new elements.</p>
  </div>
  <!-- ... content ... -->
</div>
```

```
<article>
  <header>
    <h1>My blog post</h1>
    <p>The same article but switched to the new elements.</p>
  </header>
  <!-- ... content ... -->
</article>
```

This makes it simple to switch to the new elements when it becomes practical (think “search and replace”).

Recap

You will certainly use `<div>` less often in HTML5 than you did in HTML 4, but it's still a valuable element to have in your toolkit. Sure, it'll be picked last for the team because everyone else is better, but it'll be the best damn generic container element there is!*

Look at the more semantically valuable elements and consider whether your content would benefit from what they offer. Need help deciding? Take

a look at some of these articles right here on HTML5 Doctor:

- [header](#)
- [footer](#)
- [aside](#)
- [article](#)
- [section](#)
- [figure](#)
- [nav](#)

For more, check our [article archive](#). And if all else fails, use a `<div>`!

* As it should be, it's the only one apart from ``. That said, if your content is just text, perhaps the oft-overlooked `<p>` would be more appropriate.