Building for the Future

Developing a Maintainable Code Base

Presented by Kevin Andrews

kandrews@industrialagency.ca

industrial.

Building for the Future: Developing a Maintainable Code Base

What is the problem?

How do we solve it?

Topics in this presentation

Create a Standardized Environment

Manage Wordpress Core and Plugins

Compile Assets with a Taskrunner

Write Clean Code

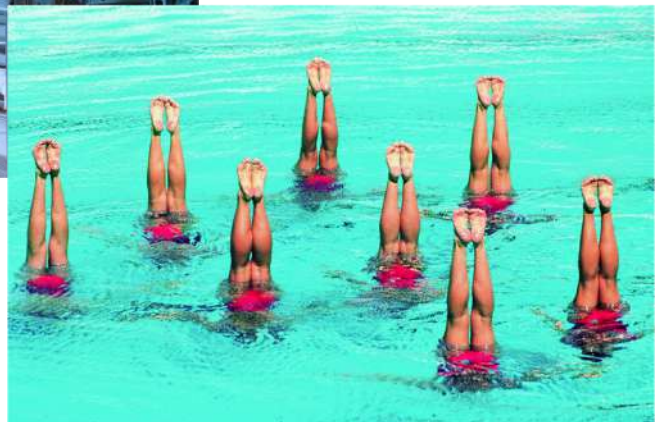
https://github.com/KAndrews101/Wordpress-Baseline

https://github.com/KAndrews101/Wordpress-Baseline

What do you mean by “Standardized Environment”?

Project setup is always the same

Your team will be in sync



Less time fixing bugs

https://github.com/KAndrews101/Wordpress-Baseline

Tools we can Leverage

https://www.vagrantup.com/downloads.html

https://www.virtualbox.org/wiki/Downloads

https://downloads.chef.io/chefdk

https://roots.io/trellis/

https://github.com/KAndrews101/Wordpress-Baseline

The Benefits of Using a Virtual Machine

- Fast onboarding of new developers
- Reduces repetition
- Mimic hosted environments
- Identical environment for everyone
- Time spent up front is time saved later

https://github.com/KAndrews101/Wordpress-Baseline

The Drawbacks

- Initial setup can take some time
- Requires sufficient RAM on your machine
- Have to host a staging environment
- Team has to maintain their own system dependencies(Vagrant, VirtualBox, Chef)

https://github.com/KAndrews101/Wordpress-Baseline

Create a Standardized Environment

HashiCorp Vagrant

VirtualBox

CHEF

Your Wordpress Project > cookbooks > wordpress > recipes > default.rb

```
execute "install_nginx" do
        user "root"
        action :run
        command "sudo rpm -Uvh
        http://nginx.org/packages/centos/...
end
```

Your Wordpress Project > Vagrantfile

```
Vagrant.configure(2) do |config|

    config.vm.box = "bento/centos-7.4"
    config.vm.hostname = 'wordpress'

    config.vm.network :private_network, ip: '172.16.231.130'
    config.vm.network :forwarded_port, guest: 80, host: 8080
    config.vm.network :forwarded_port, guest: 9200, host: 9201

    config.vm.provider "virtualbox" do |v|
        v.memory = 2048
    end

    config.vm.provision "chef_solo" do |chef|
      chef.run_list = [
        "recipe[wordpress]"
      ]
    end
end
```

Vagrant Commands

vagrant up

vagrant ssh

[vagrant@wordpress ~]$ cd /vagrant

vagrant reload

vagrant destroy

vagrant global-status

vagrant provision

https://github.com/KAndrews101/Wordpress-Baseline

Manage Wordpress Core and Plugins

https://github.com/KAndrews101/Wordpress-Baseline

What's the Problem?

Updating Wordpress and plugins through the admin panel is dangerous!

Bedrock

https://github.com/KAndrews101/Wordpress-Baseline

Download and Configure Bedrock

Roots Bedrock

https://github.com/roots/bedrock

Your Wordpress Project

src

.env

```
DB_NAME=wordpress
DB_USER=root
DB_PASSWORD=root

WP_ENV=development
WP_HOME=http://172.16.231.130
WP_SITEURL=${WP_HOME}/wp
```

https://github.com/KAndrews101/Wordpress-Baseline

Install and Manage Project Dependencies with Composer

https://getcomposer.org/download/

```
Your Wordpress Project$ vagrant ssh

cd /vagrant/src && composer install
```

https://wpackagist.org/

```
composer require wpackagist-plugin/plugin-name
composer remove wpackagist-plugin/plugin-name

composer update
```

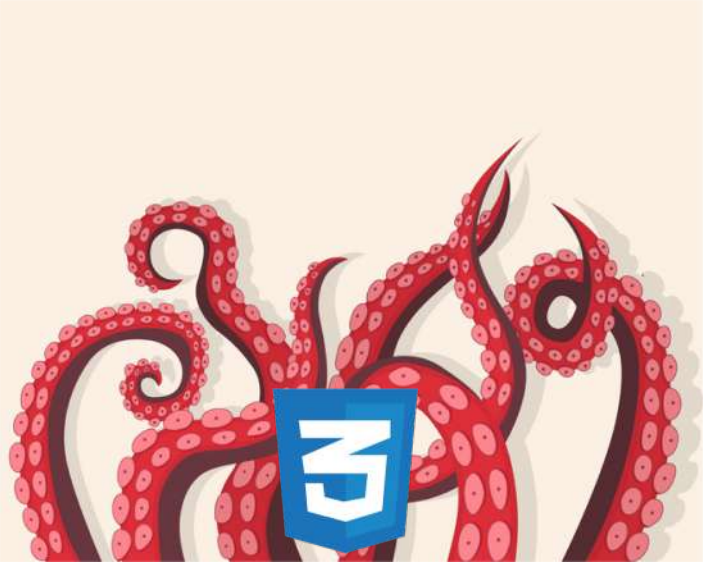
Your Wordpress Project > src > composer.json

```
"roots/wordpress": "5.*",
"roots/wp-config": "1.0.0",
"roots/wp-password-bcrypt": "1.0.0",
"wpackagist-plugin/custom-post-type-ui": "*",
"wpackagist-plugin/simple-page-ordering": "*",
"wpackagist-plugin/ninja-forms": "*",
"wpackagist-plugin/wordpress-seo": "*"
```

Compile Assets with a Taskrunner

https://github.com/KAndrews101/Wordpress-Baseline

This is a PSA. STOP Using CSS. START Using a CSS Preprocessor.



https://github.com/KAndrews101/Wordpress-Baseline

Sassquatch.ca

https://github.com/KAndrews101/Wordpress-Baseline

Installing Development Dependencies

https://nodejs.org/en/download/

```
npm install gulp-cli -g
```

package.json

```
{
  "devDependencies": {
    "dotenv": "^4.0.0",
    "gulp": "^3.9.0",
    "gulp-autoprefixer": "^4.0.0",
    "gulp-concat": "^2.6.0",
    "gulp-notify": "^3.0.0",
    "gulp-plumber": "^1.1.0",
    "gulp-rename": "^1.2.2",
    "gulp-sass": "^3.1.0",
    "gulp-uglify": "^3.0.0",
    "gulp-wrap": "^0.15.0",
    "gulp-insert": "^0.5.0"
  }
}
```

```
npm install
```

```
npm install gulp-sass --save-dev
```

https://github.com/KAndrews101/Wordpress-Baseline

Building and Running Gulp Tasks

gulpfile.js

```
var // Package Variables
    dotenv = require('dotenv').config({path: '.env'}),
    gulp = require('gulp'),
    sass = require('gulp-sass'),
    uglify = require('gulp-uglify'),
    autoprefixer = require('gulp-autoprefixer'),
    concat = require('gulp-concat'),
    rename = require('gulp-rename'),
    plumber = require('gulp-plumber'),
    notify = require('gulp-notify'),
    insert = require('gulp-insert'),
    // Environment Variables
    srcPath = process.env.SRC_PATH,
    assetPath = process.env.ASSET_PATH,
    styleName = process.env.STYLE_NAME,
    scriptName = process.env.SCRIPT_NAME;
```

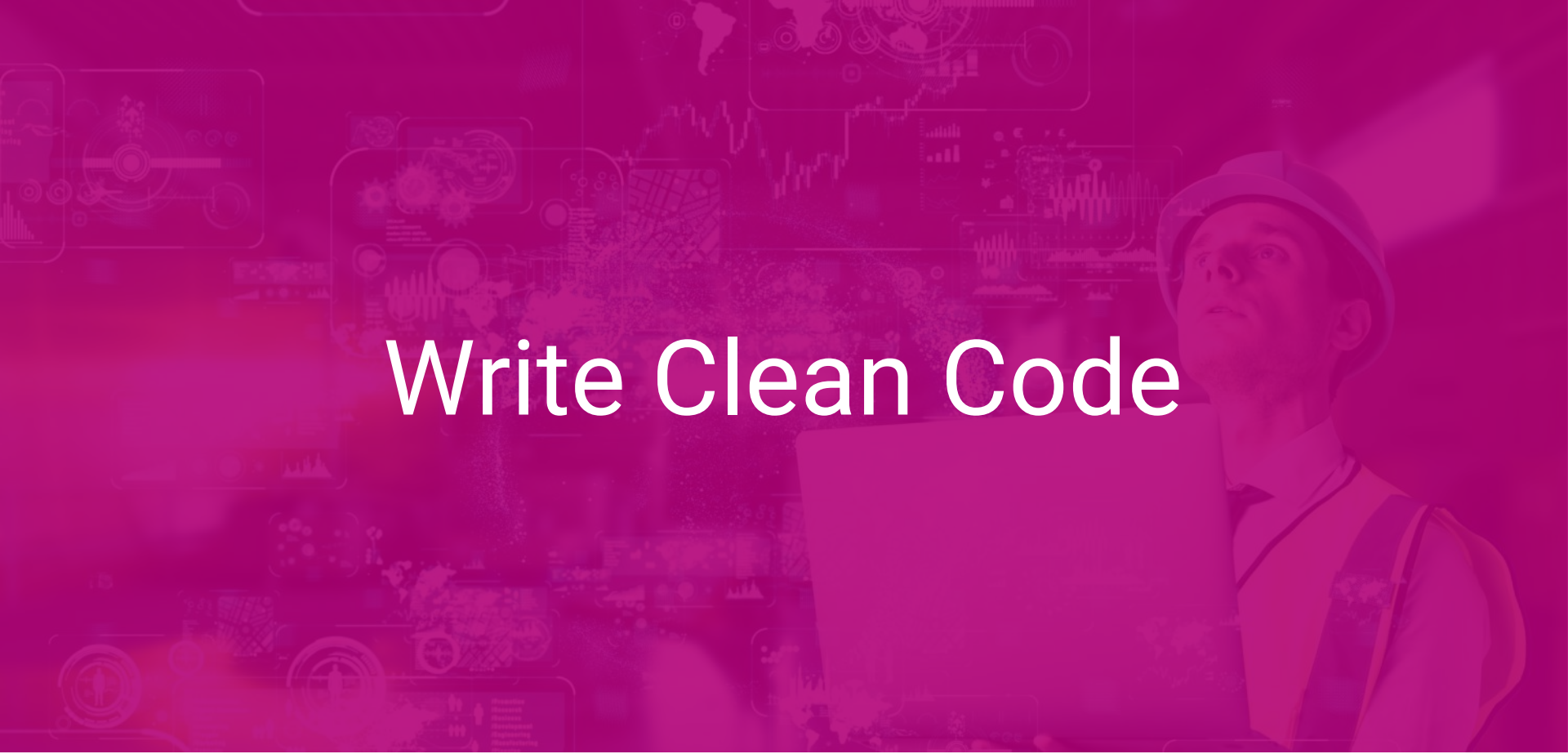
```
gulp.task('default', ['sass', 'scripts', 'watch']);

gulp.task('build', ['sass', 'scripts']);

// Compiles both unminified and minified CSS files
gulp.task('sass', function () {
  gulp.src(srcPath + assetPath + '/styles/' + styleName + '.scss')
    .pipe(plumber())
    .pipe(sass({
      outputStyle: 'expanded',
      includePaths: ['node_modules']
    }))
    .on('error', onError)
    .pipe(autoprefixer({
      browsers: ['last 100 versions'],
      cascade: false
    }))
    .on('error', function (err) {
      console.log(err.message);
    })
    .pipe(sass({
      outputStyle: 'compressed'
    }))
    .pipe(rename(styleName + '.min.css'))
    .pipe(gulp.dest(srcPath + assetPath + '/styles/min'));
});
```

Run your tasks

```
gulp
gulp sass
```

https://github.com/KAndrews101/Wordpress-Baseline

Write Clean Code



https://github.com/KAndrews101/Wordpress-Baseline

What can we do to Improve the Quality of our Code

- Be as modular as possible
- Limit use of plugins
- Standardize syntax
- Document your code

https://github.com/KAndrews101/Wordpress-Baseline

Making Your Code Modular

- The Theme
  - assets
    - styles
      - components
        - theme.scss
    - scripts
      - min
        - animations.js

theme.scss

```
/* THEME COMPONENTS
------------------------------------------------------------------------*/
@import "components/lists";
@import "components/tables";
@import "components/buttons";
@import "components/widgets";
```

Use a CSS Naming Convention

http://getbem.com/naming/

```
.button{
        &__icon{
                }
        &--has-icon{
                }
}
```

```
.button__icon{ }
.button--has-icon{ }
```

https://github.com/KAndrews101/Wordpress-Baseline

Making Your Code Modular

- The Theme
  - partials
  - functions
  - page-templates
  - post-templates

```php
function get_post_type_templates($single_template) {
  global $post;

  $single_template = dirname( __FILE__ ) . '/post-templates/single-' . $post->post_type . '.php';

  return $single_template;
}
add_filter( "single_template", "get_post_type_templates" );
```

functions.php

```php
$partials = WP_CONTENT_DIR . '/themes/sassquatch/functions/';

require_once $partials . 'config.php';
require_once $partials . 'widgets.php';
require_once $partials . 'svg.php';
require_once $partials . 'appearance.php';
require_once $partials . 'menus.php';
```

front-page.php

```php
<?php get_template_part('partials/latest-news'); ?>
```

https://github.com/KAndrews101/Wordpress-Baseline

Limit Use of Plugins

Too Many Plugins

- Difficult to keep track
- Slows down your site
- May create redundancies
- Increased chance of conflicts

Audit any plugins you do use

https://github.com/KAndrews101/Wordpress-Baseline

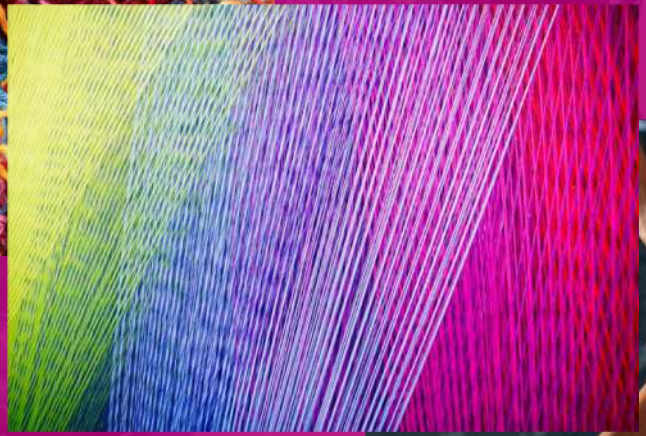
Document Your Code

Without documentation, people will get lost

- Readme.md
- Inline Comments
- Hosted webpage
- Well written code can document itself

https://github.com/KAndrews101/Wordpress-Baseline

Conclusion



Presented by Kevin Andrews

kandrews@industrialagency.ca

industrial.

Thank you!

Presented by Kevin Andrews

kandrews@industrialagency.ca

industrial.

Questions?

Presented by Kevin Andrews

kandrews@industrialagency.ca

industrial.