Fork me on GitHub

StackLayout

A flexible width, component based CSS layout system

Mockup a basic web page in 10 Steps using StackLayout

Sneak peek of the result

1. Start by dividing the page into 3 typical page rows and wrap everything in a `.stack`:

```
<div id="page">
    <div class="stack">
        <div id="header">
        </div>
        <div id="content">
        </div>
        <div id="footer">
        </div>
    </div>
</div>
```

This Gist brought to you by GitHub. stacklayout-mockup-step1.html view raw

A `.stack` is the outer most parent component that wraps all the other components. You can actually use any stack component as a parent because they are infintely nestable, but `.stack` is the only component that is full width *and* allows other stack components as children.

2. You can then split each of these rows up into components, easily defining flexible, percentage based widths. You can choose from divisions of 2,3,4, 5, full width and auto width components. A full list of available class names is here.

Get Involved

Quick Start

Key Benefits
and points of difference from grid layout systems

Get Technical
Features, enhancements and browser support

The Philosophy
or "Why I created StackLayout by Campbell McGuiness"

Let's start with `#content` and create 2 column components:

```
        <div id="content">
            <div id="column1" class="stack3of4"> <!-- lar
            </div>
            <div id="column2" class="stack1of4"> <!-- sma
            </div>
        </div>
```

This Gist brought to you by GitHub. stacklayout-mockup-step2.html view raw

3. Now let's add some content to `#column1`:

```
            <div id="column1" class="stack3of4">
                <div class="stackContent">
                    <h2>My Awesome Heading</h2>
                    <p>This is sample content</p>
                </div>
            </div>
```

This Gist brought to you by GitHub. stacklayout-mockup-step3.html view raw

You'll notice I've used a `.stackContent` component to wrap the content. This class is **always required to wrap content** and can be applied to any element. See the `<h1>` in step 7 for an example.

`.stackContent` components have a primary and secondary function:

1. **CSS Reset and hook element**. All content must be wrapped by a `.stackContent` so the `letter-spacing:-0.65em` which allows `inline-block` components to behave nicely, is normalised. You can either apply the class to an existing element, or add that element to the relevant lines of CSS. See Using Semantic Class Names for more information. You can also target content using `.stackContent` as a hook element for defining global font properties and style.
2. **Full width components**. They are given the `display:block` declaration, so also function as a full width components. `.stackContent` components can be added as siblings or children of any other stack component. See the next step for a demonstration.

4. Now you may want 3 columns underneath this `.stackContent` component. You can just add 3 `.stack1of3` components as siblings of the `.stackContent` component:

```
            <div id="column1" class="stack3of4">
```

```
            <div class="stackContent">
                <h2>My Awesome Heading</h2>
                <p>This is sample content</p>
            </div>
            <div class="stack1of3">
            </div>
            <div class="stack1of3">
            </div>
            <div class="stack1of3">
            </div>
        </div>
```

This Gist brought to you by GitHub. stacklayout-mockup-step4.html view raw

Notice that no container or clear fix element is required to make the `.stack1of3` components start a new row. The `.stackContent` component occupies the full width of its parent, so the `.stack1of3` components simply 'stack' up underneath it on a new row like text. This is one of the big advantages **StackLayout** has over grid layout systems. If you decide you want 4 columns, simply change the class of each component to `.stack1of4` and add another. If you're wondering about semantic class names click here, or check out the Semantic Class Names mockup page.

5. Then we add some more content:

```
        <div id="column1" class="stack3of4">
            <div class="stackContent">
                <h2>My Awesome Heading</h2>
                <p>This is sample content</p>
            </div>
            <div class="stack1of3">
                <div class="stackContent">
                    <h3>Another Heading</h3>
                    <p>Even more sample content</p>
                </div>
            </div>
            <div class="stack1of3">
                <div class="stackContent">
                    <h3>Another Heading</h3>
                    <p>Even more sample content</p>
                </div>
            </div>
            <div class="stack1of3">
                <div class="stackContent">
                    <h3>Another Heading</h3>
                    <p>Even more sample content</p>
                </div>
            </div>
        </div>
```

This Gist brought to you stacklayout-mockup-step5.html view raw

6. Now let's add some content to `#column2` by populating it with 2 full width `.stackContent` components:

```
            <div id="column2" class="stack1of4">
                <div class="stackContent">
                   <h3>Column Heading</h3>
                   <p>Side column sample content</p>
                </div>
                <div class="stackContent">
                    <h3>Another Heading</h3>
                    <p>Side column sample content</p>
                </div>
            </div>
```


7. Ok, so that was pretty easy, now let's look at the `#header`. I'm going to use full width `.stackContent` components for this. Note the use of `<h1>` and `<ul>` as `.stackContent` components:

```
        <div id="header">
            <h1 id="title" class="stackContent">Basic Moc
            <div id="banner" class="stackContent">
            </div>
            <ul id="nav" class="stackContent">
            </ul>
        </div>
```


8. Let's add some sample content:

```
        <div id="header" class="stack">
            <h1 id="title" class="stackContent">Basic Moc
            <div id="banner" class="stackContent">
                <img width="1100" src="my-banner.png" alt
            </div>
            <ul id="nav" class="stackContent">
                <li><a href="mockup1.html">Basic Mockup</
                <li><a href="mockup2.html">Wrappers and N
                <li><a href="mockup3.html">Template Varia
                <li><a href="mockup4.html">Semantic Class
            </ul>
        </div>
```

by GitHub.

9. Now i'm going to use some `.stackAuto` components in the `#footer`:

```
        <div id="footer" class="stack">
            <div class="stackAuto">
                <p class="stackContent">&copy; 2011 Your
            </div>
            <div class="stackAuto">
                <a class="stackContent" href="#">License<
            </div>
            <div class="stackAuto">
                <a id="twitter" class="stackContent" href
            </div>
        </div>
```

This Gist brought to you by GitHub. stacklayout-mockup-step9.html view raw

`.stackAuto` components are great if you've got a bunch of components with variable width. I'm using them here because i'm not sure about the width of the `#footer` items, but still want these components behave nicely. It this case the `#footer` items will behave like centered text because the parent `.stack` component has `text-align:center` applied by default. You can easily apply `text-align:left/right` to the `#footer` which would only affect the orientation of the footer components. Checkout this mockup for more examples of how you might use `.stackAuto` components.

10. Now let's see the finished markup:

```
<div id="page">
    <div class="stack">
        <div id="header">
            <h1 id="title" class="stackContent">Basic Moc
            <div id="banner" class="stackContent">
                <img width="1100" src="my-banner.png" alt
            </div>
            <ul id="nav" class="stackContent">
                <li><a href="mockup1.html">Basic Mockup</
                <li><a href="mockup2.html">Wrappers and N
                <li><a href="mockup3.html">Template Varia
                <li><a href="mockup4.html">Semantic Class
            </ul>
        </div>
        <div id="content">
            <div id="column1" class="stack3of4">
                <div class="stackContent">
                    <h2>My Awesome Heading</h2>
                    <p>This is sample content</p>
```

```
                </div>
                <div class="stack1of3">
                    <div class="stackContent">
                        <h3>Another Heading</h3>
                        <p>Even more sample content</p>
                    </div>
                </div>
                <div class="stack1of3">
                    <div class="stackContent">
                        <h3>Another Heading</h3>
                        <p>Even more sample content</p>
                    </div>
                </div>
                <div class="stack1of3">
                    <div class="stackContent">
                        <h3>Another Heading</h3>
                        <p>Even more sample content</p>
                    </div>
                </div>
            </div>
            <div id="column2" class="stack1of4">
                <div class="stackContent">
                    <h3>Column Heading</h3>
                    <p>Side column sample content</p>
                </div>
                <div class="stackContent">
                    <h3>Column Heading</h3>
                    <p>Side column sample content</p>
                </div>
            </div>
        </div>
        <div id="footer">
            <div class="stackAuto">
                <p class="stackContent">&copy; 2011 Your
            </div>
            <div class="stackAuto">
                <a class="stackContent" href="#">License<
            </div>
            <div class="stackAuto">
                <a id="twitter" class="stackContent" href
            </div>
        </div>
    </div>
</div>
```

This Gist brought to you stacklayout-basic-mockup1.html view raw by GitHub.

There you have it, a basic mockup website in 10 steps using **StackLayout**

Check out the other mockup

Check out the other mockup pages for more demonstrations:

- Template Variations
- Wrappers and Nesting
- Semantic Class Names

STACKLAYOUT IS THE BRAIN CHILD OF CAMPBELL MCGUINESS

Campbell McGuiness is a member of the Working Software team and has a passion for hand coding highly semantic, cross browser & device compatible HTML & CSS.

StackLayout is sponsored by Decal CMS

Contact Me

Email me @camslizzle

Connect With StackLayout

Facebook @stacklayout

STACKLAYOUT RESOURCES

The Good CSStuff

stacklayout.css

stacklayout_lte_ie7.css

Cross Device Support

stacklayout_mobile.css

Pages

Mockup pages Test page

The Lot

All StackLayout Resources (ZIP) All StackLayout Resources (TAR)

GET INVOLVED

This is a living project. I'm hoping **StackLayout** will grow through people using it, so please let me know how you're using it, what went wrong and what went right.

Github

See the latest developments on our Wiki

Fork This Project

Not sure how? Read this.

Who Is Using StackLayout?

Click here to find out

A

production