CSS guidelines

1. **CSS Basic Specifications**

Version: CSS 3 and Bootstrap 3

In principle, CSS is managed with an external file.

※ When using CSS3, judge by IE whether it should be applied by htc or js or progressive enhancement.

1. **External CSS file list**

|  |  |  |
| --- | --- | --- |
| **File name** | **Contents** | **Storage folder** |
| common.css | CSS reset & default CSS  (Manage default style for OOCSS and responsive) | / css / |
| style.css | Style sheet file for all pages + main layout |
| xxx.css (ex: index.css) | Specific style sheet file of each page |

1. **Naming rules for selector names**

In principle, ID / class names are related by using half-width alphanumeric characters and written in lowercase letters.

***Example:*** *.header or #header*

If the abbreviated word is not general and it is difficult to understand, please list it without abbreviation.

***Example:*** *.customer*

When naming, if the associated word is more than one word, describe the **two** words subsequent by (-).

***Example:*** *.line-img-product or #line-product*

However, differences in naming rules due to use of plug-ins etc. are somewhat acceptable

***Description example***

*#adHead*

**Organizing classes and id one scientifically**

This is to assist you later when reviewing your code easier and do not tangle with his naming, while also easily if applied some CSS preprocessor or BEM.

|  |  |  |
| --- | --- | --- |
| **Instead of naming** | **Change** | **Or** |
| *.titlepost {}*  *.headerpost {}*  *.contentpost {}* | *.post {}*  *.post-title {}*  *.post-header {}*  *.post-content {}* | *.post {}*  *.post > .title {}*  *.post > .header {}*  *.post > .content {}* |

1. **Style applied to area**

In principle, each area give the following **Tag,** **ID, class** name.

|  |  |
| --- | --- |
| **File name** | **Applicable area** |
| header | header |
| nav | Global navigation |
| .breadcrumbs | Breadcrumb navi |
| .row | Content area and sidebar |
| footer | footer |
| section | When dividing each section by main content area and sidebar |
| #section | Description section ID |
| .item | When further division is required in each section |
| .inner | Further, when it is necessary to divide the area one level downward |

1. **Helper classes**

**Helper classes** is the predefined class attribute some simple and frequently used. For example, text-align, font-weight, color ... Class synergy in the process of writing HTML and you just add it to class without having to redefine the property. This approach is useful when you use a custom CSS framework and the need to "very few" properties, or you do not choose the name of the class is suitable for that element, or elements that do not need to specify the class to do something with javascript.

The advantage of doing this besides you do not need to redefine some properties are not necessary, if the CSS as much, you will save quite a lot of attributes to declare again and will reduce the size of the CSS file.

Ex: I use framework bootstrap

*<p class="text-center">*

*<button type="button" class="btn btn-success btn-md text-uppercase">*

*View More*

*</button>*

*</p>*

Instead of having to write your CSS as follows to the discretion custom

*p { text-align: center; padding-top: 20px; }*

*p > button { background-color: #fff; padding-right: 30px; padding-left: 30px }*

Then I will declare for the additional class can reuse for other cases later

*.padding-top-20 { padding-top: 20px; }*

*.padding-right-30 { padding-right: 30px; }*

*.padding-left-30 { padding-left: 30px; }*

*.btn.btn-bg-white { background-color: #fff; }*

Thus in their HTML will declare as follows

*<p class="text-center padding-top-20">*

*<button type="button" class="btn btn-success btn-md btn-bg-white text-uppercase padding-right-30 padding-left-30"> View More </button>*

*</p>*

1. **Css file size reduction**

CSS file size reduction is performed by removing unnecessary white space, line breaks, the last semicolon in class or id ... This stage is also called *minify*. For easy distinction between files and file minify minify not, you should add the name .min minify files.

Ex: *style.css after going into style.min.css minify.*

**Note:** You can minify by <http://www.cleancss.com/css-minify/> or tool Koala-app

1. **Combine external CSS**

Combine external CSS is the easiest, instead of having to write a css files into HTML documents, you just collect those files into a single file. Content files will contain the entire contents of the files, so you'll reduce multiple requests to the server and server offload for many.

Ex:

*<link rel="stylesheet" href="/css/header.css">*

*<link rel="stylesheet" href="/css/navigation.css">*

*<link rel="stylesheet" href="/css/footer.css">*

**Combine**

*<link rel="stylesheet" href="/css/style.css">*

1. **OOCSS(Object-Oriented CSS)**

Create a class for each design and code it with OOCSS (Object-Oriented CSS).

If it consists of reusable modules, it reduces the number of man-hours required for coding, as well as the speed of reading pages.

Ex:

*.btn-primary {*

*display: inline-block;*

*padding: 6px 12px;*

*margin-bottom: 0;*

*font-size: 14px;*

*font-weight: 400;*

*line-height: 1.42857143;*

*text-align: center;*

*white-space: nowrap;*

*vertical-align: middle;*

*border: 1px solid transparent;*

*border-radius: 4px;*

*color: #fff;*

*background-color: #337ab7;*

*border-color: #2e6da4;*

*}*

*.btn-success {*

*display: inline-block;*

*padding: 6px 12px;*

*margin-bottom: 0;*

*font-size: 14px;*

*font-weight: 400;*

*line-height: 1.42857143;*

*text-align: center;*

*white-space: nowrap;*

*vertical-align: middle;*

*border: 1px solid transparent;*

*border-radius: 4px;*

*color: #fff;*

*background-color: #5cb85c;*

*border-color: #5cb85c;*

*}*

***I will gather some of the properties of this class into a class 2 others named btn:***

*.btn {*

*display: inline-block;*

*padding: 6px 12px;*

*margin-bottom: 0;*

*font-size: 14px;*

*font-weight: 400;*

*line-height: 1.42857143;*

*text-align: center;*

*white-space: nowrap;*

*vertical-align: middle;*

*border: 1px solid transparent;*

*border-radius: 4px;*

*}*

*.btn.btn-primary {*

*color: #fff;*

*background-color: #337ab7;*

*border-color: #2e6da4;*

*}*

*.btn.btn-success {*

*color: #fff;*

*background-color: #5cb85c;*

*border-color: #5cb85c;*

*}*

1. **ID allocation in the <body> tag**

Distribute the style by attaching an ID to each <body> tag of each page.

Based on OOCSS's idea, descendant selectors are short, we try to describe generic style as much as possible.

|  |  |
| --- | --- |
| **File name** | **Applicable area** |
| **# pagehome** | Top page |
| **# pagecontact** | Contact Us |
| **# pagecompany** | Company overview |
| **# pagesearch** | Search results |
| **# pagedetail** | Search Details |

1. **CSS hack**

CSS hacks should be kept with minimum use and thoroughly not to deviate from the style sheet syntax prescribed by W3C.

**W3C CSS** Validation Service

*http://jigsaw.w3.org/css-validator/validator.html.en*

1. **The basic style of CSS**

About application of style

· Use tab key indentation and line feed to describe CSS in an easy-to-understand manner.

· Set "CSS source format option" of Dreamweaver and customize it for easy reading based on the description

**Description example**

*body { font-size: 82%; color: #444; line-height: 137%; }*

1. **Abbreviation (short hand)**

If abbreviation of the property is possible, use abbreviations, keep in mind short reading source code in consideration of reading speed.

**Color specification (In the case of a Gala site, six digits are described properly)**

*.hoge {color: # 000;}*

**background image**

*.hoge {background: #FFF url ( "/ img / common / bg.gif") repeat-x 0 0;}*

**border**

.hoge {border: 1px solid # CCC;}

**Margin · Padding ※ In this case padding-bottom: 5px;**

*.hoge {*

*margin:12px 0;*

*padding:0 0 5px;*

*}*

1. **About unit**

The recommended units for each property are different. Please refer to the table below.

|  |  |  |
| --- | --- | --- |
| **Property** | **Unit** | **Remarks** |
| Font-size | *% Or px* | In principle, make the character size variable so that the user can specify it. When fixed in design px is also acceptable. |
| Line-height | *Unit none or px* | It is specified with the unit pear to be applied according to the character size.  Note that when you specify em, the influence on child elements will differ.  For LIR method of image replacement or px designation when fixed as design. |
| Background-Position | *% or px* | If you change the font size, such as when displaying an icon, you do not have to change the background-position specification  Because it is good, em designation. When design is necessary,% and px are also acceptable. |

1. **About character size**

Default font-size is 14px, with a line-height of 1.428.

Considering accessibility, the font size is basically variable.

Specify font size to be user variable in the form of %.

1. **About text color**

Pay attention to the contrast with the background color in consideration of accessibility.

Omitted things that can be omitted in the 3-digit hexadecimal notation *(example: # FF0000 → # F00)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Character state** | **CSS selector** | **color** | **Hexadecimal notation** | **decoration** |
| Normal | Body | black | # 333 or # 444 | --- |
| Link | *a:link* | Blue | # 06C or default | default |
| Visited | *a:visited* | Purple | default | default |
| Mouseover | *a:hover* | Blue | default | default |
| Active | *a:active* | Blue | default | default |

1. **Media queries**

We use the following media queries in our Less files to create the key breakpoints in our grid system.

*@media (max-width: @screen-xs-max) { ... }*

*@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }*

*@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }*

*@media (min-width: @screen-lg-min) { ... }*

1. **CSS Preprocessor**

CSS preprocessor is one way extension of CSS or also coi it is a language. Purpose of Easily CSS preprocessor is for you all the CSS structure, diminished time one right rewrite the code, easily applied OOCSS... it is also variable, class inheritance, and easily create a separate library for you to manage, inherit and reuse in other projects. CSS preprocessor but now there are a lot of today's most popular Less and Sass. Read more: http://sass-lang.com/

**Sass file structure of the basic**

*scss*

*|-- style.scss*

*|-- \_variable.scss*

*|-- \_helper.scss*

*|-- \_mixins.scss*

*|-- components*

*|-- -- \_header.scss*

*|-- -- \_footer.scss*

*|-- mixins*

*|-- -- button.scss*

*|-- -- label.scss*

In style.scss file contents are as follows:

*@import "variable";*

*@impprt "helper";*

*@import "mixins";*

*// Components*

*@import "components/header";*

*@import "components/footer";*

***Note****:*

*\_variable.scss: variable contains information about the shared variable*

*\_helper.scss: helper will contain / import the helper class*

*\_mixins.scss: mixins will contain / import the custom class or classes to be reused multiple times*

1. **CSS comments**

When describing the CSS of each category and page, describe where CSS is for CSS in comments.

Headers and footers, such as the following headlines are added for each step according to the content part, and the readability of CSS is improved.

It is desirable that an index list is placed at the head of the CSS file and the heading number is also written in the heading (large) and the heading (middle).

\* For details, see common.css and style.css of CB-STANDARD.

**Headline (Large)**

*/ \* =========================== ========================*

*1. style setting*

*============================= ====================== \* /*

**Heading (middle)**

*/ \* ------------------------------------------------------------------------ -------------------------------*

*1-1. Font color styles*

*-------------------------------------------------------------------------------------------------------- \* /*

**Heading (Small)**

*/ \**

*Page top styles*

*-------------------------------------------------------------------------------------------------------- \* /*

1. **Basic check items**

Be sure to check "common.css" "style.css" of CB-STANDARD before actually coding.

Since the common.css class to be used for general purpose has been prepared, based on the concept of **OOCSS (Object-Oriented CSS),** be appropriately utilized.

In CSS coding, style previously described in style.css is modified.

If the size of the site is somewhat large, describe style common to the site in style.css and prepare CSS for each page.

1. **Image replacement**

For image replacement by CSS, text-indent: -9999px is adopted.As a reason, when characters are searched by the browser, it is possible to prevent hidden characters from being displayed

It means that it is a method.

※ Since Google declares it to be deprecated, use image replacement with the minimum necessary.

**Description example of image substitution**

*#navi li a {*

*display: block;*

*overflow: hidden;*

*width: 136px;*

*height: 30px;*

*height: 30px;*

*background: #FFF url("/img/common/gn.gif") no-repeat;*

*text-indent:-9999px;*

*}*

1. **Transmission at CSS**

If you want to rollover images and buttons with links easily with CSS, if you specify class “:hover”

Opacity allows images to be transmitted. However, please do not transmit logo images in principle.

\* Since it is transmitting only to the last, if the background image overlaps behind the button image that you want to transmit, set the background color to the tag surrounding the image

Specify or avoid images by preparing rollover images.

Prepare and avoid images during lure

1. **Column specification**

Grid systems are used for creating page layouts through a series of rows and columns that house your content.

Read more <http://getbootstrap.com/css/>

Float: Float can be cleared by applying clearfix to the hierarchy above the box specifying Left / Right.

If you describe it as follows, you can easily create 2 column layout.

***2 column of sample***

*<div class = "row clearfix">*

*<div class = "col-md-6"> </ div>*

*< div class = " col-md-6"> </ div>*

*</div >*

1. **Cancellation of the float**

■ **Use clearfix**

By specifying .cleartfix of bootstrap.css as the parent element, it is possible to cancel the float.

**■ overflow: use hidden**

You can override the float with overflow: hidden on the parent element. There are cases where problems such as display being interrupted are occurred, but convenient,

Use with caution.