ARROW CONSULTING & DESIGN

Skinning 101

Skinning for DotNetNuke

About Ralph Williams

Ralph Williams is a UX/Graphic Designer & Web Developer at Arrow Consulting & Design.

He has presented at multiple Code Camps, Day of DNNs, and DNN World 2011& 2012

Created Awesome Cycles site template DNN 6.

Ralph's blog can be found at www.RalphWilliams.com and is featured on www.DotNetNukeBlogs.com

ARROW

Topics Covered

1. Setting up your skinning environment
2. Introduction to Skinning (Skinning Basics)
3. Creating a Skin
4. Skin objects
5. Skinning the DDR menu
6. Creating Containers
7. Creating and testing a Skin package

Setting up Skinning Environment

- Create a local installation of DotNetNuke
  - Microsoft WebMatrix *(easiest)*
  - MakeDNNSite by Mike van der Meulen
    http://mikevdm.com/dnn
  - Manually (see DNN site for tutorials)

ARROW CONSULTING & DESIGN

Walkthrough of Site Setup

Introduction To Skinning Basics

DotNetNuke Skin:

A skin is the design applied to the standard DotNetNuke portal to change the layout and design. Each Skin will define Panes, locations where you can place modules onto a DotNetNuke page.

Introduction To Skinning Basics

Content Pane:

A location within the skin that defines where content will be placed. Each skin requires at least one content pane named “contentpane”.

Introduction To Skinning Basics

Module:

An extension of the DotNetNuke framework that provides functionality that the users can perform. Modules are what makes up a DotNetNuke website's content.

Introduction To Skinning Basics

Containers:

Containers frame individual modules and can be applied to one or more modules, or can be set globally across all modules.

Introduction To Skinning Basics

Skin Package:

A skin package is a collection of individual skin files, a DNN manifest file, and any associated images, stylesheets, and JavaScript.

Introduction To Skinning Basics

DNN Manifest File:

A Skin Manifest is a configuration file that tell the DotNetNuke Installer how to handle items (or properties) during the Extension installation process.

Skin folder structure

Root
- Portals
  - _default folder (host level)
  - [portalid] (site level)
    - Skins
      - Skin Name
        - Skin.ascx
        - Skin.css
        - Skin.doctype.xml
    - Containers
      - Container Name
        - Container.ascx

CSS Hierarchy

Normal mode

- Default.css
  - Reset.css
  - Typography.css
- Module.css
- Skin.css
  - CSS/Typography.css
  - CSS/Content.css
- Container.css
- Portal.css (sort of)

Admin mode

- Default.css
  - Reset.css
  - Typography.css
- Admin.css
- Module.css
- Skin.css
  - CSS/Typography.css
  - CSS/Content.css
- Container.css
- Portal.css (sort of)
- ControlBar.css

ARROW CONSULTING & DESIGN

THANK YOU.