Blueprint css framework

Getting started

online resources

Blueprint Home: blueprintcss - Google Code[1]

- BlueprintCSS 101 | Blue Flavor[2]
- Hands on with Blueprint, a CSS Framework :: CSS Demos[3]
- Tutorial - blueprintcss - Goolge code[4]

blueprint 101 by blue flavor

Basics

core of BP does (on all browsers)

1. mass reset of browser defaults
2. set new defaults for basic stuff fonts and certain styles. (well scaling)
3. grid layout methodology
   (puts down by default a 24 col grid, with each col 30px wide + 10 gutter)
4. print media support

Grid Builder tool

Blueprint Grid CSS Generator[5]
(3rd party tool, also generates a grid.png file to use as a page background during development)

Interesting note about designer workflow

(For me not knowing the typical designer toolchain) first generating the grid and then making that into a photoshop grid rather then the other way around! See also some ref to a photoshop grid-generator: http://www.andrewingram.net/articles/gridmaker_reboot/

> NOTE
> /*to self*/ the mentioning of photoshop makes me think about having a spot for such 'off line' designer files in the module dir layout of the sources (svn) -- in an attempt to include such files in the code repository, making them a integral part of the project-development-source

Advise

To make sure you've seen _all_ that is in blueprint before thinking about adding stuff yourself just so you don't overlook.

Tips

- one can avoid unnecessary div elements
- one can have more then one .container (multiple ones will create different rows)

1. http://code.google.com/p/blueprintcss/
5. http://kematzy.com/blueprint-generator/

Comments

- similar initial feeling about 'semantic' classes versus layout driven ones. However you've gotta love this quote:
  - So you might have a div whose id=”secondaryContent” and class=”column” and you can wake up in the morning without feeling guilty about last night. Yes, you’ve compromised your rigid principles somewhat by including a presentational label, but you haven’t destroyed your page’s semantic structure. The important point is that you maintain your html’s integrity and do not create your grid with a sea of divs.
  - see also: http://www2.jeffcroft.com/blog/2007/aug/09/myth-content-and-presentation-separation/
  - NOTE
    thought in Kauri context:
    Eventually there might be a pipeline component on the server-side translating semantic classes into blueprint classes based on some configuration, or some template 'decorator' feature?
    That would bring the best of both worlds for markup that is not as much part of a template-xhtml rendition for the browser, but is really semantical markup from some CMS?
  - all in all looks like blueprint has less re-ordering capabilities then yui (for now at least)
- 2currently no liquid layouts (as of October 2007. might be solved already)
- all frameworks are bigger then needed: you send down some unused code! (total size is currently < 30k)
- yui css seems to require more div nesting with the yui-g, yui-u trick

hands on with blueprint

usage

linking in blueprint:

```
 <link rel="stylesheet" href="css/blueprint/screen.css" type="text/css" media="screen,
projection">
 <link rel="stylesheet" href="css/blueprint/print.css" type="text/css" media="print">
 <!--[if IE]><link rel="stylesheet" href="css/blueprint/lib/ie.css" type="text/css"
media="screen, projection"><![endif]-->
```

defining and filling a grid: (base grid has 24 columns)

```
   <div class="container">
       <div class="column span-24">Header</div>

       <div class="column span-16">Content</div>
       <div class="column span-8 last">Navigation</div>

       <div class="column span-24">Footer</div>
   </div>
```

some own testing

- print:
  - forms seem to print-preview badly (have to check with latest version later)
  - same remark for the message 'specials

other remarks

- structure stays a lot cleaner then with yuicss imho (no, or surely less divitis). The Non-semantical classnames argument easily applies to both solutions.
- unlike yuicss there doesn't seem to be a publicly hosted version of these css scripts (with URI's well-tagging the versioning)

> NOTE
> probably doesn't matter since kauri will probably provide some module packaging for both anyway.

tutiorial on google-code

various extra things learned

- structure
  - main starters:
    - screen.css
    - print.css
  - that are refering to lib/*
    - grid.css
    - typography.css
    - buttons.css
    - reset.css
  - compacted version for live deployment
    - lib/compressed.css
- typography uses a 18px baseline height. >> NOTE: images (and all other graphical inset stuff) should have heights that are a multiples of this unit. (see http://www.alistapart.com/articles/settingtypeontheweb)
- print.css alllows to specify your domain-name for correct display (with content: "yours" attr(href);) of otherwise relative urls
- remember to use the 'last' class on the last column on a given row (removes the trailing margin)
- columns can be nested.

scanning the source

based on svn checkout 2008-02-08

```
blueprint/src/grid.css
blueprint/src/ie.css
blueprint/src/forms.css
blueprint/src/grid.png
blueprint/src/typography.css
blueprint/src/reset.css
blueprint/src/print.css
```

reset.css

- very similar to yui, with nuances though (one wonders why there is not more allignment on this part)
  - specially get the feeling that blueprint version expects the 'typography.css' to be included as well (more then css expects 'base', in fact yui recommends not to)
- note thought the uinlike yui this doesn't set standard background-foreground colors

typography.css

- img floats left by default in <p> add .right on the image to let it float right
- specific text-type classes targetted to appearance:
  - .small, .large (note: large seems to break the vertical grid/ regular baseline spacing)
  - .hide
  - .quiet, .loud, .highlight
  - .added, .removed
- or to force proper (vertical) allignment: .top, .bottom

grid.css

- coulmns do seem to require their proper divs (you can't just leave them direclty on ul, p, ... but H* do seem to work ok)
- use class .showgrid to show the grid.png as a background
- classes to use
  - .append-N, .prepend-N to post- resp. pre-fill with empty space (padding)
  - .span-N to take up the required space
  - .border: right-hand side of column gets a border,
  - .colborder: will add an additional column of padding
  - .pull-N: pull column to previoous (left) column
    .push-N: push column to next (right)
- other know-worthy
  - hr versus hr.space (last one takes same space but doesn't show the line)
  - .clearfix
    - included general float 'n clear fix described here: http://www.positioniseverything.net/easyclearing.html

print.css

nothing shocking here

ie.css

interesting here again to see somewhat of a coupling/expectation: ie.css mixes redefining stuff from both the reset.css and the grid.css

fancy-type.css

Provides extra classes for advanved text manipulation
Includes

- indentation of sibling paragraphs
- .alt
- .dquo for special looking double-quote in titles
- .incr for sidenotes
- .caps for small-caps

buttons.css

Odd thing to note: this plugin is outside the svn-trunk, but nicely included in the release zip.

- Gives you great looking CSS buttons, for both <a class="button"> and <button>.

    - including positioning of embedded images (button-icons to show)
    - adds classes .positive and .negative for different coloring for confirming resp. denying actions
- Demo: http://particletree.com/features/rediscovering-the-button-element/

Other sources

Appraisal

- http://ajaxian.com/archives/blueprint-css-framework-typography-matters