

This document is an overview of how to customise the theme of your Prism 3 tenancy and how to work with the theme system. It covers the structure of Prism 3 pages, along with methods and approaches to applying your own styling in a robust manner.

### Custom Designs and Default Templates

Although Prism allows you a great deal of control over the look and feel of the application, we recognise that not all organisations have the necessary in-house skills to create and maintain these. We therefore offer two options, a “Default Design” and “Design Services”.

The Default Design can be enabled at tenancy setup and provides a basic set of styles, that can be customised with minimal effort. See the Default Design section for more information.

For information on our Design Services for a full or partial design please contact [libraries-sales@capita.co.uk](mailto:libraries-sales@capita.co.uk) or your account manager for a quotation.

# CAPITA

## Contents

<b>Prism 3 Design Guidelines</b> .....	<b>1</b>
Version 4.0.....	1
Custom Designs and Default Templates .....	1
Principles .....	4
Theme Structure .....	5
Site.css .....	5
General Page Structure .....	6
Homepage Page Structure .....	8
Search Results Page Structure .....	8
Item Page Structure.....	11
Account Page Structure .....	12
Lists Page Structure .....	13
Future Pages .....	13
Page Identifiers.....	13
Multilingual Fragments .....	14
Interface Text.....	14
Assets .....	14
Scripts.....	15
Prism Web Addresses .....	16
Secure Pages .....	17
Base Urls .....	18
Working with your Design .....	19
Admin System .....	19
Reseting a theme .....	20
Using sandbox .....	20

Using the Release Preview .....	21
CSS Styling Notes .....	22
Mobile Interface .....	24
Customising the Mobile Interface .....	24
Catalogue Tour .....	27
Customising the tour.....	28
Common Patterns.....	30
A Typical Homepage .....	30
Moving the Homepage Search box .....	31
Linking to New Stock Searches.....	33
Javascript Extensions.....	33
Testing Your Design .....	34
Browser Versions .....	34
Accessibility .....	34
Useful Tools.....	35
Useful References .....	35
The Default Design .....	36
Design Restrictions.....	37
Commenting Changes .....	38
Feedback .....	38

## Principles

Prism's core styles work on the following core principles:

- Be neutral
- Provide good defaults
- Allow localisation
- Client-side only

**Be neutral** – because Prism allows lots of customisation, we try to go with conservative and neutral choices for the core styles so they'll work with your branding.

**Provide good defaults** – Wherever possible we look to provide the best defaults we can for labelling, structure and design. We also aim for these defaults to be usable and accessible, focusing on how people will use the application.

**Allow localisation** – Prism allows for both a translated interface (currently Welsh and English) as well as local text and template changes.

**Client-side only** – because of Prism's hosted nature, we can't allow uploading of server-side components by customers to extend the system. We therefore focus on allowing customisation client-side via HTML, CSS and Javascript within themes.

## Theme Structure

At its basic level, a Prism theme consists of the following files:

site.css

homepage.html

headerlinks.html

header.html

footer.html

javascript.html

login.html

noresults.html

Each of these files serves a particular purpose within the page's structure. They are added into the page when it is built.

### Site.css

The site.css file is the most important, it contains your modifications to Prism look and feel. Prism has a core stylesheet that is always applied and contains the base structure and common design elements (forms, account area and so on). You can add your own styles in the site.css file and it is appended to the base css at run-time. This allows you to overwrite existing styles on a case by case basis to match your institutional site.

## General Page Structure

The main page structure of a Prism page is shown below (with extra elements removed). It's important to note that this diagram is structural and not a layout diagram. Due to the nature of HTML and CSS, how a document is structured can be totally different to its display.



Each of these blocks serves a particular purpose within Prism:

div#headerFragment	Contains all content from the header.html file
#headerlinks	Links at the top of the Prism application panel, relating to core catalogue functions. Its children divide the focus into two types, application links and user links.
ul.appLinks	This element contains the application-level links. Home, Help, Accessibility and so on. The headerlinks.html fragment allows you to add your own localised links here.
ul.userLinks	This element contains user-related links and form elements. Currently this is Account/Login/Lists information, but we may add more in the future.
div#pageHeader	This block contains information that sits at the top of every page application panel.
h1#appTitle	The application title, by default this is “Prism”, but you can change by asking us to put in a custom interface label. In the base stylesheet this element is hidden by default.
div#searchBox	The application search box. This is the most used element in Prism, so keep it prominent.
div#pageContent	The main content of the page. This will vary depending on the page of the application, search content is different to homepage content.
div#pageFooter	The footer information of the application. Includes the footer.html fragment and the Powered by Prism notice.

From a design perspective, the important thing to remember is that the header.html and footer.html allow you to “wrap” the Prism application is localised branding. If you want to include your logo, institutional branding or similar elements, then header.html is the best place to add them.

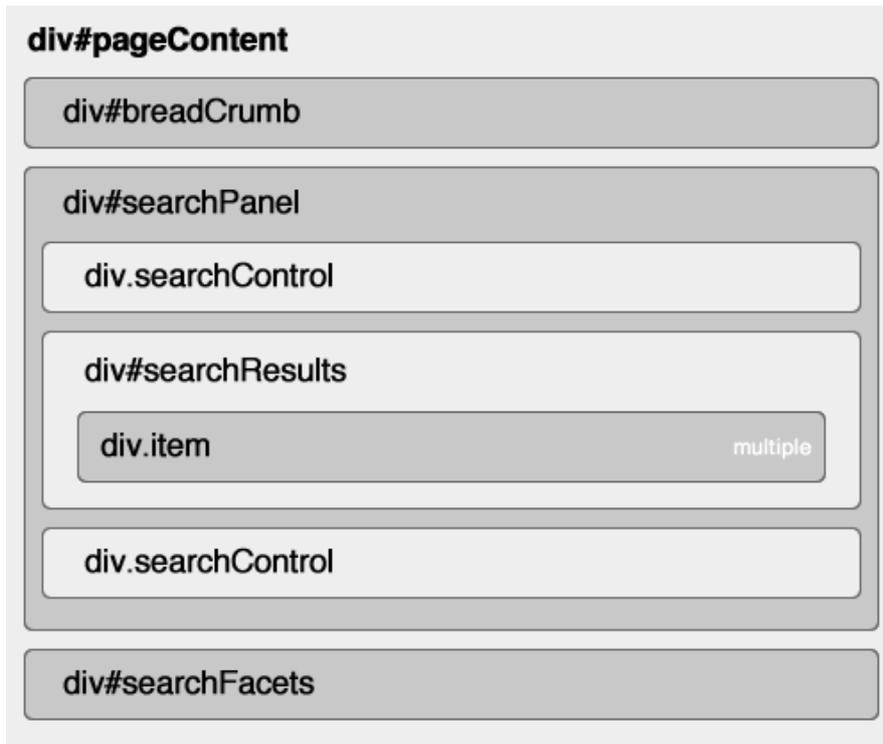
## Homepage Page Structure

The homepage of your Prism tenancy differs from other parts of the application, in that the majority of its structure is available for your own content. This gives you a lot of flexibility to style the entry point as you desire by customising homepage.html and site.css.

For examples of how you can use the homepage.html fragment, see the section on Common Patterns.

## Search Results Page Structure

The structure of #pageContents for search results looks like this:



The main elements are defined below:

div#breadCrumb	A breadcrumb built from search terms and facets.
div.searchControl	Controls for the search, paging, ordering and so on. Appears at both top and bottom of results.
div#searchResults	The results for a search.
div.item	Each individual item in a search.
div#searchFacets	Facets available for filtering this set of search results. Note, by default this is styled to appear before the results, even though in the document order it is after them.

## No Results Fragment

The `noreresults.html` fragment appears on search pages with no results. It is treated a bit differently to others as it allows you to access the search query and insert it into your custom content, for example in a link to another search system such as Copac, the British Library or Google Books.

Taking advantage of this new functionality is easy, just insert the following snippet anywhere in the fragment:

```
{{ query }}
```

If you'd like to provide links to prepared searches on other discovery interfaces you'll need to change that slightly and "url encode" it; this makes sure it's properly formatted for use in a hyperlink:

```
{{ query|url_encode }}
```

So, to add a fragment linking a no results page to searches on Google Books, Copac and the British library, the code would be:

```
<p>You can also try your search for "{ { query } }" on some other catalogues:</p>
```

```
<ul>
```

```
  <li><a href="http://copac.ac.uk/search?&any={ { query|url_encode } }">Search for "{ { query } }" on COPAC</a></li>
```

```
  <li>
```

```
    <a
```

```
href="http://explore.bl.uk/primo_library/libweb/action/search.do?mode=Basic&vid=BLVU1&vl(freeText0)={ { query|url_encode } }&fn=search&tab=local_tab">
```

```
    Search for "{ { query } }" on the British Library OPAC</a></li>
```

```
  <li>
```

```
    <a href="http://www.google.com/search?tbm=bks&q={ { query|url_encode } }">Search for "{ { query } }" on Google Books
```

```
    </a></li>
```

```
</ul>
```

## Item Page Structure

Item pages make up the bulk of Prism display. The overall structure of the page is as follows:

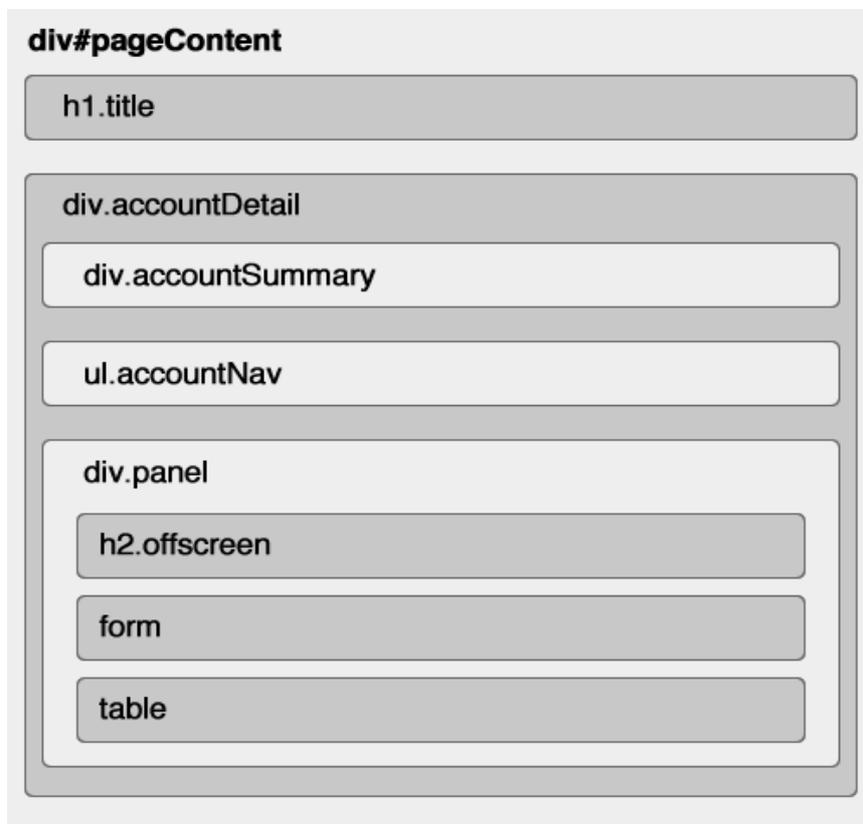


Each of these elements serves a distinct purpose, as detailed in the table below:

div#breadCrumb	Breadcrumb of search arriving at item.
div#itemActions	Containing find more by facets.
div#itemControl	Wrapper information related material.
div.item	An individual item - reused on search results.
div.image	Container for the cover image
div.itemContent	Container for item information.

div.summary	Summary for this item. Contains title, author and key information.
div#availability	Availability of this item at branches.
div#details	More detailed item information, notes and so on.

## Account Page Structure



h1.title	Account page title
div.accountDetail	Wrapper for account information
div.accountSummary	Summary of account information statements
ul.accountNav	Tabbed navigation for the various account section. Each li has a class allowing for tabs to be hidden independently.

div.panel	Panel of account information
h2.offscreen	Sub heading for account section
form	Renew all form (renewals page only)
Table	Tabular data for this section of account information.

In addition to the account pages Prism allows you to add a login page fragment, login.html, giving more specific login instructions before the current login form.

## Lists Page Structure

ul.listmenu	A menu showing all the lists a user has created
div.list	A container for a list of items
div.export	Functionality relating to the export of lists in citation formats and list deletion.

## Future Pages

Prism is an evolving application, and as such we may in future add new pages for new features. Wherever possible, we will try to reuse structural elements of existing pages. We expect to add pages to cover authors, series and specific pages for differing formats in the future.

## Page Identifiers

Each page type of the application has a specific identifier, allowing you to target css and javascript based on the page in question. These identifiers are applied to the body element and are shown below:

#index	Home page
#accessibility	Accessibility information page
#searchhelp	Search help page

#renderitem	Item page
#searchaction	Search results page
#login	Login page
#lists	Lists page
#account	Account page

So if you want to target the search box only on the homepage, you could use:

```
#index div#searchBox{
    color:red;
}
```

## Multilingual Fragments

Each of the fragments in Prism can also be set as a translated version. So you can have `header.html` and `header.cy-gb.html` to switch the header links file if you have multiple languages.

## Interface Text

Text within the Prism interface uses a translations system. This allows us to translate the interface into a variety of languages and also allows you to over-ride default interface text based on your libraries needs.

For example, the default heading for the facets column on a search page is “Refine Your Search“, you might want it changed to “Modify your search”.

To have this change made please raise a service request. It is helpful if you can specify the page url you are requesting the change for, as well as the current text and the precise requested text change.

## Assets

As well as the core fragment files, you can also upload additional assets for use in Prism: images, javascript or other types of file. You can even add additional web pages to your tenancy by uploading html files.

When you upload files into your site's theme they can be accessed at the web address:

```
{tenancy}/assets/{filename}
```

All assets also have a versioning identifier (eTag), which you will sometimes see in the url as:

```
{tenancy}/assets/ec7d42e2-e214-4b01-8dfd-3c46e224412f/{filename}
```

This is simply a versioning identifier and can mostly be ignored and replaced with – as shown above.

## Scripts

The **javascript.html** fragment is designed to hold all your javascript and make sure it is run after the page renders. This speeds up the load time for your catalogue, by delaying script loading until after content has rendered.

You may have read that javascript files should be linked from the `<head>` of the document. This is an outdated practice that should be avoided except in a few specific cases (such as webfont loaders).

If you put your scripts in the `javascript.html` fragment, you can take advantage of Prism's local version of jQuery and therefore avoid duplicate downloads of the same library. To access this add the following script before any others:

```
<script type="text/javascript">  
//use PRISM jQuery object  
$=jQuery=PRISM.jquery;  
</script>
```

## Overriding PRISM scripts

Advanced users may wish to override Prism's default Javascript features. All Prism features are made available as properties of the PRISM object in Javascript. On page load the Prism object runs all methods in the PRISM.actions array, so removing these actions can unset default behaviour. Alternatively rewriting the methods called by these actions can provide different behaviour. Interrogating the PRISM object in the browser console will give you more details. This is only recommended for developers already familiar with Javascript.

## Prism Web Addresses

You may wish to link to particular pages from parts of your theme. You can link to any of these pages using the following form:

Homepage: {tenancyname}/

My Account: {tenancyname}/account

Accessibility Page: {tenancyname}/accessibility

Feedback Page: {tenancyname}/feedback

A search: {tenancyname}/items

An item: {tenancyname}/items/{bib\_id}

A list: {tenancyname}/lists/{list\_id}

{tenancyname}/searchhelp

{tenancyname}/advancedsearch

Simply replace {tenancyname} with the root web address of your tenancy, for example the link to searchhelp on the Broadminster tenancy is:

[//capitadiscovery.co.uk/broadminster/searchhelp](http://capitadiscovery.co.uk/broadminster/searchhelp)

If you have a custom domain name enabled, the url might be:

[//library.broadminster.ac.uk/searchhelp](http://library.broadminster.ac.uk/searchhelp)

## Secure Pages

Prism can work on both http and https, we switch to https when a user logs in so their data for the account is not sent over the open web. This means that your theme needs to work in both http and https. Any assets you upload to the admin console are automatically available in both states.

You may encounter problems with mixed content warnings if you include scripts, images, iframes or css from remote pages. There are several ways to address this.

Assuming you have a secure server to host files on, you can do one of two things:

### **Detect the protocol via Javascript**

With this option, you load assets via Javascript and use a statement like the following to detect whether http or https is used:

```
var protocol = ("https:" == document.location.protocol) ? 'https://' :  
'http://';
```

And insert any elements into the page using Javascript.

### **Use protocol independent uris.**

With this option, you adjust all asset urls to use the following form:

```

```

Note the lack of http: before the //. This will cause the asset will be loaded with the same protocol as the main page.

If you don't have a secure server to host your files on, then it is best to use assets uploaded to Prism only. In this case simply refer to any item by a relative url and the base url in Prism will handle whether it is requested securely or not.

## Base Urls

All pages in prism have a <base> element set. This means that if a web address you use in a theme is not a full url, the base url of your tenancy is prepended to it.

For example if you add a link: `<a href="feedback">Feedback Page</a>` the link will go to `http://capitadiscovery.co.uk/{yourtenancy}/feedback`, and will work whether on http or https.

## Working with your Design

### Admin System

All Prism tenancies are administrated via the `admin.capita-libraries.co.uk` website. You will be given credentials to access this site and make amendments when your tenancy is set up.

The system allows you to configure a variety of settings for the application, one of these is the theme files.

The screenshot shows the 'Theme' configuration page in the Capita Libraries Admin interface. The page has a top navigation bar with 'Capita Libraries Admin', 'Dashboard', 'Account settings', and 'Logout'. A left sidebar menu is titled 'BROADMINSTER' and includes items like 'Overview', 'Operator', 'Feedback Email Addresses', 'Community Features', 'Theme' (highlighted), 'Layout Modules', 'Collections', 'Locations', 'Facets', 'Reset Tenant Cache', 'User Content Moderation (Lists)', and 'User Content Moderation (Tags)'. The main content area is titled 'Theme' and contains a yellow warning box: 'Please note: Any changes made here will overwrite your theme and cannot be undone. Ensure you download a backup, using the button below, before proceeding.' Below this is a 'Download theme backup' button. A table lists theme files with columns for 'Filename', 'Size', and 'Type', and a 'Delete' button for each row. At the bottom, there are two upload sections: 'Upload single file to this folder (overwrites an existing file of same name):' and 'Upload new theme as zip file (overwrites all files):', each with a 'Choose File' button and an 'Upload' button.

Filename	Size	Type	
assets	-	folder	
images	-	folder	
footer.html	218 bytes	text/html	Delete
header.css	1.7 kb	text/css	Delete
header.html	209 bytes	text/html	Delete
headerlinks.html	73 bytes	text/html	Delete
homepage.html	6.2 kb	text/html	Delete
javascript.html	16 bytes	text/html	Delete
login.html	89 bytes	text/html	Delete
noresults.html	554 bytes	text/html	Delete
site.css	7.9 kb	text/css	Delete
test2.txt	76 bytes	text/plain	Delete

The theme admin page allows you to upload individual files into the root folder, upload a single zip file containing a whole theme to update or download a backup of your theme.

If you do upload a whole theme, be careful to keep the file size down. Anything you upload will have to be downloaded by your users! Uploading a theme can take some time if there are many files or big files. It's easy to forget to remove script documentation or examples, which make the whole process take longer than needed. It is therefore best to only add what you're using to the theme.

Note: You should always download a copy of your theme to use when making changes and add changes to those files rather than working from an older copy. With the nature of the tenancy setup, it is possible that somebody has made changes to your theme since your last edit, so the best way to be sure you are using the up to date version is to re-download.

## Resetting a theme

In order to speed up the display of Prism, assets and styles relating to your theme are aggressively cached in the application. This means when a user accesses your prism tenancy the page loads quickly.

When you load in new elements of a theme, you may find they don't display right away as the old ones are still in the application cache. To clear these out, you need to request <http://capitadiscovery.co.uk/{tenancyname}/resettenantcache> to clear the cached versions. You can also clear the cache using the link in the admin console.

It's worth noting that resetting the cache may make the very next request for your catalogue slower as the cache is refreshed, so it's best to avoid doing this too much during peak hours.

## Using sandbox

All Prism tenancies are set up so that you have two sites to work with. Your main live site that you point the public at, and a sandbox site where you can try out new designs and features before making them live. When you log in to the admin site, both your tenancies will be visible. Your sandbox will be located at:

<http://capitadiscovery.co.uk/{tenancyname}-sandbox/>

When you test design changes, take a copy of your live theme by downloading it from [admin.capita-libraries.co.uk](http://admin.capita-libraries.co.uk). This ensures you have the latest version of the live theme, including any modifications made by other people. Make any modification you require and upload to the sandbox tenancy to test them.

## Using the Release Preview

Prism is an evolving piece of software. When a new version becomes available, we place it on a demo web address so that you can test any new functionality and check your themes work with the new version. This web address is: <http://demo.capitadiscovery.co.uk/{tenancyname}>

You can use the demo url for both sandbox and live tenancies, allowing you to preview changes to themes for the new release. This testing time period is usually two weeks, though may be longer if issues are discovered.

Note: If you are using Prism on your own hostname then demo is not available at your own url, you will have to use <http://demo.capitadiscovery.co.uk/{tenancyname}> to preview changes.

## CSS Styling Notes

### CSS browser targeting

Prism includes a robust method of targeting older versions of Internet Explorer with CSS rules to account for browser quirks. By prefixing a CSS selector with a particular class you can target older versions of the browser with specific fixes. For example:

```
.item{  
    width:98%;  
}  
  
.ie6 .item{  
    width:100%;  
}
```

would declare a normal width on newer browsers of 98%, but force IE6 to set the width to 100%. This method prevents the use of ugly and problematic “CSS hacks” that can cause unexpected behaviour when we upgrade Prism.

Note: We will be dropping active support for Internet Explorer version 6 in the near future as usage of this browser drops to minimal levels. We will move to a passive support phase, where it receives a stripped down, plain, but usable version of the main styles only.

### CSS Merging

Prism's CSS handler merges the files together, reducing network requests. Where previously we linked to three files base.css, site.css and print.css, these files are now wrapped together as one, with the mobile interface included if enabled. This merged file has a url of assets/{etag}/merged.css, which means it looks like it's in your theme so existing image references work.

When this file is served it is sent with far future expiring cache headers. This means that the file will always be cached, reducing requests for large CSS files and speeding your catalogues display. If you change your site.css or config.ini, or we release a new version of Prism, running a `resettenantcache` will change the file's web address and force a re-download.

## CSS selectors

The more specific your CSS selectors are, the less likely they are to break or cause consequences you aren't aware of. This is especially true where we reuse generic marker classes like `last`, `first`, `more`, `author` or `formElement`.

For example, we have a class `.formElement` that is added to all form element groups, not just the those on a particular page. If in doubt about how an item is reused elsewhere, include a prefix to the selector using a page id, for example: `#feedback .formElement`.

## CSS Technique to Avoid

Avoid applying background images or fixed width to generic HTML elements, as this will cause issues when new functionality is added. Particularly avoid applying these to all `a`, `button` or `input` elements as we are very likely to add new features that have these elements.

Particularly avoid patterns such as fixed width elements with fixed width backgrounds to achieve button shadows and for styling text inputs. Use tiling images or CSS3 features like `box-shadow` to achieve these effects.

## CSS Prefixes

If you are adding classes or ids for CSS or JavaScript hooks, it's best to add a local prefix to avoid potential collisions.

For example, you might decide to create a class `.panel` that conflicts with Prism's `.panel` class. By making your class `.mylibrary-panel` you can ensure future releases won't ever cause conflicts.

We are currently moving Prism to a more robust CSS grammar where we namespace Prism visual UI classes with `.ui` and highlights theme-able elements with `.theme` we recommend not adding your own styles to any class prefixed with `.ui` and to consider this reserved.

## Mobile Interface

Mobile interface, or more correctly small screen interface, is turned on in the admin console via the Theme Options link. We recommend you enable it on your sandbox first to test theme compatibility. There are two options: enabling the mobile interface and enabling html5.

When enabled, main effect of the mobile interface will add extra CSS that override existing themes in a mobile context. These changes are adaptive, which means that when you size a modern browser like Chrome or Firefox or Internet Explorer 9 to the targeted window size the mobile interface will come into effect. Extra features are also enabled such as swipe-based paging and background loading of search results.

The Prism mobile interface makes use of a number of features from the new version of HTML (the markup language of the web) and Javascript APIs. These features are only enabled if you tick the HTML 5 box in the admin console, otherwise older XHTML4 markup is used.

### Customising the Mobile Interface

The mobile interface is a series of additional stylesheets and behaviours for small devices. It aims for a more “app-like” feel, which focuses on core tasks and their requisite elements of the interface, often hiding extra detail behind reveals to optimise for the small screen size.

Because mobile is a more fragmented space than desktop, where slight changes can have major impacts you might not see on one particular phone, the mobile interface has a more restricted set of configuration options. This makes the theme more reliable across devices and enables us to fix in one place rather than across multiple tenancies.

Configuration is achieved by using a file called **config.ini** that is uploaded as part of your theme files via the admin console. Rather than write CSS rules it allows you to specify parameters that we turn into CSS for you. This means you don't have to worry about browser inconsistencies, since we handle that for you.

Note: When mobile is turned on we strip all !important declarations from site.css to enable us to override all styles more easily. This may cause oddities in older stylesheets that utilise !important to target browser bugs in older Internet Explorer variants, we recommend you move to using the provided .ie6 styles for this.

A config.ini looks like this (note: lines may wrap):

```
;this is a comment...

[Mobile]

touch_icon = //capitadiscovery.co.uk/{tenancy}/assets/-
/images/start_icon.png

mobile_header_background = //capitadiscovery.co.uk/tenancy/{assets}/-
/assets/images/mobile_header.png

mobile_form_element_border = #E3E5F5
mobile_form_element_background = #fff
mobile_form_button_background = #fafafa
mobile_form_button_background = #f0f0f0

mobile_tag_background = #D2E3FF
mobile_tag_border = #80B2FF
mobile_tag_text_colour = #333

mobile_text_colour = #222
mobile_link_colour = #28489D
```

This initial set of configurable parameters is conservative and allows for form element styling, facet / tag styling and main text colouration. It also allows you to specify the background image for the application title and the touch icon used when creating a shortcut icon on iPhone/iPad.

The web addresses for these images should be full urls, with a protocol independent uri so they work under secure and insecure connections. Note the lack of http: on the urls used, for example:

```
//capitadiscovery.co.uk/{tenancy}/assets/-/images/start_icon.png
```

The sizes for iOS start icons are 57x57 pixels. You can learn more about these at:

<http://developer.apple.com/library/IOS/#documentation/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html>

You may also want to specify a custom background for the application header. This sits at the top of all pages in the mobile interface. It should be 50px high, by a maximum of 480px wide. Note that the 480px is a maximum and that the image will be truncated on smaller screens.

## Extra Links

Mobile hides header and footer fragment detail to save space and reduce to the core functionality. There may be items, such as links to terms and conditions, that you are required to include a link to and wish to re-instate on mobile. Be cautious of what you choose to add, since it is far easier to overload the interface on a mobile device than on a desktop.

The best option is to add this to the headerlinks.html fragment as a menu item, giving it the class of mobile. This will hide it on desktop devices, but place it in the openable menu on the top of the screen should access be needed:

For example:

```
<li class="mobile"><a href="{url}">Mobile Help</a></li>
```

## Javascript Extensions in Mobile

If you make use of any extensions to the Prism interface, image carousels and so on via either Juice or jQuery you may wish to disable these when mobile is enabled as they may cause slowness due to extra network requests and more involved scripting.

You can set these to not run when mobile is enabled by checking the PRISM.mobile object property in javascript. This is set to true when the main window is smaller than 480px. You can then turn of features that aren't suitable to run in a mobile context.

```
if (PRISM.mobile) {  
    //code to run mobile essential extensions  
}
```

```
else{
  //code to run normally
}
```

You could also modify how Juice is included in the page to use this too, reducing your downloads on mobile devices even further. So for example, the following could be added to your javascript.html fragment:

```
<script type="text/javascript">
if(!PRISM.mobile){
  document.write('<script type="text/javascript" src="assets/-
/js/juice.js"></script>');
}
</script>
```

This would only include the juice.js file if mobile is not enabled.

If you are using extensions, we recommend you move to using the version of jQuery bundled with Prism, to avoid downloading the same library multiple times.

## Catalogue Tour

Take the Tour is a javascript enhancement to Prism that helps new users learn about the functionality of the system. We provide a default tour, though the tour is totally customisable based on your needs.

To enable the tour you need to add a couple of lines to your theme's config.ini file :

```
[Tour]
tour_en-gb = /js/tour.en-gb.js
tour_item = 65964
tour_search = London
button_location = body
```

The first element `tour_en-gb` says that there is a tour file at that location for the English language. In this case it is the location of the default tour. The `tour_item` is the bib id of an item to be used as an example in the tour. `tour_search` is the search term to be used in the tour, this needs to be a term resulting in multiple pages. `Button location` is where the tour start button is appended in the document (by default at the very end).

To add a Welsh language tour you would use:

```
tour_cy-gb = /welsh_translation_location.js
```

## Customising the tour

The javascript API for the tour allows you to create a custom script file with your own sections, or add a new section to the default tour. The default tour file can be found at: <http://capitadiscovery.co.uk/js/tour.en-gb.js> . We recommend reading this file as the first first step.

To build up pages in the tour and the steps within them, you can utilise several methods on the `PRISM.TOUR` object.

For example, to make a step for the tour with a CSS selector identifying the element to label, some text describing it and where the tour box is positioned:

```
var step=TOUR.makeStep('#s',  
                        'This is the search box.',  
                        'below');
```

To add a page to the tour that is run on the homepage, has the title basic searching and has this step to the tour:

```
TOUR.addPage('home', 'Basic searching', [step]);
```

Adjust the text of a particular page and step.

```
TOUR.setTourText(1, 2, 'Local text customisation');
```

Set the title of a particular page of help:

```
TOUR.setTitle(1, 'Catalogue introduction');
```

Insert a new page after an existing tour page:

```
TOUR.insertPage (1, 'home', 'Homepage features', array_of_steps);
```

Replace a tour page with a new one:

```
TOUR.replacePage (1, 'home', 'Search steps', array_of_steps);
```

## Changing the Tour Start Button

The tour start button is configured in the config.ini, but styling comes from the default stylesheet. The base styling is:

```
#startTour{  
    margin:1em;  
    background:green;  
    color:white;  
    border: 1px solid #006400;  
    width:22em;  
  
    position: absolute;  
    right:0;  
    top:0;  
}
```

This works to place it visually at the top of the document. Set the position to `position:static` and alter the config.ini variable `button_location` to place the button elsewhere in your theme.

## Common Patterns

We often get asked about how to implement certain features in a theme. Some of the common requests are discussed below.

### A Typical Homepage

It's common for libraries to want two panels of content on the homepage, one for new or interesting books, one for news and attendant content. The following code will give you a basic set of panels to work with.

Add the following your homepage.html file:

```
<div id="hp-primary-content" class="hp-panel hp-wide">
  <h1>Books we Love</h1>
  <div class="hp-items">
    <!--page item blocks from search here -->
  </div>
</div>

<div id="hp-secondary-content" class="hp-panel hp-short">
  <h2>Library News</h2>
  <ul class="hp-news">
    <li><a href="http://">News Item Link</a></li>
    <li><a href="http://">News Item Link 2</a></li>
  </ul>
</div>
```

Then add the following to site.css:

```
#index .hp-panel{
    float: left;
    display: inline;
    padding: 1em 1%;
    border: 1px solid #eee;
}
#index .hp-short{
    width: 32%;
    padding-top: 1.5em;
}
#index .hp-wide{
    width:60%;
    margin-right: 2%;
}

#index #pageContent{
    overflow: hidden; /*clear floated elements*/
}
```

## Moving the Homepage Search box

Often libraries want the focus of the homepage to be the catalogue search, and so want the search box “moved” from its normal position in the header.

The easiest way to achieve this is to add your own search box code in the `homepage.html` fragment and then hide the normal search box on the homepage.

```
<form id="homeSearchForm" method="get" action="items">
    <label for="homeSearch">Search the library catalogue</label>
    <input id="homeSearch" type="text" value="" maxlength="1000" name="query"
/>
```

```
<input type="submit" value="Search" />
<a href="advancedsearch" class="searchBuilderActivate">Build search</a>
</form>
```

You can then style the two searchboxes with different looks and feels, by targeting using the `#index` selector to specify the homepage only. For example you could add the following to your `site.css`:

```
#index #searchForm{
    display:none;
}
#index #homeSearchForm{
    padding:1em; /*extra padding */
}
```

This new search box will not have the new search builder functionality by default. You will need to add it by adding a snippet to Javascript to trigger it. Add this to your `javascript.html` fragment:

```
<script type="text/javascript">
if(PRISM.page=='index' && !PRISM.mobile){
    PRISM.search_builder_shift=40;
}
</script>
```

By default the search builder opens above the pages main hidden search box. The value of `search_builder_shift` should be altered to ensure it appears as needed on the homepage.

If you want a link or button to trigger the search builder, simply add the class `'searchBuilderActivate'` to it.

## Linking to New Stock Searches

This feature requires the new MarcGrab software. Assuming you want to show the latest additions to the collection called “Main Catalogue”, enter `collection:“Main Catalogue”` in the search box and click search.

Copy the url from the address bar and append the following to it:

```
&sort=shelveddate%3Ad&limit=10
```

This modifies the search to return in order of shelved date and show only the last 10.

More detail at <http://blogs.capita-libraries.co.uk/prism/2010/04/15/new-stockprepared-searches/>

## Javascript Extensions

Using javascript you can add extra functionality to your tenancy. We recommend using jQuery for extending Prism. jQuery is a Javascript library that solves the cross-browser headaches of javascript and has a variety of extensions that can be used to enhance the way information is displayed.

In addition to jQuery, you may wish to use library specific extensions provided by Juice. It uses jQuery as a base and allows you to add extensions with a range of functionality, from cross searching to embedding maps. We recommend you use at least version 0.7.1 of Juice, as this is the latest, most stable version.

More detail of how to add Juice to your Prism tenancy can be found online at: <http://juice-project.org/PrismJuice>

## Testing Your Design

Once you have your theme finished and uploaded to your sandbox tenancy, it's best to run through a few checks.

### Browser Versions

If you are not a practiced web designer, it's easy to fall into the assumption that if a design looks right on your particular setup, it works for everybody else.

For example, some institutions use the Internet Explorer 6/7 browsers, but those browsers account for only around 3% of Prism's traffic. If you only test on these older browsers, with ~~their~~ many bugs, you may find your theme breaks on more modern systems.

When you make big changes to a design, it's best to test it with the current and previous version of popular browsers as these make up the majority of your users.

### Accessibility

By default we make Prism as accessible as we possibly can, aiming for WCAG 2 level 2. However, when you make modifications to themes it is possible through adding javascript or changing CSS to hinder accessibility.

Common issues to look for:

- Making colour contrast too low in your styles
- Missing textual alternatives on forms or images
- Design breaking when resizing text due to using pixel widths
- Not providing keyboard accessibility in extensions

## Useful Tools

Browsers all now come with the advanced debugging tools:

**Firefox Developer Tools** - <https://developer.mozilla.org/en-US/docs/Tools>

**IE Developer Tools** - [www.microsoft.com/download/en/details.aspx?id=18359](http://www.microsoft.com/download/en/details.aspx?id=18359)

**Chrome Developer Tools** - <https://developers.google.com/chrome-developer-tool>

**Modern IE** - <http://www.modern.ie/>

A site dedicated to testing sites in modern versions of Internet Explorer. Includes downloadable virtual machines for running outdated versions for testing.

**Firefox Accessibility Toolbar** - <https://addons.mozilla.org/en-US/firefox/addon/juicy-studio-accessibility-too/>

Tools including a colour contrast analyser.

## Useful References

These guidelines can't be a guide to building with CSS, Javascript and HTML. Fortunately people have already written on these topics extensively, we highly recommend the following sources:

### Books

Designing with Web Standards, Jeffrey Zeldman

Bulletproof Web Design, Dan Cederholm

DOM Scripting, Jeremy Keith

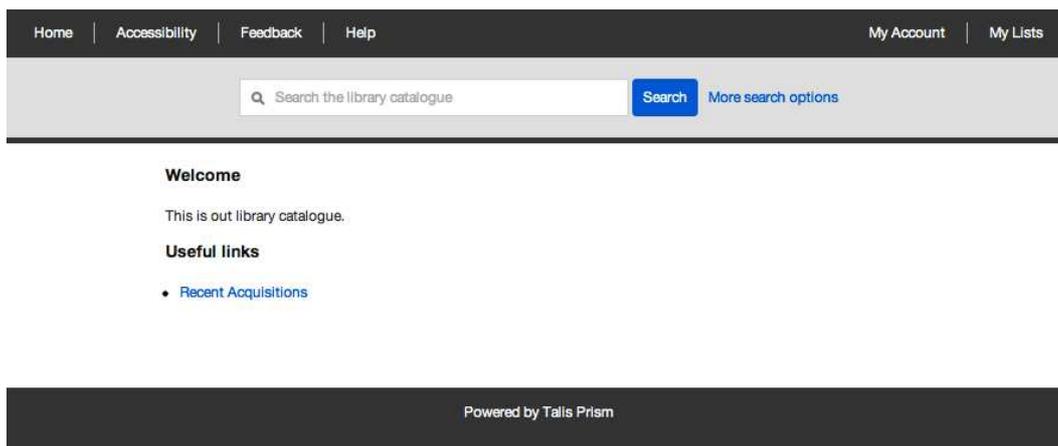
Web Accessibility: Web Standards and Regulatory Compliance, Bruce Lawson et al

### Websites

Our forum pages at <http://forums.capita-libraries.co.uk/> detail many of the features and patterns for developing Prism themes as well as other topics like the Linked Data API.

## The Default Design

The default design is a neutrally coloured standard theme that can act as a base to build from: We are currently working on a new version, which can be viewed at <http://capitadiscovery.co.uk/newdemo/> and can also be applied as a base to work from.



## Design Restrictions

With each application release, new functionality may be added, or existing functionality amended. This means your theme must not be limited to the current behaviour and structure, as this is guaranteed to change with every release.

To preserve the consistency and usability of the application, the following restrictions should be observed. Failure to do so can result in unpredictable behaviour or theme breakages when we update the system or add new functionality. Fixes to themes that have not followed these guidelines are chargeable.

### 1. Don't remove the search box

Removal of the search box, search button from any of the pages is strictly prohibited. The exception to this is replacement search boxes within homepage.html.

### 2. Don't alter key links

The following links must be present on all screens:

- A link to the catalogue home page
- Accessibility Help
- Feedback
- My Account
- My Lists
- Powered by Capita's Prism

**Do not remove links or duplicate core functionality with your own copies, this applies most to the `#headerLinks` panel. This may result in functionality becoming unavailable to your users.**

### 4. Prism application panels styling

In general, if we add new features, they will appear in the main application panel (`#pageContent`) or the userlinks area (`.userLinks`). This means to avoid issues when we upgrade the system, the easiest solution is to only apply minimal changes to the layout of these core areas.

**Focus your styling on the header.html and footer.html.**

## 5. Don't use fixed pixel widths

If you set widths on common elements, for example buttons, to fixed pixel widths, you will run into problems when interface text changes, new elements are added or simply when a user resizes the browser text. Size and position using % or ems to avoid issues. Leave enough room in your theme for extra links in the `.userLinks` area, as we will add links to new features there.

## Commenting Changes

Occasionally our support staff will make changes to your tenancy theme on your request. When we do so, we'll leave a comment in place explaining the change with the date and initials.

Width: 30px /\*Fixed widths - 12/03/09 MM\*/

## Feedback

These design guidelines will change as Prism evolves. If you have any feedback or suggestions for further coverage, please contact us at [libraries-sales@capita.co.uk](mailto:libraries-sales@capita.co.uk)