DRUPALCON BALTIMORE

PLUGINS, COMPOSER AND PHP7 OH MY!

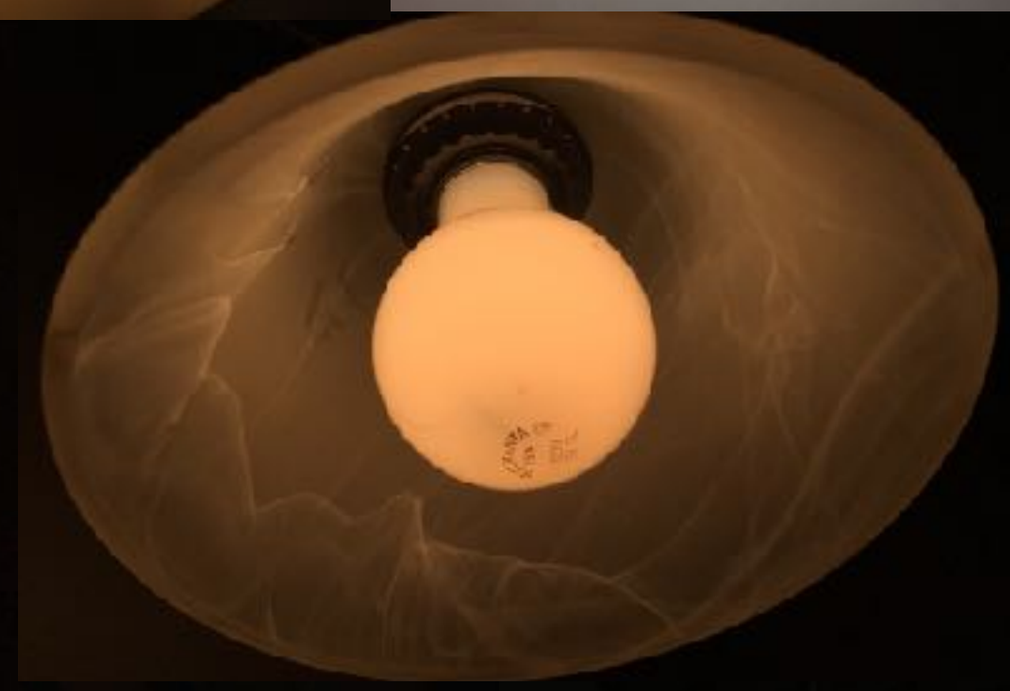
CTOOLS CO-MAINTAINER
DRUPAL 8 CONTRIBUTOR
PAGE LAYOUT SPECIALIST
PLUGIN SUBSYSTEM CO-AUTHOR
ACQUIA TECHNICAL CONSULTANT

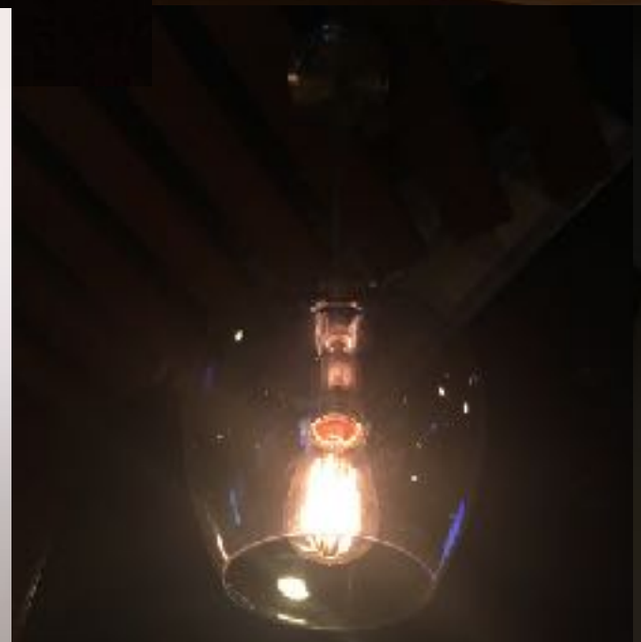
KRIS VANDERWATER










Sunlight



Joomla!™

WordPress

CodeIgniter

yiiframework

Symfony

Laravel

ZEND FRAMEWORK

CakePHP

?

History

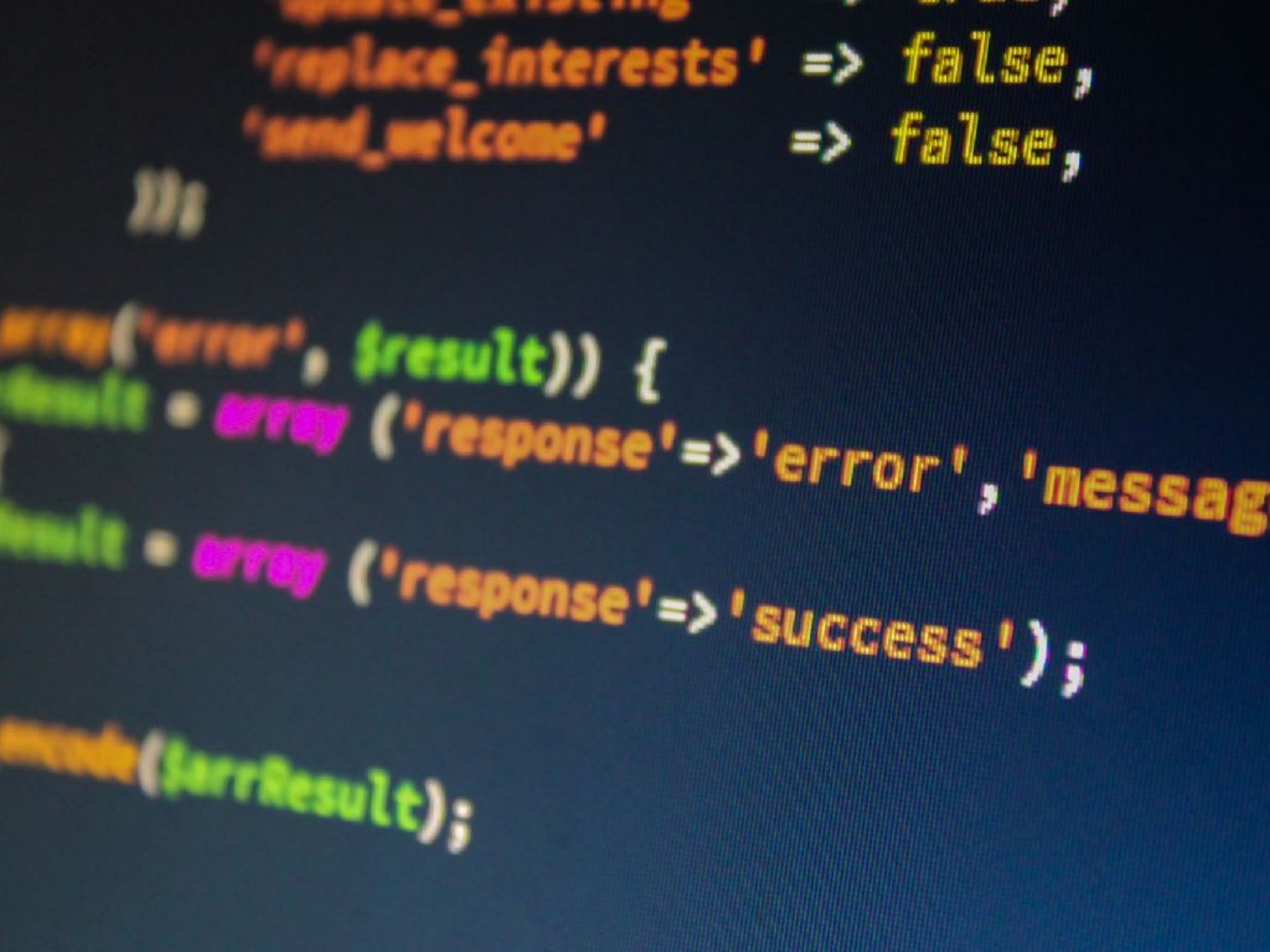

COMPOSER

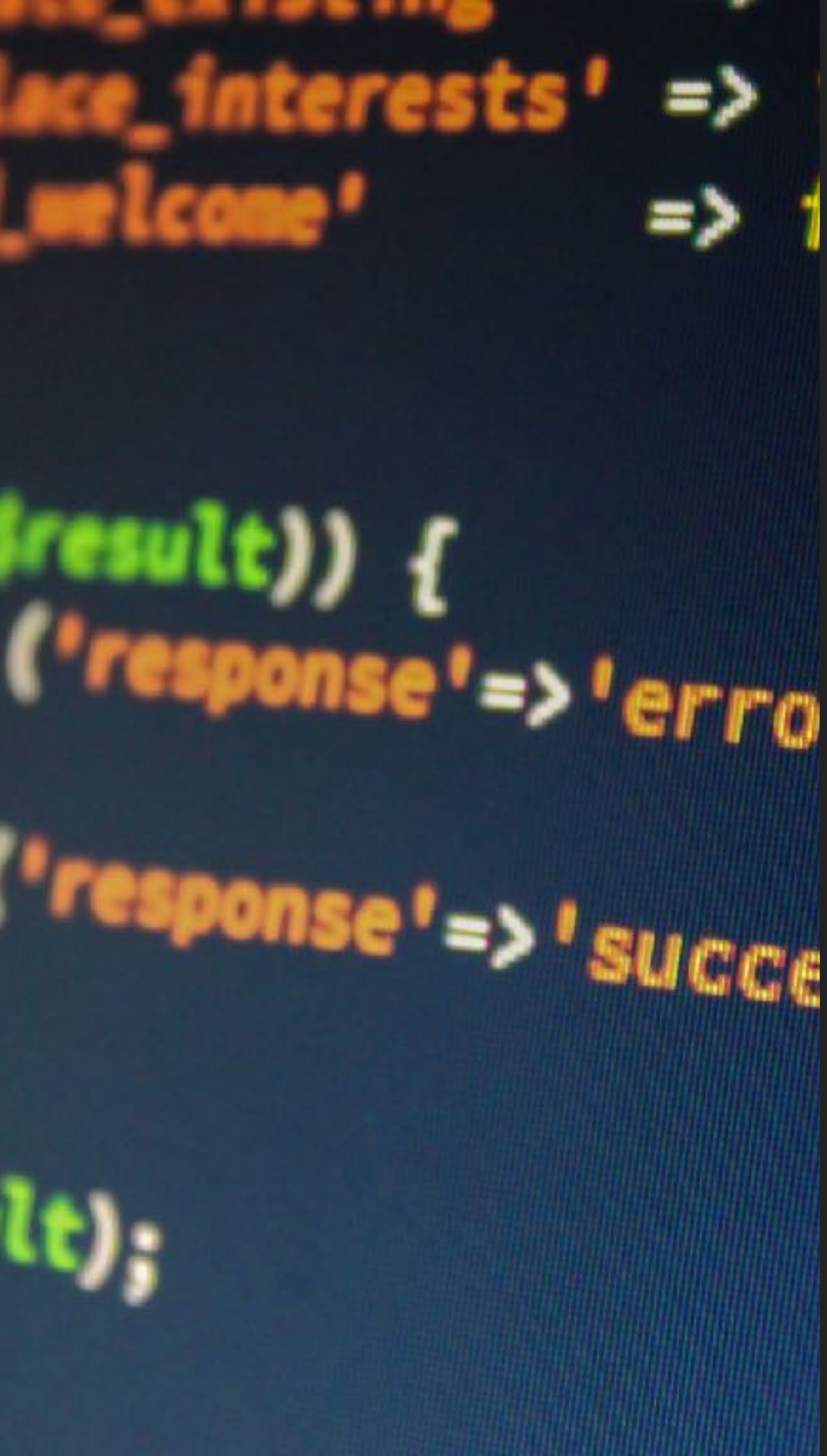
https://www.drupal.org/node/1826054

Expose Drupal Components outside of Drupal

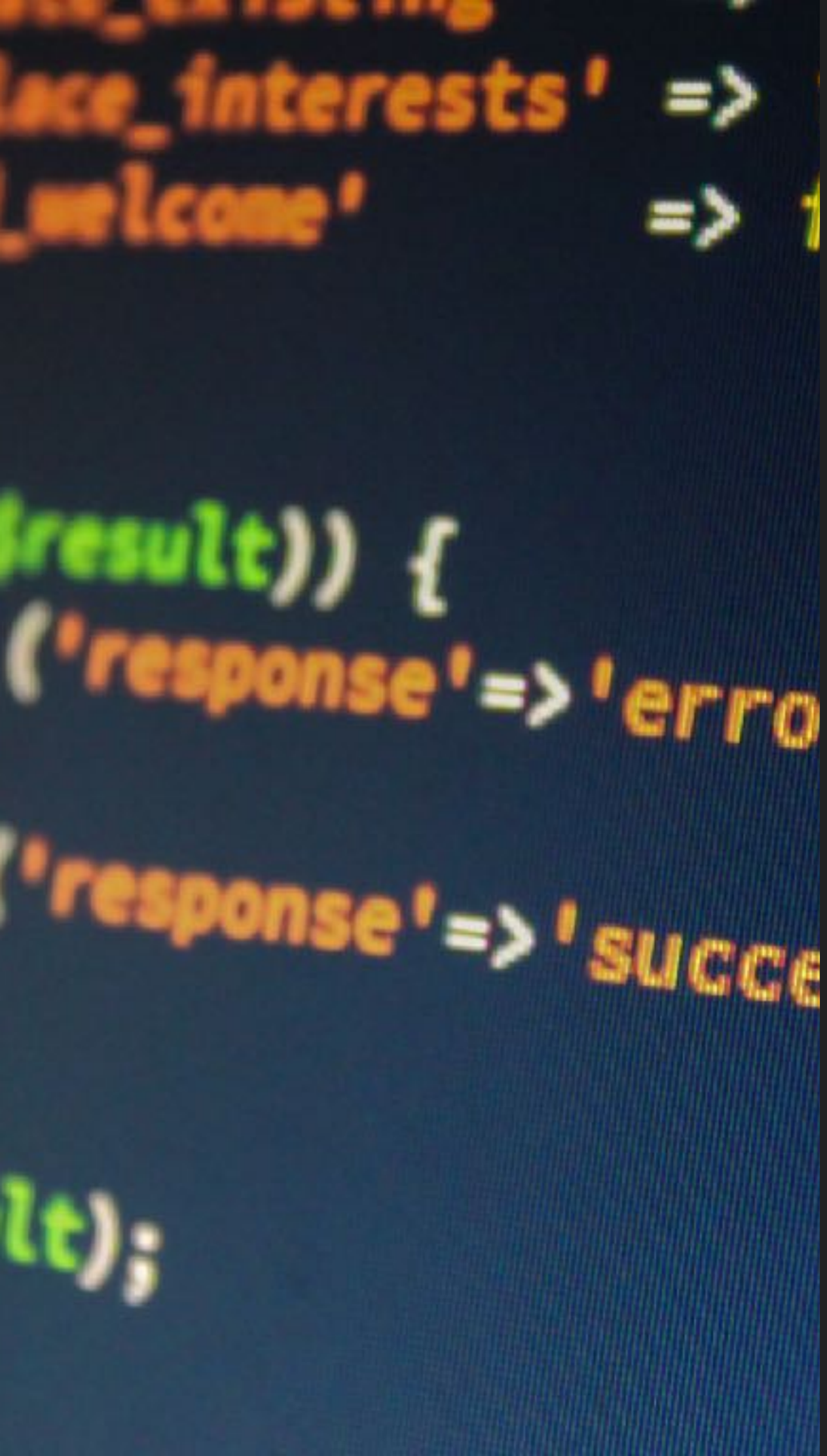
https://www.drupal.org/node/1497366

Committed by Dries:
July 14th 2012 (Nearly 5 years ago)


'replace_interests' => false,
'send_welcome' => false,
('error', $result)) {
array ('response'=>'error', 'messag
array ('response'=>'success');
($arrResult);

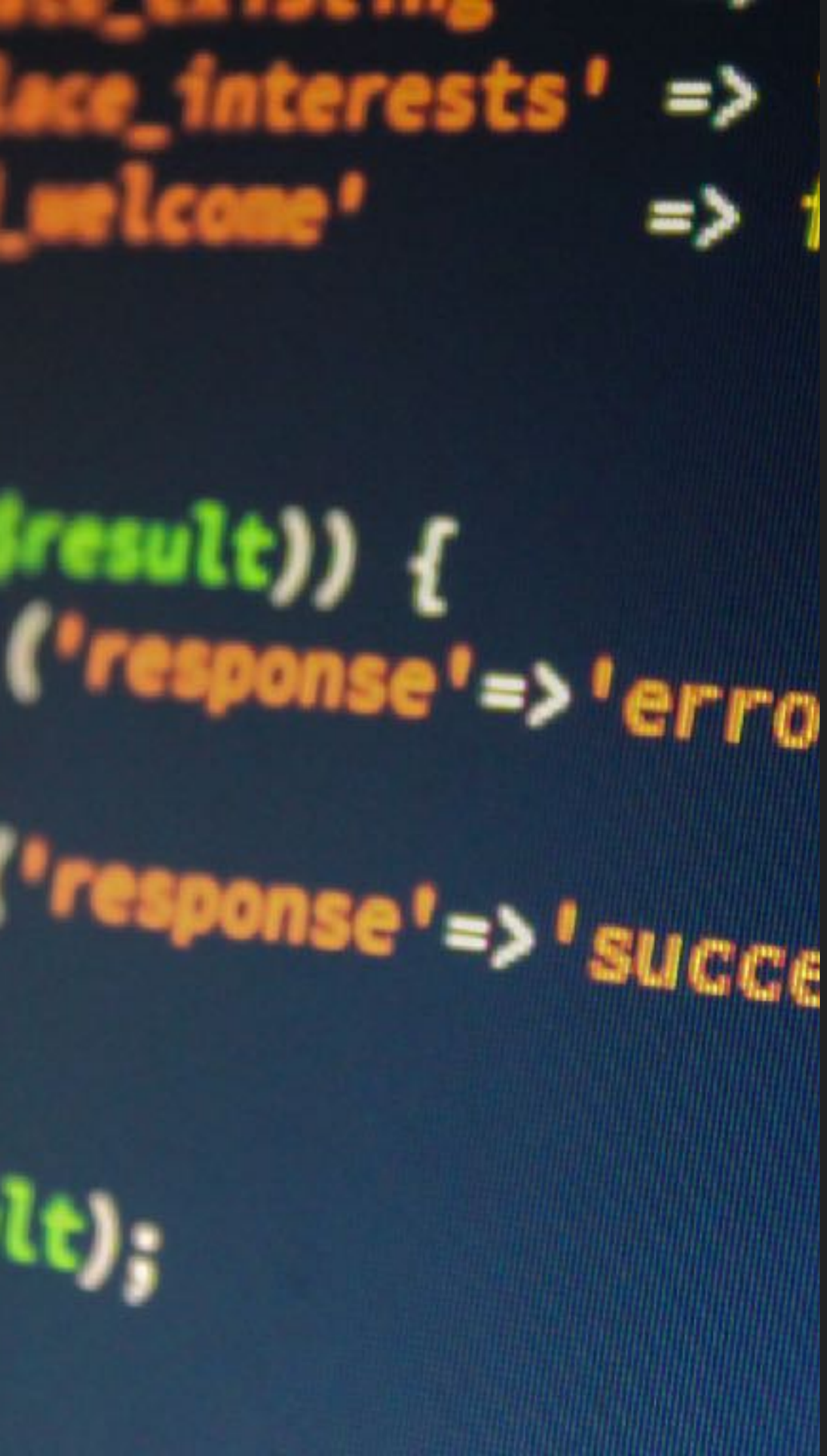


PRIORITIES



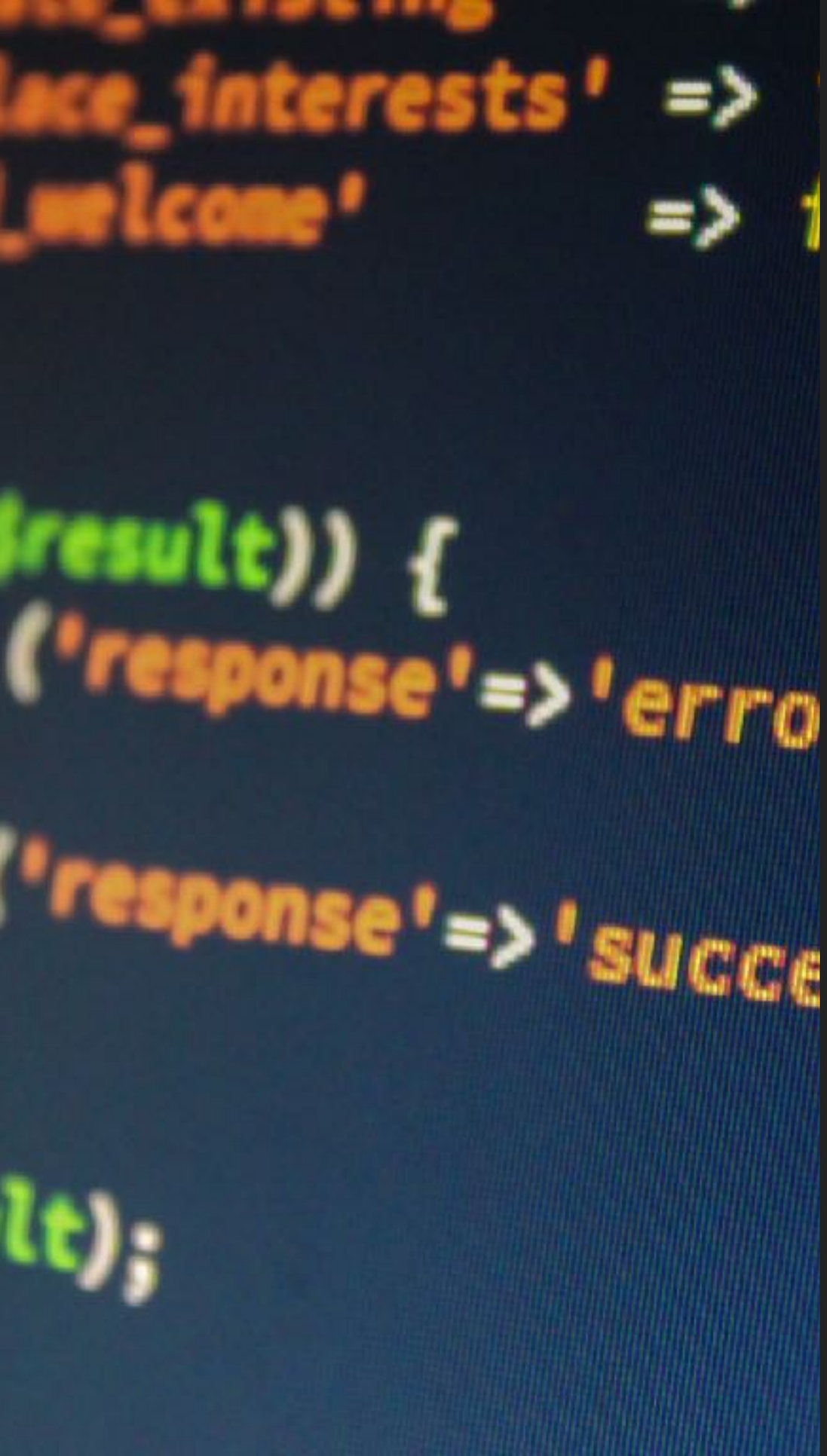
1. Static Factory Methods

PRIORITIES



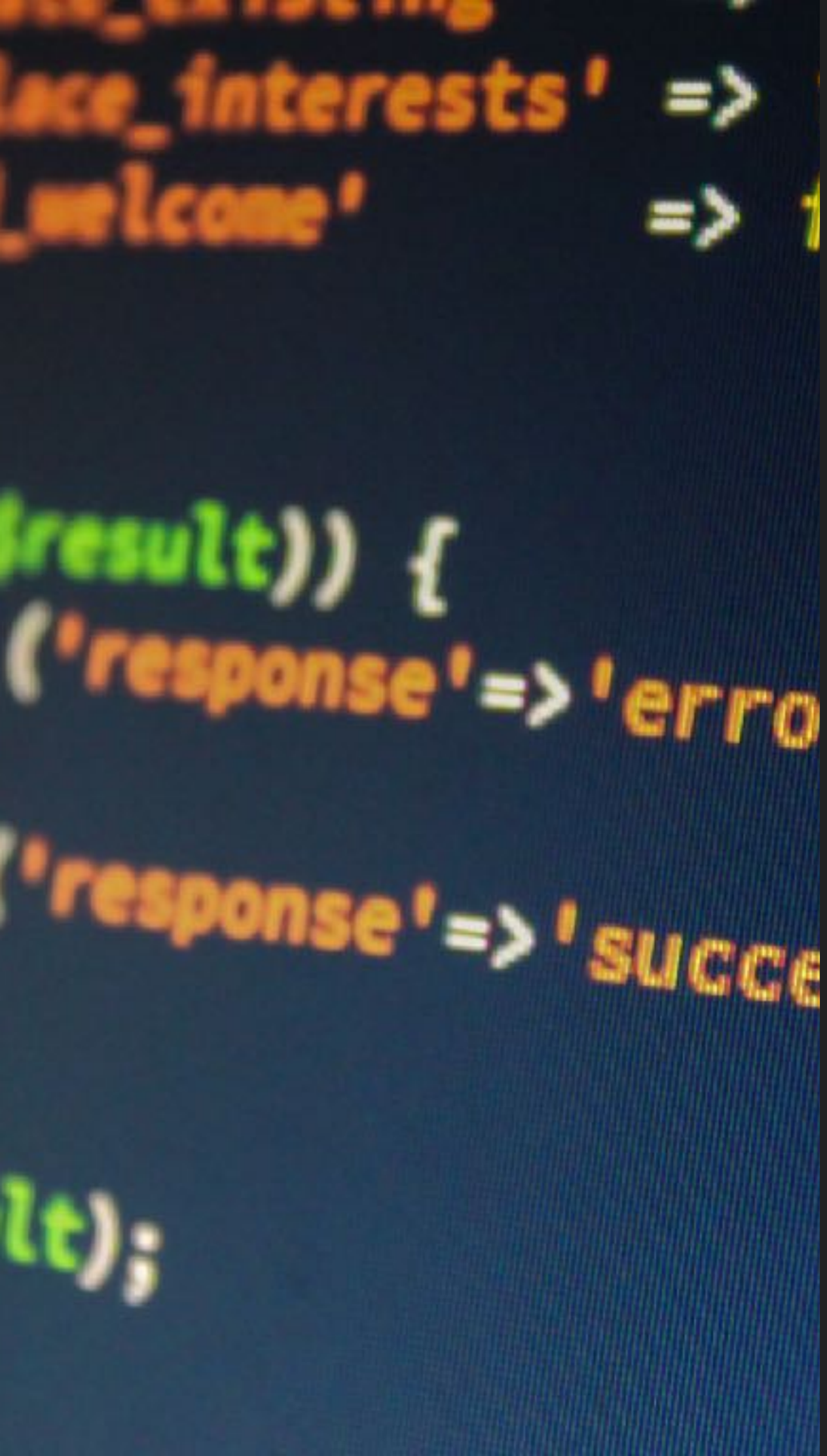
1. Static Factory Methods
2. Constructors

PRIORITIES



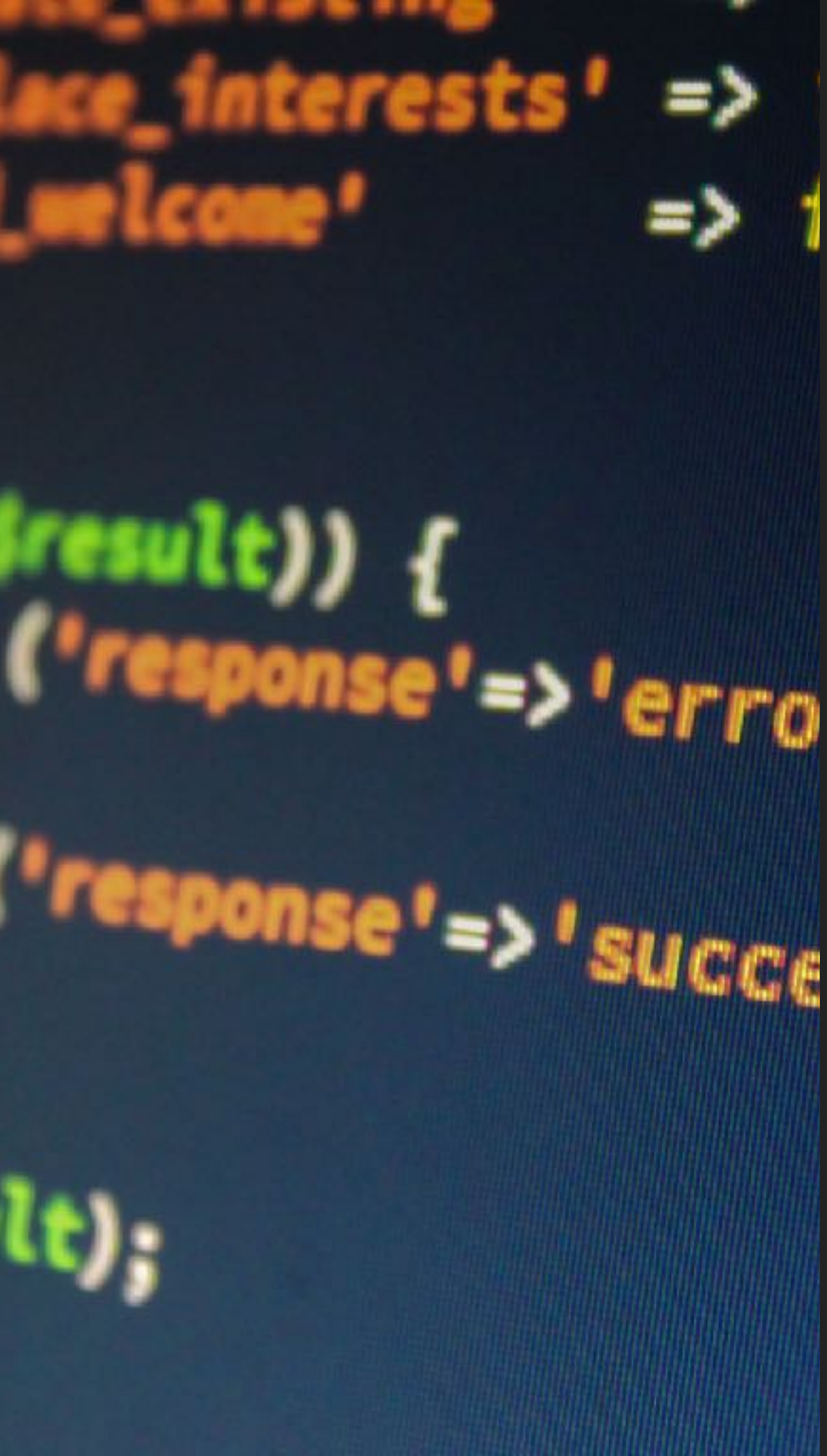
1. Static Factory Methods
2. Constructors
3. Array based Plugin Definitions

PRIORITIES



1. Static Factory Methods
2. Constructors
3. Array based Plugin Definitions
4. Plugin Filtering

PRIORITIES



1. Static Factory Methods
2. Constructors
3. Array based Plugin Definitions
4. Plugin Filtering
5. Plugin Mutating

PRIORITIES


php7

php7

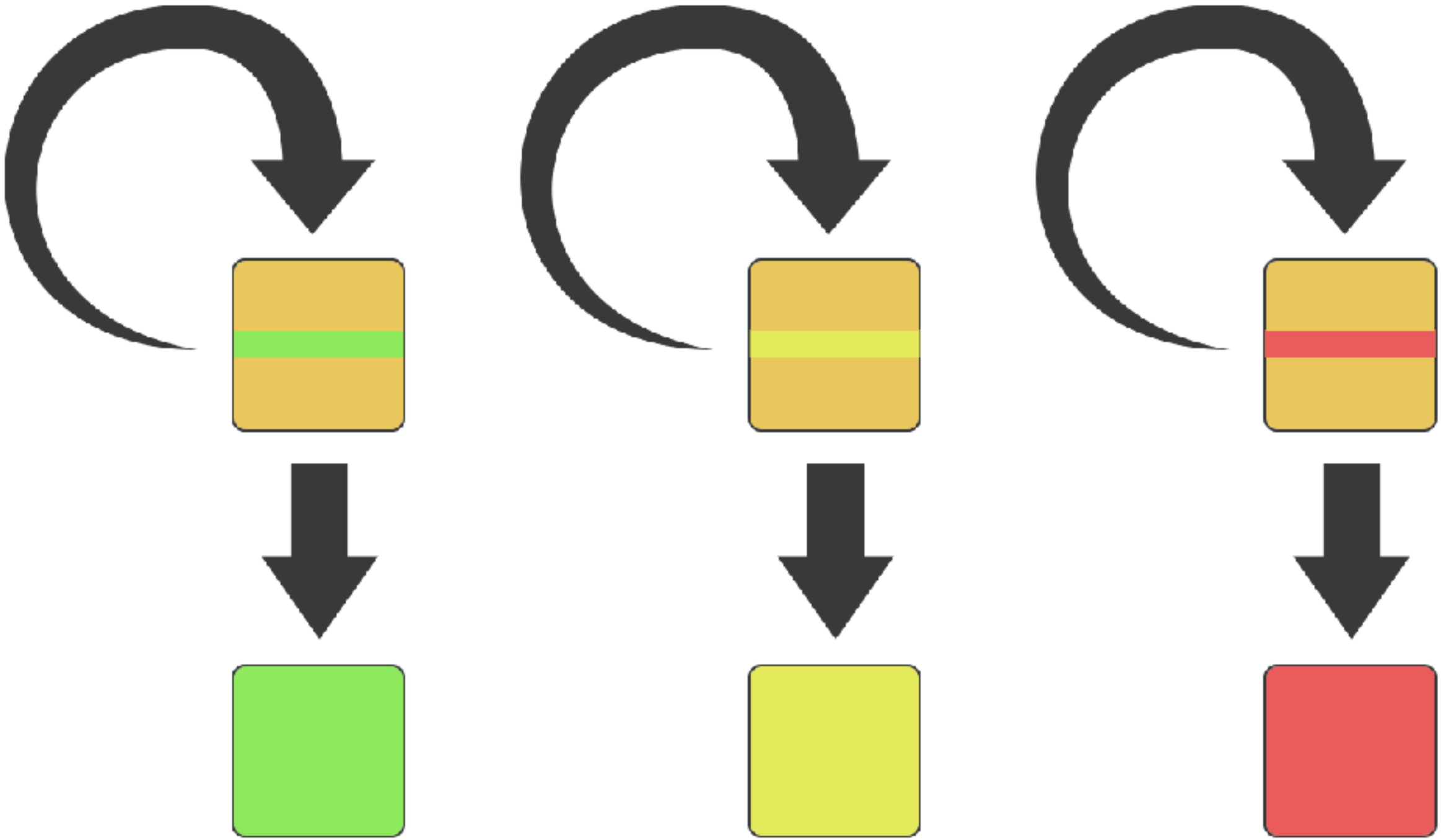
and 5.6

STATIC FACTORY

Plugin 1

Plugin 2

Plugin 3



Factory 1

Factory 2

Factory 3

Class

Plugin 1

Plugin 2

Plugin 3

PLUGIN::__CONSTRUCT()

PLUGIN::__CONSTRUCT()

- D8

PLUGIN::__CONSTRUCT()

- D8
  - array $configuration

PLUGIN::__CONSTRUCT()

- D8
  - array $configuration
  - $plugin_id

PLUGIN::__CONSTRUCT()

- D8
  - array $configuration
  - $plugin_id
  - $plugin_definition

PLUGIN::__CONSTRUCT()

- D8
  - array $configuration
    - Dumping ground for all the things...
  - $plugin_id
  - $plugin_definition

PLUGIN::__CONSTRUCT()

- D8
  - array $configuration
    - Dumping ground for all the things...
    - Kinda sucks...
- $plugin_id
- $plugin_definition

PLUGIN::__CONSTRUCT()

- PHP Component

PLUGIN::__CONSTRUCT()

- PHP Component
  - No prescribed constructor or parameters.

PLUGIN::__CONSTRUCT()

- PHP Component
  - No proscribed constructor or parameters.
- Custom Constructors

PLUGIN::__CONSTRUCT()

- PHP Component
  - No proscribed constructor or parameters.
  - Custom Constructors
  - Custom Factories

PHP 5.6

PHP 5.6

- Variadics

PHP 5.6

- Variadics
  - Replacement for func_get_args()

PHP 5.6

- Variadics
  - Replacement for func_get_args()
  - WAAAAAAY better

PHP 5.6

- Variadics
  - Replacement for func_get_args()
  - WAAAAAAY better
- Now with type-hints!

PHP 5.6

- Argument Unpacking

PHP 5.6

- Argument Unpacking
  - Essentially the opposite of variadics

PHP 5.6

- Argument Unpacking
  - Essentially the opposite of variadics
  - Like exploding an array into a function

PHP 5.6

- Argument Unpacking
  - Essentially the opposite of variadics
  - Like exploding an array into a function
- call_user_func_array()

PHP 5.6 VARIADICS & ARGUMENT UNPACKING EXAMPLE

```php
public function createInstance($definition, ...$constructors) {
  $class = $definition->getClass();
  return new $class($definition, ...$constructors);
}
```

PHP 5.6 VARIADICS & ARGUMENT UNPACKING EXAMPLE

```php
public function createInstance($definition, ...$constructors) {
  $class = $definition->getClass();
  return new $class($definition, ...$constructors);
}
```

```php
public function createInstance(FooInterface …$constructors) {…
```

PHP 5.6 VARIADICS & ARGUMENT UNPACKING EXAMPLE

```php
public function createInstance($definition, ...$constructors) {
  $class = $definition->getClass();
  return new $class($definition, ...$constructors);
}
```

```php
public function createInstance(FooInterface …$constructors) {…
```

```php
public function createInstance(…$constructors) : PluginInterface {
```

PLUGIN DEFINITIONS

PLUGIN DEFINITIONS

- D8

PLUGIN DEFINITIONS

- D8
  - Kris' inexperience showing

PLUGIN DEFINITIONS

- D8
  - Kris' inexperience showing
  - Objects technically supported but painful to implement

PLUGIN DEFINITIONS

- D8
  - Kris' inexperience showing
  - Objects technically supported but painful to implement
  - Thank Tim Plunkett for it even being possible

PLUGIN DEFINITIONS

- D8
  - Kris' inexperience showing
  - Objects technically supported but painful to implement
    - Thank Tim Plunkett for it even being possible
  - Frustrating since you are generally writing objects but end up with arrays… :-p

PLUGIN DEFINITIONS

- PHP Component

PLUGIN DEFINITIONS

- PHP Component
  - Write it proper

PLUGIN DEFINITIONS

- PHP Component
  - Write it proper
  - Objects by default

PLUGIN DEFINITIONS

- PHP Component
  - Write it proper
  - Objects by default
  - Proper PHP Interfaces make array style easy to get

PLUGIN DEFINITIONS

- PHP Component
  - Write it proper
  - Objects by default
  - Proper PHP Interfaces make array style easy to get
  - Objects are nicer to use than arrays

PLUGIN DEFINITIONS

- PHP Component
  - Write it proper
  - Objects by default
  - Proper PHP Interfaces make array style easy to get
  - Objects are nicer to use than arrays
    - Methods for your definitions

PLUGIN FILTERS

PLUGIN FILTERS

- D8

PLUGIN FILTERS

- D8
  - Plugins by available context

PLUGIN FILTERS

- D8
  - Plugins by available context
  - Only core could have done this easily

PLUGIN FILTERS

- D8
  - Plugins by available context
    - Only core could have done this easily
- Custom Blocks

PLUGIN FILTERS

- PHP Component

PLUGIN FILTERS

- PHP Component
  - Custom Filter classes

PLUGIN FILTERS

- PHP Component
  - Custom Filter classes
  - Multiple Filter passes

PLUGIN FILTERS

- PHP Component
  - Custom Filter classes
  - Multiple Filter passes
  - Blocks by Context & reusable custom blocks

PLUGIN FILTERS

- PHP Component
  - Custom Filter classes
- Multiple Filter passes
  - Blocks by Context & reusable custom blocks
- Filtered on demand

PLUGIN FILTERS

```
$filters[] = new PluginDefinitionFilter1();
$filters[] = new PluginDefinitionFilter2();
$filters[] = new PluginDefinitionFilter3();
$definitions = $discovery->getFilteredDefinitions(…$filters);
```

PLUGIN FILTERS

```php
$filters[] = new PluginDefinitionFilter1();
$filters[] = new PluginDefinitionFilter2();
$filters[] = new PluginDefinitionFilter3();
$definitions = $discovery->getFilteredDefinitions(…$filters);
```

```php
$filter1 = new PluginDefinitionFilter1();
$filter2 = new PluginDefinitionFilter2();
$filter3 = new PluginDefinitionFilter3();
$definitions = $discovery->getFilteredDefinitions($filter1,
$filter2, $filter3);
```

PLUGIN MUTATORS

PLUGIN MUTATORS

- D8

PLUGIN MUTATORS

- D8
  - Derivers

PLUGIN MUTATORS

- D8
  - Derivers
  - Alter hooks

PLUGIN MUTATORS

- D8
  - Derivers
  - Alter hooks
- Initially implemented via decorator pattern

PLUGIN MUTATORS

- D8
  - Derivers
  - Alter hooks
  - Initially implemented via decorator pattern
  - Decorator replaced with hard-coded expectations

PLUGIN MUTATORS

- D8
  - Derivers
  - Alter hooks
  - Initially implemented via decorator pattern
  - Decorator replaced with hard-coded expectations
    - Order of operations problem

PLUGIN MUTATORS

- PHP Component

PLUGIN MUTATORS

- PHP Component
  - Mutates the full set of available plugin definitions

PLUGIN MUTATORS

- PHP Component
  - Mutates the full set of available plugin definitions
  - Can easily introspect definitions

PLUGIN MUTATORS

- PHP Component
  - Mutates the full set of available plugin definitions
  - Can easily introspect definitions
  - Derivatives already implemented

PLUGIN MUTATORS

```php
public function mutate(PluginDefinitionInterface …$definitions) : array {
  $results = [];
  foreach ($definitions as $definition) {
    if ($definition instanceof PluginDefinitionDerivativeInterface) {
      $deriver = $this->deriverResolver->getDeriverInstance($definition->getDeriver());
      foreach ($deriver->getDerivativeDefinitions($definition) as $pluginDefinition) {
        $results[$pluginDefinition->getPluginId()] = $pluginDefinition;
      }
      continue;
    }
    $results[$definition->getPluginId()] = $definition;
  }
  return $results;
}
```

TERMINOLOGY

TERMINOLOGY

- Dictionary

TERMINOLOGY

- Dictionary
- Type

TERMINOLOGY

- Dictionary
- Type
- Set

TERMINOLOGY

- Dictionary
- Type
- Set
- Filter

TERMINOLOGY

- Dictionary
- Type
- Set
- Filter
- Mutator

?

WHY THIS SESSION?

- Component reaching degree of maturity

TEXT

WHY THIS SESSION?

- Component reaching degree of maturity
- Drupal Opportunity

COMPONENT MATURITY

COMPONENT MATURITY

- Hedron

COMPONENT MATURITY

- Hedron
- Comunicata

COMPONENT MATURITY

- Hedron
- Comunicata
- PHPJmeter

COMPONENT MATURITY

- Plugin Component

COMPONENT MATURITY

- Plugin Component
  - Usable

COMPONENT MATURITY

- Plugin Component
  - Usable
  - Drupal outside of Drupal

OPINION


DRUPALCON
AMSTERDAM
2014

CONCLUSION

CONCLUSION

- Drupal is awesome

CONCLUSION

- Drupal is awesome
- PHP7 is awesome

CONCLUSION

- Drupal is awesome
- PHP7 is awesome
- Using plugins outside of Drupal is awesome

CONCLUSION

- Drupal is awesome
- PHP7 is awesome
- Using plugins outside of Drupal is awesome
- Many other Drupal apis would be awesome to have too

CONCLUSION

- Drupal is awesome
- PHP7 is awesome
- Using plugins outside of Drupal is awesome
- Many other Drupal apis would be awesome to have too
- Can we make this happen?

QUESTIONS?

Join Us for Contribution Sprints

Friday, April 28, 2017

First-Time Sprinter Workshop
9:00am-12:00pm
Room: 307-308

Mentored Core Sprint
9:00am-12:00pm
Room:301-303

General Sprints
9:00am-6:00pm
Room:309-310

#drupalsprints

WHAT DID YOU THINK?

Locate this session at the DrupalCon Baltimore website:
http://baltimore2017.drupal.org/schedule

Take the survey
https://www.surveymonkey.com/r/drupalconbaltimore

THANK YOU!