Basic Layouts

CSS layouts have changed the way that we are able to design websites. At one time all layouts were done with the <table>. This was both slow and hard to manage.

Table free layouts have been used for almost a decade now while they are an improvement over the <table> they have flaws.

Structure

In this example we are looking the the most basic layout. Assume that this page is a standard content page that has a header, navigation, content area and footer.

```
<header></header>
<nav></nav>
<article></article>
<footer></footer>
```

The issue now is that we have no way to limit the width of the elements. One option would be with set the width on each element there are some issue that come along with that. The box model kicks in when a width is set.

Add Wrapper

The other option is to wrap all the element with an other element like a <div>. This will allow you to define a width on it.

```
<div>
    <header></header>
    <nav></nav>
    <article></article>
    <footer></footer>
</div>
```

Adding IDs

While the structure is correct at this point we want to add an id or class to help identify this <header> from any other <header> that may be on the page.

There are some who say you should never use an id because they are too specific. I disagree with this. I feel that a well place id can help identify which element you are targeting.

```
<div id="container">
    <header id="globalHeader"></header>
    <nav id="globalNav"></nav>
    <article id="contentArea"></article>
    <footer id="globalFooter"></footer>
</div>
```

Styles

Now that we have the structure in place we need to write the css to support the markup.

Fixed width

In this example we are setting up a page that is a fixed width of 800px.

```
<style>
    #container {
        width: 800px;
    }

    #gloablNav, #contentArea {
        float: left;
    }

    #gloablNav {
        width: 200px;
    }

    #contentArea {
        width: 600px;
    }

    #globalFooter {
        clear: both;
    }
</style>
```

This will create a fixed two column layout. The width can be adjusted any value just note that the #globalNav and #contentArea must add up to the width of the #container.