## General Work Issues

*Please always try to keep in mind the project’s budget targets. These targets should not govern every action, but being aware of them is important. On the same level, please try to be aware of project deadlines and think ahead. If you feel that a certain piece of work can’t be completed in the time provided, relay this to the project manager or the development team leader. This information is readily available in the timesheets system and in the client folder, respectively.*

Having a positive attitude towards figuring out a new mechanism of coding something is a good thing, but if you feel after a while that you are not yet close to figuring out, it might be time to consider some external input.

When being assigned work, honesty is the best policy! If you feel that you don’t know how to do something or you feel that another team member could complete it far quicker, speak up!

## Cross-Browser Compatibility and Accessibility

A new web development should (unless stated otherwise) be supported aesthetically in all modern browsers, plus Internet Explorer versions 7+.

Functionality of a new web development should by usable in all modern browsers, plus Internet Explorer versions 7+. Exceptions can be made for Internet Explorer versions 7 and 8 on a case-by-case basis.

Whilst developments do not need to be 100% accessible (unless otherwise stated), they should be accessible within reason. Ensure that images have alt/title tags and wherever possible for fancy fonts, use text over images. Test how well you have built the page by turning off CSS – does the document still flow and show the same level of prominence for elements as with styling?

Unless specifically stated otherwise, a web development will not be Responsive/Adaptive.

## Content

When coding a masthead image into a template, the masthead select box in the Paprika CMS should be used to allow image alteration.

All Lorem Ipsum style text entered into any development should be regular Lorem Ipsum. All placeholder images should be either a solid colour or relevant to the development in some way. Under no circumstances should alternative “Lorem Ipsum” (i.e. comedy text) or inappropriate images/videos be used when setting up web pages.

Wherever possible, all content on all pages should be directly editable by the client. This should be possible using normal page content, content blocks or an individual module. Do not rely on tags within a content area that could be deleted by inexperienced CMS users. For example, div tags within a WYSIWYG editor will no doubt end up being accidentally deleted at some point throughout the website’s lifespan.

Great care should be taken to make sure that any content entered into a WYSIWYG is directly replicable by a client with no knowledge of HTML/CSS/Javascript.

Unless stated otherwise, keep SEO considerations near the front of development priorities. Avoid duplicate content and aim for appropriate use of header tags / paragraph tags and HTML order. Refer to 1.1c and turn off CSS to check how the page looks- this is how Google reads the page.

## Site construction

Unless stated otherwise, a new web development will use the Paprika CMS system. A fresh copy of the CMS should be downloaded from our core installation at www.paprikacms.com/cpanel and uploaded to the development environment. Ask a technical support agent to assist you with this if required.

It is recommended to have a brief chat with the template designer before proceding with a new template construction to make sure that the intended functionality of the page is fully understood.

Whilst Homepage designs will often be templated first, it is important that all styles with the exception of styles unique to the homepage be coded into the default template stylesheet. This stylesheet should then be @include(d) into the homepage stylesheet.

When working on an already existing development, care should be taken to not damage already completed work on the current page, and other pages on the development. This applies mostly to work on StyleSheets and JavaScript. Be sure to know which elements around the site use a style before changing it.

Functionality in the middle of a CMS page should be placed inside a custom function and included on the page using the available dropdown in the Paprika CMS.

JavaScript files should be included in the HEAD of the document and not at the bottom of the page.

## Plugins

Ensure that any plugins used on a development have an MIT or equivalent license.

When sourcing a plugin,

1. Ensure that it is fit for purpose.
2. Ensure that it can be modified if required.
3. For heavy modifications, ensure that the same effect could not be achieved in less time by developing from scratch.
4. Always keep in mind the weight and memory impact of the plugin. Most machines are not powerful enough to run 500k of JavaScript plugins.

Bootstrap and other similar CSS frameworks are specifically forbidden without consent from either the Development team leader or Digital Production Director.

Jquery: Please download the latest version of jQuery or use a version from a CDN that will not change. This avoids embarrassing issues where websites completed years ago suddenly break because the jQuery version on the CDN was updated too far.

When encountering odd JavaScript issues when using a new version of jQuery and a plugin, please first try using jQuery Migrate as chances are the plugin simply hasn’t been updated to run with the newer versions of jQuery yet.

## General CSS points

Where possible, fonts should be either accepted web fonts or FontFace fonts. Failing this, Cufon is an acceptable fall back as an absolute last resort.

FontFace designations and any @import statements must go at the top of the first stylesheet on a page.

Wherever possible, all content containers should be self-expandable in terms of height so that when the amount of content changes, the container expands / contracts to fit. Always assume something is going to change in terms of content so think ahead and cater for it.

Use of !important: !important was introduced as a quick way of bypassing the main CSS cascade. Please do not use !important unless there is no other obvious way around a problem. Improper use of !important makes debugging more difficult and with stylesheets that are used on more than one page / template (common with our CMS), can interfere with previously completed work. Always check other website pages for impacts made from editing shared stylesheets.

Centering a non-text element: To make a non-text element centrally aligned; a width and margin:auto should be applied to it. Half-width negative margins and 50% left declarations are not a recognised way of achieving central alignment.

With defining active / hover elements and their children, this should be done using display none/block rather than removing the item in question from the viewport using positioning.

Font sizes: Font sizes should always be declared in EMs. One exception to this rule is setting a font size on the body - percentage should be used here. This is recognised as the most cross-browser compatible and accessible way of declaring font sizes.

Responsive CSS: When going beneath the width accepted as “landscape phone”, please switch to using percentages as widths, exceptions may apply but doing this we are able to cover all portrait phones without any issues. Both old and new models have been known to be a fair shot from the usual 320 pixel mark.

## Design issues

If planning to change anything on the design that doesn’t look right, please consult with the designer or team leader or the production director. It is not uncommon that a designer might make a mistake in alignments or sizes.

In many cases the designer will have thought of a way to code the design, if you are unsure of what could do the trick, consult with the designer.