DrupalCON 2016 Notes

By Ajinkya Kulkarni

Date & Location: May 9 to 13 at The Ernest N. Morial Convention Center, New Orleans, LA 70130

Stats from the closing session:

- 3102 attendees
- First time attendees: 33%
- Most popular first names
  - Amy (20)
  - Chris (64)
- Diversity:
  - Female: 23%
  - Male: 75%
  - Other: 0.3%
  - Declined to share: 1.7%
- Speakers
  - Female: 19%
  - Male: 81%
- Peak Day:
  - Tuesday at 10 am
    - 2164 devices online using 370 Mbps
- 37,519 steps per day walked
- Where do we come from? (top 10)
  - **United States**
  - **Canada**
  - **India**
  - UK
  - Germany
  - Ukraine
  - Brazil
  - Japan
  - Argentina
  - France

Keynote – Dries Buytaert (http://buytaert.net)

- Drupal 8.0 – released on 11/19/15
- 3300+ people contributed to Drupal 8 vs. 954 people in Drupal 7
- **Drupal 8.0 features**
  - Improved authoring experience
  - Mobile optimized
  - Modern PHP standards
  - Configuration management
  - Better caching
  - Web services APIs

    - Translation in 100 languages
    - 200+ more features
- **New features in Drupal 8.1** – released on 4/20/16
    - Experimental UIs for migrations
    - Composer support
    - Spellcheck support in CKEditor
    - JavaScript testing
    - BigPipe for perceived performance
    - Improved help and documentation
- Frequent releases with new features & backwards compatibility
    - Every 6 months
    - No need to wait for 4 and half year to get your contribution out
- **Feature direction**
    - Richness – How many capabilities it has
    - Reach – How many people use it
    - Who should we favor when making product decision – from survey of 2900 people
        - Content Authors – 46%
        - Site Builders – 29%
        - Back-end Developers – 13%
        - Front-end Developers – 12%
    - Customer experience (CX)
    - **Developer experience (DX)– 25%**
    - **Editorial experience (EX) – 75%**
    - Vision – Build the leading platform to assemble the world's best digital experiences.
        - Not just for websites, also for mobile apps
    - New feature ⟵ Vision + Survey data + Discussion + Market
    - **Proposed Initiatives**
        - **Media** (EX) – Authors and editors need simple drag-drop media and asset handling
        - **Blocks and Layout** (EX) – Site builders need intuitive tools to build pages, change layouts, and add & arrange blocks with live preview
        - **Data Modeling Tools** (EX) – You are a site builder making a site for foodie meetups where you share and discuss recipes. You want to be able to easily create an event content type with associated recipes.
        - **Fully Object Oriented API** (DX) – This is one of the most valuable things to do right now, but not an initiative.
        - **API-First** (DX) : Integrate with other systems, bring your content anywhere, display it as you wish.
            - Web services in Drupal 8 are currently scattered across diverse set of feature set
                - Core REST modules
                - Relaxed Web Services
                - Services
                - Entity Query API
            - We need a full feature set and unified REST API in Drupal Core
            - Comparison at http://buytaert.net/an-overview-of-web-service-solutions-in-drupal-8
        - **Theme Component Library** (DX): Convert Drupal's complex combination of templates and huge render arrays into a simple, structured tree of reusable components.
        - **Cross Channel** (CX)

            - Becomes webless – Internet of Things, Notification, Mobile Commerce
            - Conversational interfaces – Amazon ECHO
            - Better digital experiences via Contextualization
            - From
                - Websites & web content → Cross-channel digital experience
                - Browser-based → Any screen, device, application
                - Everyone sees the same content → Personal, contextual experiences
                - Backend tied to front-end delivery → Decoupled architecture for flexible delivery

        - **Orchestration** (CX)
            - Taking customer experience beyond the page
                - Updates in Drupal
                - Notification
                - Mobile Commerce
                    - Orchestration
                        - Context
                        - Frequency
                        - Channel
                        - Preferences
                        - Context
                - Editing content beyond pages – audio messages, notification

    - **Planned Initiatives**
        - **Workflow**– Authors and editors need easy-to-use tools to share, review, approve, stage and collaborate on content before it's live
    - **Active Initiatives**
        - **Migrate** – Migrate content between two Drupal sites
- Happening Today
    - Drupal is powering all of the MTA kiosks in New York
    - Telsa powers their app using Drupal 8
    - Lufthansa using Drupal 8 to power in-flight system.
    - Drupal 7 adoption is going down and Drupal 8 adoption is increasing
- We are playing the long game, and we will win
- Slides (162 MB) - http://buytaert.net/files/state-of-drupal-may-2016.pdf
- Recording - https://youtu.be/Ksp5JVFryEg

How to implement performance metrics and dashboards for your digital agency – Casey Cobb

- What gets measured gets managed
- Three signs of a miserable job
    - Immeasurable
    - Irrelevance

  - Anonymity
- Don’t put too much pressure on one side of the wall
  - Realize that people are usually doing the best they can. There might be deeper reasons as to why a metric isn’t being achieved
- Metrics should promote discussion and communication to a solution
- How to think about developing metrics and a dashboard
  - Start simple if possible
  - Metrics are about improvement, not blame
  - Break everything down to the lowest common unit
- Small inefficiencies can have compounded & exponentially toxic effects on
  - Your team’s happiness
  - Your work product
  - Your company profitability
- What is Accidental Evil? – When individuals make decisions that incur costs that the individual doesn’t have to pay for. https://medium.com/@caseycobb/when-accidental-evil-makes-your-job-miserable-2bfd19ff7410#.9m7u9je9v
- What should we measure?
  - What’s not working right now?
  - What keeps you up at night?
  - What’s simplest to measure?
    - Ticket burn %
    - Tickets with AC, estimates, user_story
    - Zombie tickets
    - Ticket re-opened rate
- Interesting Realizations
  - We were able to improve our estimate accuracy by digging into the data and correlating accuracy data with estimation
  - Swiss Cheese model of fault tolerance
  - How can we reinforce our culture and values through metrics
- Project Management Metrics
  - Problem tickets across all projects
  - Status of tickets in workflow
  - Team allocation across all projects
  - Sprint tidiness
  - Sprint completion %
- Business Development
  - Calendar accuracy – timelyapp.com
  - Pipeline into the future
  - Social media metrics
    - Revenue targets being achieved?
    - Worked vs. Scheduled (project and team)
    - PM Billable vs Project %
    - Team feedback per week
    - Accidental Evil per week
- Dashboard tool – https://www.periscopedata.com/
- Coaching for growth and improvement https://www.manager-tools.com
- Slides - https://events.drupal.org/sites/default/files/slides/Drupalcon_2016_Metrics_Talk.pdf

- Recording - https://www.youtube.com/watch?v=q7gopctEBiE

Grow Your Own – Jody Hamilton, CTO & Co-founder of Zivtech

- Lack of available talent to hire is a common refrain of business owners in our field
- Give up on looking and complaining. To create a sustainable business, you need to be able to grow your own
- Growing your own means hiring smart, motivated people with all the right soft skills and investing in them for the long haul
- In return for helping give them a career, expect that they'll reward you with loyalty, will care about helping to teach your newer staff, and will work together with a cohesive vision
- What you Gain
    - Retention and loyalty
    - Close, harmonium team
    - Developer with other skillsets
    - Culture of training
- Ingredients
    - An office
    - Willingness to teach
    - Empathy
    - Time
- On-Ramps
    - QA
    - Documentation
    - Site building
    - Training materials
    - Automated test writing
    - First level support
- Quality First
    - Code Review
    - Feature Branches and Pull Requests
    - Billing Levels
    - Estimates and Discounts
- How others are doing?
    - http://www.metaltoad.com/blog/metal-toad-open-source-internship-model
    - http://www.curveagency.com/blog/drupal-apprenticeship-mentoring
- Teachable Moments
    - Focusing on the current
    - Requires availability and time
    - Honors the value of what comes up naturally
- Mentor Driven Development
    - Learn by teaching
    - Junior staff can create documentation and training materials
    - For sustainable growth, most important lesson for jr devs is how to teach others
- **Teaching as Leadership**
    - High ROI activity
    - Value, encouragement and praise teaching in your team

    - No, everyone is not an autodiadact
- **Let Your People Grow**
    - No one going to be just like you
    - Find individual's strength
- **What to Avoid**
    - Divas and Lone wolves
    - Unpaid interns
    - Contractors
    - Cutbacks
- Diversity and the Talent Pool
    - Most of the talent pool will not apply to your job listing
- **Where to Recruit**
    - Bootcamps
    - Meetups
    - Public trainings
    - Friends of employee
    - Colleges
- **How to Evaluate**
    - Soft skills are hard to teach
    - Intellectual honesty
    - Teaching ability
    - Learning trajectory
    - Strengths outside our field
- Recording - https://www.youtube.com/watch?v=JWTMfT3vIsE

"New and improved …" Selling the value of Drupal 8 – Jeffrey McGuire (Acquia)

- Drupal 7 was a huge success
- Drupal powers 5% of all website with identifiable CMS
    - This is 2% of all websites
- Drupal.org has more than 1.1 million user accounts
    - And 37,000 developer accounts
- The Ultimate Guide to Drupal 8 - https://www.acquia.com/sites/default/files/library/attachment/ultimate-guide-drupal-8.pdf
- Language Support
    - 4 pillars in Drupal 8
        - Language – Base services for all modules dealing with data. Not just multilingual
        - Interface – Interface translation has build-in update feature, improved usability
        - Content – Field translation in-built API for all entities. Content translation module provides user interface.
        - Config – Common configuration system handles blocks, views, field settings. Unified translation.
    - **Mobile first**
        - HTML 5
            - Mobile Optimization

    - Built-In Video/Audio Playback
    - Offline Caching
    - Browser Cross-Compatibility
    - Cleaner Code
  - Responsive out of the box
  - Mobile first admin and editing
  - Web API-driven
- **Better accessibility for 285 million people**
  - WAI-ARIA markup
  - Aural alerts
  - Accessible Inline Forms Error
  - Controlled Tab Order
  - Alt Text
- **Scalability and speed**
  - More precise caching
  - Client side caching
  - Dynamic content substitution
  - RefreshLess - https://www.drupal.org/project/refreshless
  - PHP Performance – lower the better
    - PHP 4.4 – 12.68 sec
    - PHP 5.0 – 12.54 sec
    - PHP 5.1 – 4.68 sec
    - PHP 5.2 – 4.2 sec
    - PHP 5.3 – 2.91 sec
    - PHP 5.4 – 2.18 sec
    - PHP 5.5 – 2.03 sec
    - PHP 5.6 – 1.92 sec
    - PHP 7 – 1.25 sec
    - HHVM 3.3.0 – 0.38 sec
  - BigPipe – 0.25 sec
    - 6.3 sec with out BigPipe
- **Proudly Invented Elsewhere**
  - Backbone.js
  - Symfony 2
    - Yaml
    - Routing
    - Validator
    - Serializer
    - HttpKernel
    - ClassLoader
    - HttpFoundation
    - EventDispatcher
    - DependencyInjection
  - Composer
  - EasyRDF
  - Doctrine
  - Assetic
  - Twig

  - Guzzle
  - jQuery
  - PHPUnit
  - jQueryUI
  - CKEditor
  - HAL/REST
- **Twig : Easy and safe templating**
  - Better security
  - Easier for themers
  - Reusable template files
  - IDE integration
- **Usability: WYSIWYG, Inline Editing**
- **Views in core**
  - Content listing
  - RSS / Atom feeds
  - Slide show
  - Rotating banners
  - Tabular listing
  - Search results
  - REST API endpoints
- **Configuration management**
- **Everything is entities and fields**
  - e.g. Comments are fields
  - Semantic field types
    - Email
    - Link
    - Phone
    - Entity reference
    - Date
  - SEO / schema.org
  - Semantic versioning
    - Faster, predictable innovation cycles
    - New significant features allowed in between big releases
    - Release minor versions on a timed cycle
  - Drupal 8.1
    - BigPipe
    - Migration modules
    - CKEditor Language and Spelling button
    - Support for automated JavaScript testing
    - More flexible admin/help page
  - Drupal LTS releases
- Recording https://www.youtube.com/watch?v=rl5yQpKynSE

Altering, Extending, and Enhancing Drupal 8 – Joe Shinderlar (drupalize.me)

- **Don’t hack core**
    - Use a hook
    - Write a plugin
    - Create a service
    - Subscribe to an event
    - submit a bug request
- **Let other change your module without hacking**
- Plugins
    - Provide configurable functionality in a way that allows either the system or the end user to choose from a provided list
    - Examples
        - Blocks
        - Field types
        - Field formatters
        - Views styles
        - Actions
    - Plugin Recipe
        - Determine the type of plugin
        - Is there a base class I can extend?
        - Implement the interface, and provide metadata so that plugin manager can find your instance
        - Plugins that perform similar functionality are of the same plugin type
            - You can implement your own new plugin types
                - See DefaultPluginManager
                - Choose a discovery method
                - Determine factory to use for initialization
                - Define an Interface and BaseClass
- Services
    - Because they are decoupled
    - Services can be swapped
    - Easier to test
    - Better reusability
    - Examples
        - Caching & Database Access
        - Assess (JS/CSS) optimizer
        - Plugin Manager
        - API Integration
        - Validation handlers
    - Recipe
        - Choose a unique name
        - Define an Interface and then implement it
        - Register the service in a mymodule.services.yml file
        - Access it via the container using the unique name \Drupal::service (‘mymodule.services_name’)
- Events

    - React to application actions/conditions without modifying the application itself.
    - Events are a common pattern in OOP, hooks are a bit of a Drupalism.
    - Event Recipe – Subscribe
        - Determine the event name
        - Define a service tagged ‘event_subscriber’
        - Implement \Symfony\Component\EventDispatcher\EventSubscrioberInterface in the \Drupal\mymodule\EventSubscriber namespace
    - Event Recipe – Dispatch
        - Add a class in the Drupal/my_module/Events namespace.
        - Add a class for event objects that extends \Symfony\Component\EventDIspatcher\Event provides additional information about an event.
        - Use \Drupal\Component\EventDispatcher\ContainerAwareEventDisatcher::dispatch(); and pass it an event name and object.
- Hooks
    - Workhorse of past Drupal versions
    - Great for altering existing data
    - Fast
    - Previously only best option
    - Hooks Recipe – Invoke
        - Pick a unique name for your hook
        - Use \Drupal\Core\Extension\ModuleHandlerInterface methods
    - Hooks or Events?
        - Example: Video Encoder Module
            - New plugin type so you can select an encoder
            - Provide a service to encode a given video file
            - User a service to interact with individual encoder API
            - Dispatch events so other modules can react
            - Invoke an alter hook to allow others to modify the configuration of an encoding profile
- Slides: https://speakerdeck.com/eojthebrave/altering-extending-and-enhancing-drupal-8
- Recording : https://www.youtube.com/watch?v=tMM-I70ksQA

Aha! Understanding and Using Render Arrays in Drupal 8 – Gus Childs (Chromatics)

- But why render arrays?
    - Problem: build feature X like the last site but slightly different
    - Solution: copy/paste?
        - Don’t Repeat Yourself
            - Solution: abstraction
    - Render arrays are abstract
    - Example:

```
$logo = array(
  '#theme' => 'image',
  '#path' => 'logo.png',
  '#alt' => t('My logo!'),
        );
```

- Printing render array in D8
  - {{ logo }}

- A responsive image render array in D8.

```
$logo_uri = 'public://logo.png';
$logo = array(
  '#theme' => 'image',
  '#uri' => $logo_uri,
  '#alt' => t('My logo!'),
  '#sizes' => '100vw',
  '#srcset' => array(
    array(
      'uri' => ImageStyle::load('thumbnail')->buildUri($logo_uri),
      'width' => '250w',
    ),
    array(
      'uri' => ImageStyle::load('large')->buildUri($logo_uri),
      'width' => '750w',
    ),
), );
```

- **Twig Attributes** : <img{{artributes}}>
- The resulting responsive image markup.
  <img src="logo.png" alt="My logo!" sizes="100vw" srcset="styles/thumbnail/public/logo.png 250w, styles/large/public/logo.png 750w">
- **Theme System**
  - hook_theme()
  - hook_preprecess_HOOK()
  - hook_theme_suggestions_HOOK()
  - Twig templating engine
- **Render API**
  - Render arrays
  - Render elements
  - Caching Attachments
  - Placeholders
- The core structure of the Render API is the render array, which is a hierarchical associative array containing data to be rendered and properties describing how the data should be rendered
- Modules have the capability of defining elements which are essentially prepackaged default render arrays
- The Form API is a subset of the Render API
- **Cacheability of render arrays** https://www.drupal.org/developing/api/8/render/arrays/cacheability
- The future
  - Refactor Render API to be Object Oriented https://www.drupal.org/node/1843798
  - Unify & simplify render & theme system
    - https://www.drupal.org/node/2702061
  - Render arrays, the current theme system and the current Render API may be removed from Drupal 9

- Slides: https://events.drupal.org/sites/default/files/slides/render-arrays-nola-guschilds.pdf
- Recording: https://www.youtube.com/watch?v=_bJsCD-tIdM

The Infinite Wonder of the Symfony Event Dispatcher – Adam Englander

- Symfony is initially difficult to understand. Once understood possibilities are endless.
- Drupal 8 with Symfony
  - Event subscribers for hooks
  - Dependency injection for code services
  - Portable code in open ecosystem
  - Fully unit testable
  - Upgrade at will
- **Highly modular code**
  - Build you modules and template like building blocks
  - Reusable code in an open ecosystem
  - Use proven libraries inside and outside the Drupal ecosystem
- **Highly Decoupled Code**
  - Injects services in the Dependency Injection Container
  - Should not require intimate knowledge of code systems , only event name and class
- **Highly Extension Code**
  - Modular code is extensible
  - Logic can be extracted into services
  - Logic can be extracted outside the Drupal environment
- How to use the event dispatcher
  - Create a service in your module that implements the EventSubscriberInterface
  - Write handlers for those events in your service
  - Return a list of handled events with the handlers and their priority with the service's getSubsribedEvents method
  - Register the service with the tag event_subscriber in your services.yml file
- Example walkthrough https://github.com/aenglander/drupal-event-dispatcher-example
- **Build a Better Drupal**
  - Create pre and post events for everything
  - Get away from copy core and modify
  - Utilize more non-Drupal libraries for more functionality
  - Accelerate plaftform growth
- Recording: https://www.youtube.com/watch?v=tEMom4kP39w

Watch the Hacker Hack – Michael Hess (University of Michigan), Ben Jeavons (card.com), Greg Knaddison (card.com)

- The only way to ensure that you are 100% secure is to unplug
- Security breaches happen even to those with huge budgets for security
- 70% of data breaches occurred in businesses with fewer than 100 employees
- Attack surface area is more than just Drupal
  - Apache/Nignx
  - MySQL/Postgres
  - Linux
  - Memcache/Redis
  - Solr
  - ImageMagick
- Types of Hackers
  - People who hack for good
  - Script Kiddies
  - Colored Hat
  - State sponsored hacker
  - Personal gain
  - Other
- **How to secure Drupal site**
  - Must use Drupal 2 factor authentication
    - https://www.drupal.org/project/tfa
    - https://github.com/d8-contrib-modules/tfa
  - Disable Devel module and PHP filters on production
  - Disable access to readme.txt or change.log file on the server
  - Disable account creation
  - Don't store passwords in any other files by mistake
  - Use drush sql-sanitize
  - drush paranoia-sql-sanitize-whitelist –v https://groups.drupal.org/node/508612
  - Follow Drupal Security Team
  - Keep all stack updated
  - Check your backups – must be able to recreate sites from backups
  - Beware of checkboxes in Drupal's permission page
  - Audit your configuration
  - Never give full html filter access to anyone except side admins
  - Password Policy https://www.drupal.org/project/password_policy
  - Security Review https://www.drupal.org/project/security_review
  - Paranoia module https://www.drupal.org/project/paranoia
  - Account Sentinel https://www.drupal.org/project/account_sentinel
  - Let's Encrypt is a free, automated, and open certificate authority brought to you by the non-profit Internet Security Research Group https://letsencrypt.org/
  - Don't use combination of SSL and non SSL pages
  - Use Fail2Ban
- CVE_2016_2315 – Git Repository Vulnurability
- Recording: https://www.youtube.com/watch?v=ek8FswgBvW4

The potential in Drupal 8.x and how to realize it – Angela Byron, Gabor Hojtsy (Acquia)

- **Semantic versioning** http://semver.org/
  - Predictable
  - Improvements every 6 months
  - Incentive to contribute
  - Backwards compatible
- **Top-down goals from committers**
  - Migrate UI
  - Frontend testing
  - Blocks and layouts
  - Configuration Management
  - Media
  - Workflow
- **Bottom-up goals from community**
  - BigPipe
  - Twig
  - Admin style guide
  - Contact for basic web forms
  - Picture support
  - Improved entities
- **Drupal development roadmap** https://www.drupal.org/core/roadmap
- **Ideas for improvement**
  - Iterate quickly and cheaply on ideas
  - Clear sign-off points to avoid wasting time
  - Involve the right stakeholder at the right time
  - Gain visibility for proposals from committers
  - Reduce barriers to entry into core for new ideas
  - Clear visibility of priorities for the community
- Lean Startup by Eric Ries – Everyone at Acquia leadership must read this one
- Plan template - https://www.drupal.org/core/initiative-proposal-template
- **Governance** - https://www.drupal.org/governance
- Who to talk?
  - Initiative coordinators
  - Subsystem maintainers
  - Topic maintainers
  - Committers
- Drupal search sucks!
- Summary
  - Get sign-off/rejection before doing tons of work
  - Validate direction with real world data vs. bikeshedding
  - Make it cheaper/easier/faster to improve core all around
  - Jump through the right hoops at the right time
  - ….
  - Profiit!

- Slides - https://events.drupal.org/sites/default/files/slides/The%20potential%20in%20Drupal%208.x%20and%20how%20to%20realize%20it.pdf
- Recording - https://www.youtube.com/watch?v=ddZ-McBq-Hg

Keynote – Everybody Hurts: Content for Kindness by Sara Wachter-Boettcher

- Everyone's history is personal, complicated, reactions are predictable.
- Make things for whole humans
- Make every decision an act of kindness
- Rethink "normal"
- Imagine your users
- Don't force false categories
- Don't assume feelings
- Don't forget who's in charge – users
- What assumptions are you putting into your work?
- Smooth, seamless, easy for whom?
- Kindness is letting users define themselves.
- Act of compassion take courage and practice.
- Kindness is adjusting to our users' needs not asking them to fit ours
- Set aside ego
- **Kindness is swallowing our pride if it will help our users**
- Every field in a form carries weight – everything is a trigger for someone
- Kindness is only asking for what we need
- Fun facts create real risks
- Ask – do I control the context of my content?
- Kindness is designing for worst
- Recording - https://www.youtube.com/watch?v=UcjT63hl4l4

Drupal8 Kickstart – An Overview for Developers by Pater Sawczynec (Pantheon)

- **D8 readings**
  - Drupal 8 API Reference - https://api.drupal.org/api/drupal/8.2.x
  - Programmer's Guide to Drupal - http://shop.oreilly.com/product/0636920034612.do
  - Most D8 online documentation older than December 2015 is very unlikely to be fully accurate
- **D8 Ecosystem**
  - Symfony
  - PHP OOP
  - PHP 7
  - Drush
  - Git
  - Markdown
  - Composer

    - Linux shess
    - SSH
    - PHPUnit
    - Behat
    - jMeter
    - MySQL
    - MySQL Workbench
    - JSON
    - jQuery
    - AJAX
    - Regex
    - Twig
    - Memcache
    - Varnish
    - Jenkins
    - Apache
    - PHPStorm
- D8 Overview
    - Leverage existing industry-standard technologies so that D8 can interface with and be programmed like other globally recognized PHP frameworks using PHP OOP concepts.
- **D8 will upgrade to Symfony 3.0 in a minor release and drop Symfony 2.x backwards compatibility**
- **D8 Essentials**
    - Composer
    - Drush
    - Drupal Console
    - Git
    - YAML
    - PHP OOPS
    - Comments / Annotations
    - Testing
- **Composer**
    - You can use Composer to install and manage D8
    - $ composer -install
    - $ composer –update
    - https://www.drupal.org/project/composer_manager
    - You can install Drush with Composer
- **Drupal Console** is an app that you can use to quickly make the scaffold of a new D8 module or a D8 service or a block plugin. Also helps with D8 debugging and admin tasks.
- **YAML** – A simple, clean format for storing structured data that is easier to read/write than XML
    - Schema files define the expected structure and allowed elements of YAML files
- **Class** – A set of methods and properties organized in a file that offer a service
    - Example – controllers, routers, forms and plugins
- **Interface** – A class with empty default methods that all other classes based on it must offer
- **Abstract Class** – A class with default abstract method that classes based on it must offer
- **Trait** – A set of PHP functions in one file that supply a useful set of related functions
- **OOP readings**
    - http://www.phpfreaks.com/tutorial/oo-php-part-1-oop-in-full-effect
    - http://www.java2s.com/Tutorials/PHP/Language_Basics/0100__PHP_Class.htm

- **Dependency Injection** – Instantiating a class, but telling the class what you want it to use to work
  - http://fabien.potencier.org/what-is-dependency-injection.html
- **Services** – Something a class offers
- **Plugins** – e.g. Blocks
- **Annotations** – Properly formatted comments are used by D8 to create documentation, identify tests, and in some cases for D8 to make discovery of services and other plugin functionality
- **Method Chaining** – used in jQuery, PHP, Drupal
- **Short Array Syntax** – As of PHP 5.4 and I use throughout D8. It replaces array() with []
- **Type Hinting** – use it to specify the expected data type of an argument in a function declaration
- **Web profiler toolbar** is provided by Devel module
- **D8 Events** – D8 uses Symfony kernel and events.
  - KernelEvents::Controller
  - KernelEvents::Exception
  - KernelEvents::Finish_Request
- **Services in D8**
  - Core functionality in D8 such as current user info, current path, node info, is logged in, module exists… these are all called services
  - Core services are declared in /core/core.services.yml
  - Services can be accessed throughout D8 via the global Drupal namespace \Drupal
  - Example - $account = \Drupal::currentUser()
- **Caching and Cache Invalidation**
  - http://www.ninjaducks.in/hacking/caching-in-drupal-8/
  - https://www.sitepoint.com/exploring-cache-api-drupal-8/
  - https://www.drupal.org/developing/api/8/render/arrays/cacheability
  - BigPipe
  - SmartCache
  - Dynamic Page Caching
  - Cache Keys
  - Cache Contexts
  - Cache Tags
  - Max-Age
- **Routing System** in D8 https://www.drupal.org/developing/api/8/routing
- Block plugin creation overview - http://mattkorostoff.com/article/programmatically-creating-a-block-in-drupal-8
- **Configuration Management**
  - Initiative - http://drupal8cmi.org
  - Principles
    - https://www.chapterthree.com/blog/principles-configuration-management-part-one
    - https://www.chapterthree.com/blog/principles-configuration-management-part-two
- **Configuration Inspector Module -** https://www.drupal.org/project/config_inspector
- D7 to D8 module upgrader - https://www.drupal.org/documentation/modules/drupalmoduleupgrader
- Simple Test - for testing the user interface and interaction on the front end https://www.drupal.org/simpletest
- **PHPUnit** – for testing discrete functionality like classes and backend functionality https://www.drupal.org/phpunit
- Testing
  - Behat
  - Drupal Extensions

    - Mink Extension
    - Mink
    - Goutte or Phantom.js
    - http://behat-drupal-extension.readthedocs.io/en/3.1/intro.html
- D8 Training Resources
    - https://drupalize.me
    - https://buildamodule.com
    - https://knpuniversity.com
    - www.safaribooksonline.com
    - https://www.youtube.com/user/DrupalAssociation
- Slides - https://events.drupal.org/sites/default/files/slides/Drupal-8-Kickstart-DrupalCon-NOLA-05-11-2016.pdf
- Recording - https://events.drupal.org/sites/default/files/slides/Drupal-8-Kickstart-DrupalCon-NOLA-05-11-2016.pdf

PHP 7: The New New PHP by Larry Garfield (platform.sh)

- **Return types**
    - Not nullable
    - Returning false on error is a big 'screw you' to your users
- New type specs
    - int
    - float
    - string
    - bool
- **Strict vs Weak mode** – weak by default
    - declare(string_types=1)
    - Use strict typing 90% of the time (everywhere except input)
- **T_SPACESHIP <=>**
- **??** – NULL coalesce, checks setness, not truthiness
- **Cryptographically Secure Pseudo-Random Number Generator**
    - random_bytes(16)
    - random_int(1,100)
    - password_hash() / password_verify()
    - For PHP 5 - https://github.com/paragonie/random_compat
    - If you are not using these functions for security, you are doing it wrong
- **Mocking**
    - https://www.sitepoint.com/an-introduction-to-mock-object-testing/
    - https://www.toptal.com/php/php-7-performance-features
- **Better Error Handling**
- **Better Assertions** – called Expectations
- Better Iterators
- Uniform variable syntax
    - Always left to right
    - Logical
- Slides - http://garfieldtech.com/presentations/dcnola2016-php7/#/

- Recording - https://www.youtube.com/watch?v=zGpaswrhFGQ

Offline-capable, decoupled Drupal 8 with React.js and React Native

- When the CMS is decoupled, a lot of new things become possible
- Spent three months building a fully offline capable web site
- Technologies used
    - React.js https://facebook.github.io/react
    - **PouchDB** - http://pouchdb.com/
    - RELAXed - https://www.drupal.org/project/relaxed
    - React Native - https://facebook.github.io/react-native/
- **Deploy suite**
    - Multiversion
    - Replication
    - Workspace
    - Deploy
    - RELAXed – RESTFul API that replicates easily. Using the API spec from CouchDB
- Workflow Initiative
- **CouchDB** – HTTP document database. Replicates easily.
- **PouchDB** – CouchDB's little brother. Runs in the browser.
- **Drupal Deploy** - http://www.drupaldeploy.org/
- https://github.com/relaxedws
- Recording https://www.youtube.com/watch?v=ovc8eHq6LXY

Keynote: Your brain health is more important than your standing desk by Dave Sawyer (FFW)

1. **Get a coach**
    a. Follow your goals
    b. Keep track of them
    c. We overestimate per day
    d. We underestimate per year
2. **Reflect**
    a. Once a week- am I still happy?
3. **Find your productive time**
    a. 2 hours time in the morning
4. **Split up your day**
    a. Dislike – 30 min
    b. Like – 60 min
    c. Dislike – 30 min
5. **Drink water**
    a. Brain – 85% water
    b. 1% dehydration results in 5% decreased cognitive functioning
    c. 2% dehydration results in fuzzy and shorter memory, problem focusing
6. **Take breaks**

    a. Focus
    b. Diffuse

7. **Find something relaxes you**
8. **Sleep**
    a. Day light saving – 17% increase in car accidents
    b. Only sleep when tired
        i. Relaxing breath
        ii. Count down from 100
9. **Keep on improving**

Workflow Initiative by Tim Millwood (Appnovation)

- Multiversion – make everything revisionable – do not delete anything – keep in archive
- Replication – what content have changed
- Workspace – like a git branches
- RELAXed Web Service – when you have separate site and need replication
- Deploy
- Revisions enabled by default
    - Make all content entities revisionable
- Phase 2
    - Revision UUID and/or hash
    - Parent revision
    - Deleted flag
- Phase 3
    - Purging revisions
- Phase 4
    - Trash
- Phase 5
    - Moderation API
    - Workspace API
- Phase 6
    - UUID Index
    - Revision UUID/hash index
    - Sequence index
- Phase 7
    - Replication services – changes, revision diff
    - 3 way merge – like git
    - Replication API
    - Conflict management API
    - Experimental UI
- Phase 8
    - Conflict management UI
    - Workspace UI
    - Revision moderation module

- Phase X – Cross site content staging
- Phase XI – Autosave – CouchDB
- Recording - https://www.youtube.com/watch?v=_rAB8DJnc8Y

Demystifying AJAX Callback Commands in Drupal 8 by Mike Miles (Genuine)

- Goal – explain Ajax callback outline
- Callback Command : JavaScript
    - Attached to Drupal.AjaxCommands.prototype, defined in misc/ajax.js
    - Accepts 3 arguments
        - Ajax
        - Response
        - Status
    - Wrapper for additional JavaScript
- Callback Command: PHP
    - Class that implements CommandInterface
    - Defines a method called 'render'
    - Returns an associative array
        - Must have element with key of 'command'
        - Value must be name of JavaScript function
- Callback Commands
    - Used in all Ajax requests
    - Composed of two parts – JavaScript function, PHP Class
    - Provided by core and modules
- To Create a Callback Command
    - Create a module
    - Attach JavaScript function to 'Drupal.AjaxCommands.prototype'
    - Define an asset library
    - Create PHP class that implements 'CommandInterface'
- To Use Callback Commands
    - Include the Ajax library and commands on the page
    - Have endpoint that returns an AjaxResponse
    - Add commands to response using 'addCommand'
- To Use Ajax Callback Command
    - Include Ajax framework and commands onto page
    - Return an AjaxResponse object
    - Attach commands with 'addCommand'
- Recording - https://www.youtube.com/watch?v=7XND83bsoG8
- Slides - http://www.slideshare.net/MichaelMiles2/demystifying-ajax-callback-commands-in-drupal-8-61931669
- Example Code - https://github.com/mikemiles86/reveal.js/tree/drupalcon-d8ajax/examples/drupal8