Plone SOLUTIONS

Plone Skinning and User Interface

Alexander Limi
Plone Solutions

Plone Symposium - July 20th, 2005

Presentation structure

- HTML and CSS primer
- Plone Skinning & tools
- Plone 2.1 differences
- Plone UI advice
- Q&A

Social structure

- *Please* – ask questions along the way!
- A couple of breaks
- Ample time for Q&A + Feedback at the end
- JIT presentation, meant to be informal

HTML primer

- Plone uses XHTML
- All tags are closed
  - Singleton tags: <img />
- All attributes quoted
  - href="http://plone.org"
- Singleton attributes:
  - selected="selected"

CSS primer

- CSS works on IDs and classes
  - CSS: #someid
  - HTML: <span id=”someid” />
  - CSS: .someclass
  - HTML: <span class=”someclass” />

Plone Skinning

- CSS-intensive
- 2.1 changes the game a bit
- Parts of the process covered by Joel in his talk earlier today

ResourceRegistries

- Included with Plone 2.1
- Allows multiple CSS/JS files to be combined into one file
- Allows TAL conditions
- Allows third-party products to register global CSS/JS

The 2.0 approaches

- CSS only, overriding existing selectors
  - Complex!
- Creating your own CSS
  - Massive overhead, a lot to maintain
- Creating your own templates
  - Useful when you are creating a totally new layout, or have public/private layouts

2.1 changes this

- CSS only, ~~overriding existing selectors~~
  - ~~Complex~~! Managable!
- Allows you to turn off *parts* of the Plone CSS
- (Also true for JS - want to get rid of the link globe?)

Plone SOLUTIONS

2.0 structure

- Mammoth plone.css (1558 lines!)
- Separate rendered files for
    - Print
    - Presentation
    - Mobile

2.1 structure

- plone.css split up in 12 files
- One style sheet for everything
  - Combines print/presentation/mobile
  - Reduces file transfer overhead
  - Caching-friendly!
- Old ploneCustom.css approach still works

Plone Themes

- Conceptual name
- Still implemented as skins, but needed to separate the concerns
- Themes are *only* about visual changes, not functional changes

Implementing Themes

- Plone product, as usual
- Uses ResourceRegistry API to disable parts of Plone CSS
- QuickInstaller keeps track of install/ uninstall
- ExampleTheme product available (soon!)

ResourceRegistry

- Simple demo

Tools

- Plone CSS is insane! (Size-wise at least ;)
- Get Firefox / Mozilla
  - Web Developer Extension
  - DOM Inspector

Web Developer Ext

- For Mozilla / Firefox
- ChrisPederick.com
- Essential if you do any Plone work

Web Developer Ext

- Demo

DOM Inspector

- Allows you to inspect CSS
- All the gory details
- Perfect for debugging

DOM Inspector

- Demo

BREAK

- Next up: Plone User Interface Guidelines

Plone User Interface

- Best Practices
- General Web and User Interface guidelines
- How to identify UI patterns
- Reusable Plone elements
- It's the mythical Plone Style Guide!

Links styling

- Commonly abused
- Underline
  - Missing
  - Dotted
- Color!

Underlined links - when?

- Body text: Always!
  - You can prettify it, but it's a basic premise of the web
- When it's obvious that the link is a navigational device: Not necessary
- Use visual sensibilities - too many underlines in one area ruins readability

Link color

- Blue color for links is a good idea
  - At least in body text
- Keep the visited/active color difference!
  - Purple/Gray

Underline on non-links

- Never use the <u> (underline) tag
  - It is even deprecated in XHTML
- Historically, underlines were the "poor man's bold/italics" – easier to do on cheap printing presses
- Just Say No, Kids!

Workflow state colors

- New in 2.1
- A very visual way of getting a quick overview
- Never rely *only* on colors only, though!
  - (Color blindness is common, especially among males - as witnessed here ;)

Workflow state colors

- Never in body text (remember? ;)
- Only visible for logged-in users
- Used in
  - Portlets
  - Listings

Workflow state colors

- Combined with the site map:
  - Instant visual security/state inspection of your site!
- Thoroughly changes the way you perceive your content workflow
- Careful with colors that are too alike or outrageous

Class/ID naming

- Don’t do this, *please*
- <span id = “red” />
  - Use semantic naming - what does the item represent? What is it?
- <a class = “link” />
  - Use the tags themselves! Avoid class-itis

Plone UI Widgets

- Reusable elements help the UI experience
- Solving problems other people have put time into
- Archetypes widgets help here too
- Steal!

Listings

- class = “listing”
- Used to present tabular data
- Pet peeve: Do not put actions on each row!
  - Delete/Copy/whatever
- Batchable actions? Make it selection +

Vertical listings

- New in Plone 2.1
- class = “vertical listing”
- Ideal for label + info representation
- Used for e.g. Event information in 2.1

Portlets

- New <dl>-based structure in 2.1
- Summarized information/listings
- Title + additional info on each item

Inline portlets

- As seen in Plone Help Center
- Useful when you want to group rapidly changing listings

Forms

- Labels
- Fieldsets
- Buttons
- Widget comparison

Labels

- <label for=”id”>
- Required for stuff to be clickable
- Good for accessibility
- Convenient for everyone
- Especially important for radio buttons

Fieldsets + Legend

- <fieldset> +<legend>
- Grouping of similar inputs
- Example: Address book card
  - Personal Info
  - Company Info
  - Related contacts

Buttons

- 4 classes in Plone
  - “context” – when other input is required
  - “standalone” – can be pressed independently
  - “destructive” – for deleting/removing
  - “search” – for search-based actions

Widget shoot-out

- There are variations on the same widget
- What version is most appropriate in which cases?

Round 1

- Checkbox vs. Multiple Selection list
  - Checkbox for less than 5±2 options
  - Multiple selection for more
  - There are exceptions!
  - Checkbox is preferred, much better device all over

About checkboxes

- Always affirmative!
- Don’t do:
  - “Check this box to *not* do X”

Round 2

- Pulldowns vs. Radio Buttons
  - Radio buttons when 5±2 options
  - Pulldowns when more
  - Radio buttons are much more explicit, can also contain much more information
  - Radio buttons are preferred in most cases

Table of Contents

- New in Plone 2.1
- For giving an overview of the document
- Used for PLIPs right now
- Meant to be a standard structure in future versions

Interactive session

- Going to show some real-world cases
- Input and suggestions very welcome

- Designing an effective template
- Designing an effective widget

Designing an effective template

- CMFCollector
- Simple issue tracker
- Minor modifications made using it a lot more pleasant
- 1-2 hours real work
- Walkthrough

Designing an effective widget

- Multiple select widget for 20 countries
  - How do we do this efficiently?
- Sometimes it's better to show *all* the information – a 4x5 grid of checkboxes can be a good approach here if country list doesn't change