Cpaterson.com Website Refactoring Project

## Requirement

Since the original version of website [cpaterson](http://www.cpaterson.com) cannot be visited by major browsers, customer wants to rebuild the website. The requirements can be summarized as following:

* Website shall be supported by all major browsers (Chrome, Firefox, IE)
* Among all the versions of the major browsers, 95% should render the page properly. The rest 5% includes old versions that even do not support CSS basic functions like IE6.
* All the contents, page layout, color style shall remain the same
* New website shall convert Adobe flash based to HTML/CSS based
* New website shall be stable enough to work 7\*24
* Good scalability
* Nice to support mobile devices

## Design

After check the old version of the website, there’re several defects which prevent the website from working normally.

* The root cause of the issue is the widely used Flash format videos. Due to flash videos are not supported by default nowadays, browsers will require users to download Adobe plugins in order to render the video. Unfortunately, users always reject the download.
* The navigation bar is using static images instead of CSS format.
* Formats are written in tag and not managed in CSS file.
* Contents including text are using image.

Based on above issues, refactor the website on following aspects:

### Reorganize Site Map

The new site map is a typical single root tree structure. One homepage provides multiple entrance to the content page, and each content page provide navigation to sibling page or homepage.



### Covert Flash Video Elements

Because main stream browsers stop supporting flash format videos by default, in order to avoid the plugin download triggered by flash elements, convert flash videos into mp4, ogg, webm format.



### Adopt CSS to Manage Style

Although the website has multiple content pages, their styles are unified and have many format (font size, color, bullet images, etc…) in common. Instead of writing CSS code everywhere in tags, manage all common styles into one file. Therefore all styles can be managed easily and duplicate code are reduced.

### Design Dynamic Page

Since each page has its subtitle and clickable items, the new website is to reuse one <div> to dynamically display related content. By using Javascript, number of pages can be reduced significantly.

Another reason is because different items may have different size, which sometimes may even need to change whole page layout. For example, packaged good

### Use Templates

Organize header, navigator, and footer content into templates to further reduce duplicity.

### Cross Brower Support

## Test Result

## Future Enhancement Recommendation

XXX

* Main entrance is based on video element

## Backup URL

* [Source code and project document](https://github.com/ZiqingQiu/cpaterson.com)
* Browser test page