

# How To Build A Custom Website

Building a custom website within Fuse is a fun, streamlined process because of the 1200px Responsive Grid Framework we have implemented into our system. If you don't know how to develop a responsive site in em's and percentages, you may find this a little difficult at first. If you don't know how, or want to read up a little bit on responsive development, some suggested reading before you start is <http://goo.gl/QxaOv>.

The Framework uses a 12 column grid, which pretty much any website design can be developed into a CSS grid system. Get familiar with how grids work!

We build websites starting with a 'template' and build upon it to customize the website to the clients needs. To do this, we change markup in the header, content, and footer and rely heavily on CSS. So, your CSS knowledge must be at a high level.

## Let's get started with some easy to follow steps:

1. The first thing you are going to want is a design mock-up.
  - A. Make sure the mock-up has fonts that are usable from Google Webfonts, Font Squirrel, or Adobe Edge Fonts. If they aren't, choose some from these resources that are similar.
  - B. Crop and save your images (using the least amount of images you possibly can. Use CSS for just about everything.
2. Now, we want to choose the base framework for the custom website. To do this:
  - A. Log into Fuse and click on Admin in the top right corner.
  - B. Click on Templates at the bottom of the left sidebar.
  - C. Search for 'Framework' and choose the Framework that suits your project. At this time, there are only two frameworks available. Content/Sidebar or Sidebar/Content/Sidebar.
  - D. Uncheck the 'Hide' option in the middle of the page and click Update.
  - E. Now, go to the Org you are going to be working with. Click on Website to go to the main website editor and click on Themes in the top admin bar. Select the square thumbnail for the Framework you just unhid.
  - F. Follow the same steps to 'hide' the template once again. We hide this Framework so our clients don't choose a stripped down starting point theme for their live website.
3. Now, you can work your magic in the Org using HTML and CSS. Make sure you are using the same ID classes from the Framework template, especially the .col1 - 12 and .last classes. Take a look at the CSS in the framework template and you'll know what to look for and use.

# Tips & Tricks

## Tag List

We have a Tag List.pdf floating around. Ask and you shall receive. These tags are unique to our system and dynamically create certain things.

## Where Are You?

When customers visit a website, they usually want to know where the business is at and how to get a hold of them right away. This is why we recommend putting the address and phone number on each page, as well as an email signup box and blog headlines on the homepage.

When you fill out the business name, phone number, hours, etc. in Basics, we have shortcodes to pull that information into the webpage. On the home page, make sure to put the email signup and blog headlines in an appropriate spot

```
{email_signup}
```

```
{blog_headline(3,true)}
```

On every page, but the address and phone number:

```
{address} or {address_line} - The later puts the address on one line.
```

```
{phone}
```

You can also include, if you'd like:

```
{email}
```

Refer to the Tag List and the Wiki ([wiki.spaboomblog.com](http://wiki.spaboomblog.com)) for more specifics on the different tags.

## Main CSS Only

DO NOT put CSS into the page itself. Always use the Main CSS box by going to Website > Settings.

## Body Classes

If pages need to be customized differently than other pages on the website, utilize the Body Classes option when you click on the HTML of a page.

## jQuery

We use jQuery a little differently than normal. Take your script, wrap it in `<script type="text/javascript">` `</script>` tags, make a new Snippet in Setup > Pages and give it a unique ID. Depending on where you need the script to execute, put in the necessary place in the website by `{page('uniqueID' ,1)}`.

## Tips & Tricks

### Parts

One of the most useful features to create areas of editable content within our system to make it easy for clients is using something called "Parts". These parts are pieces of information that can be styled anyway you like and are unique on a per Org basis.

To get an idea of how useful Parts can be, check out a new (or normal) org in the Website Editor mode. Notice there is basically one Edit Content icon to update all of the main content area. This isn't very customizable as creating DIVs and SPANs in the editor is never a good idea.

Now to go SpaBoom and look at this website in the Website Editor. Yes, that is awesome.

A Part with an ID of 'part\_box\_callout' will contain the same information on the Home page as it does on the Services page. You need to give each Part a unique ID to be different. However, these can all be styled the same by giving it a class name. An example of a utilizing a Part is:

```
<div boomtime="left" class="col12" id="part_blogroll">{part('blogroll')}</div>
```

If you have questions or need help using our system on a website building level, chat or email [kevin.donnigan@spaboom.com](mailto:kevin.donnigan@spaboom.com).