Performance

on a budget

Chris Jansen

ibuildings
WEB & MOBILE APP DEVELOPMENT

About me

- 5+ years - full stack developer
- Graduate student
  - Research on open source software
- Father

Thanks

ibuildings

WEB & MOBILE APP DEVELOPMENT

undpaul

Topics

- Definition of performance
- A method to the madness
- Performance improvements & examples

Definition

of performance

Definition of performance

Measurable performance

- Page load time
- Resource usage
- Response time

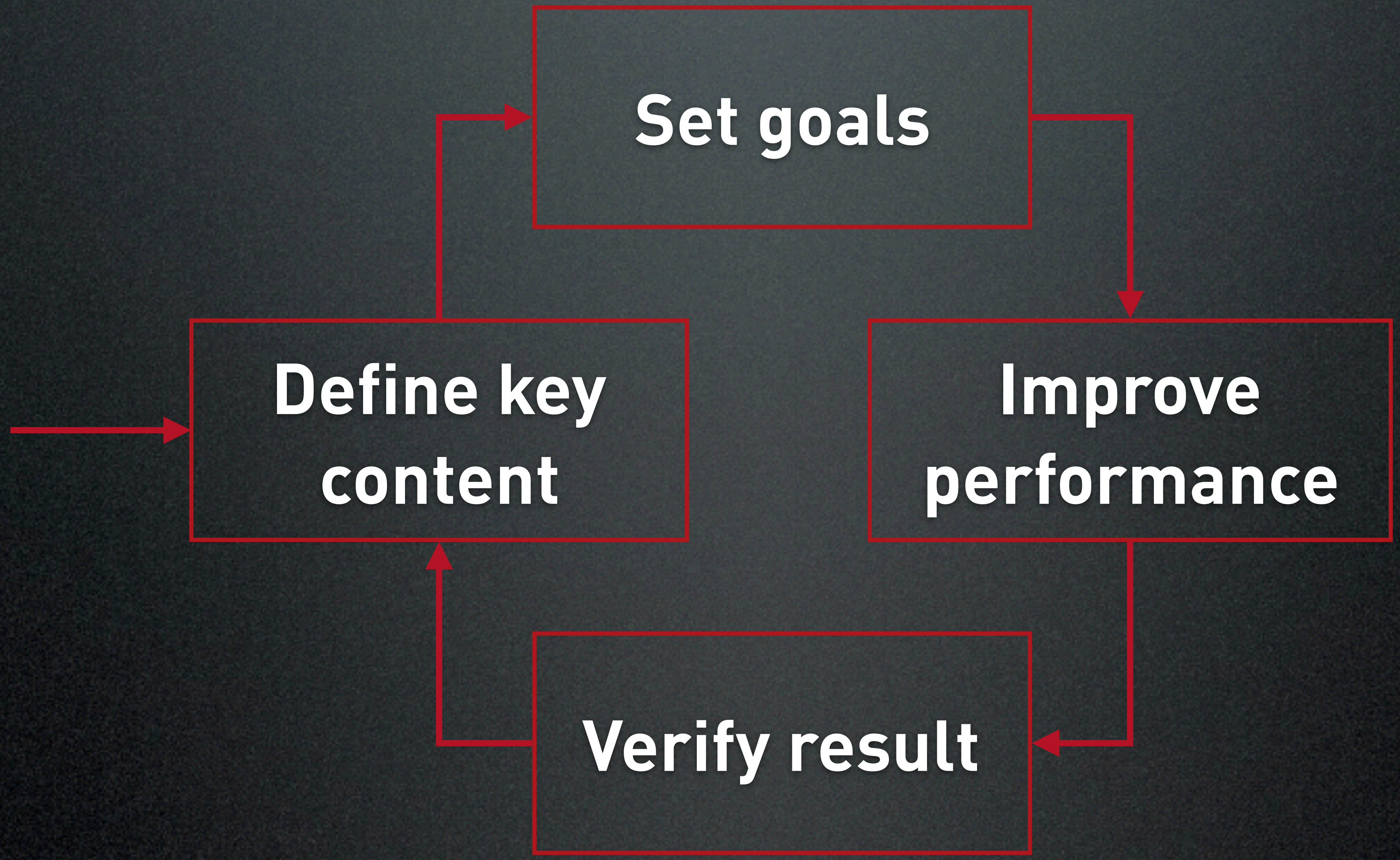
Perceived performance

- Rendering of useful content
- Activity feedback

A method

to the madness

A method to the madness



Define key content

What is your visitor looking for?

- Conversion pages
- Landing pages
- Key information
- etc.

Don’t forget your editors

- Important overviews
- etc.

Set goals

How should your key content perform?

- Be S.M.A.R.T about it.

Specific

Measurable

Acceptable

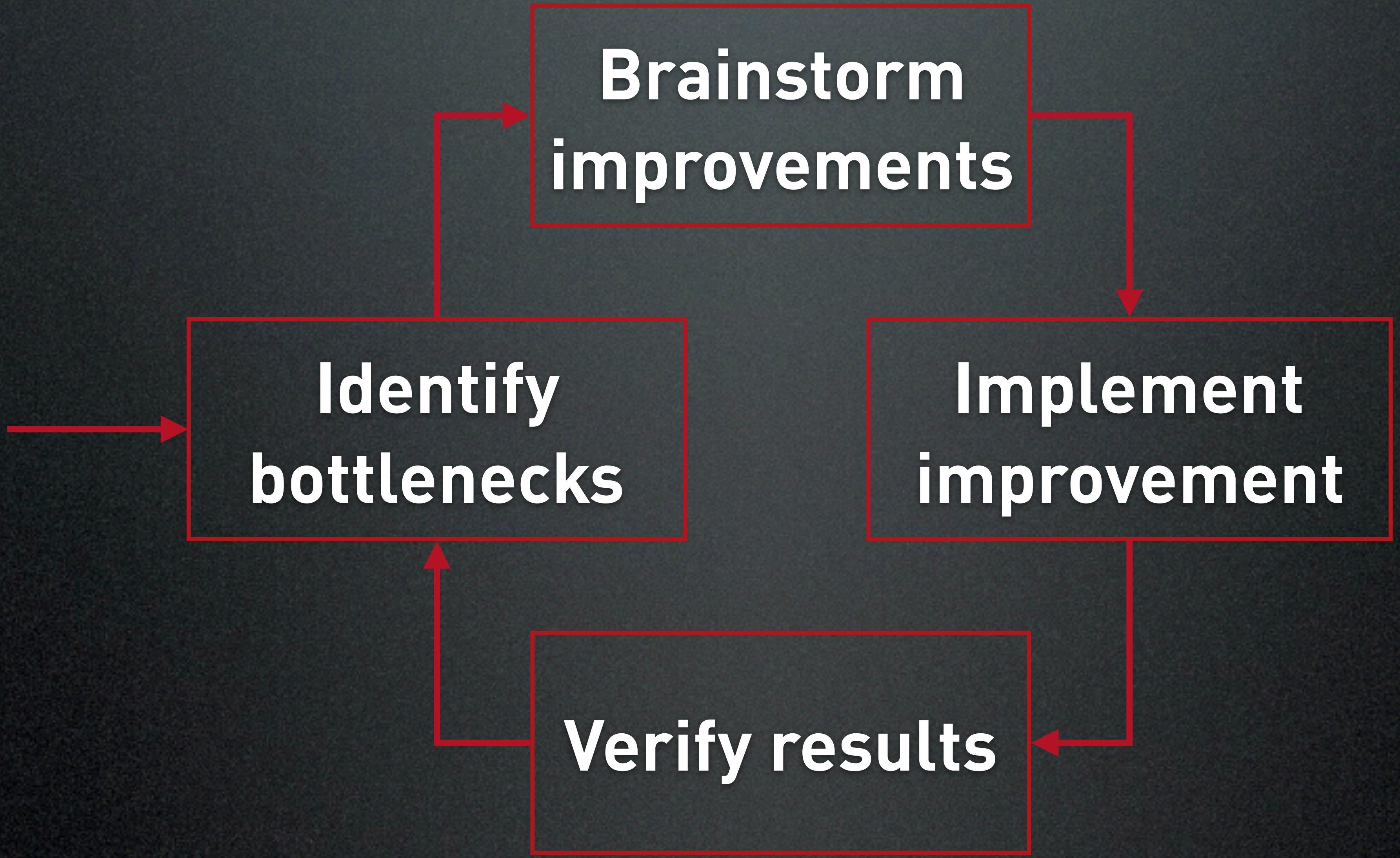
Realistic

Time-bound

Examples:

- A search query should not take more than 350ms.
- On a desktop pc, the homepage's first view should be fully loaded in under 3 seconds, repeat views in under 1,5 second.

Improve performance



Improve performance - Identify Bottlenecks

Measure current performance

- Browser request timeline
- Benchmark: Apache benchmark, Jmeter, etc.
- Profile: Xhprof, Xdebug, etc.

What is my environment telling me?

- High load?
- Slow queries?
- etc.

Improve performance - Brainstorm improvements

Make a list of possible improvements.

- What is essential for your stakeholder and what is not?
- What components make up this page?
  - Custom code/exotic modules required?
  - Many or expensive SQL Queries?
  - Many images?
  - etc.

Think abstract, looking at code/configuration distracts you from the real problems.

Improve performance - Implement improvement

Now it’s time to look at code

- Pick one change
- Highest expected gain first
- Implement it

Improve performance - Verify results

Compare new performance to previously measured performance.

- No improvement? -> Discard the change -> Next iteration

Compare new performance to goals.

- Still sub-par performance? -> Next iteration
- Performance meets goals? -> Go celebrate!

Verify results

- Do the results meet your goals?
- Did you find possibilities for structural improvement?

Performance

Improvements and examples

Measurable performance

Reduce page load time

- Reduce number of requests
- Optimise code

Reduce resource usage & response time

- Optimise configuration
- Queue expensive operations
- Offloading the database
- Caching

Measurable performance - Reduce number of requests

Why?

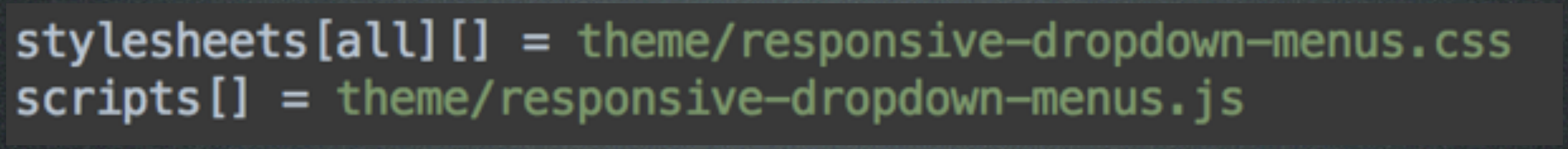
- Fetching resources takes time
- Each request takes up server resources
- Especially requests requiring Drupal to bootstrap

Measurable performance - Reduce number of requests

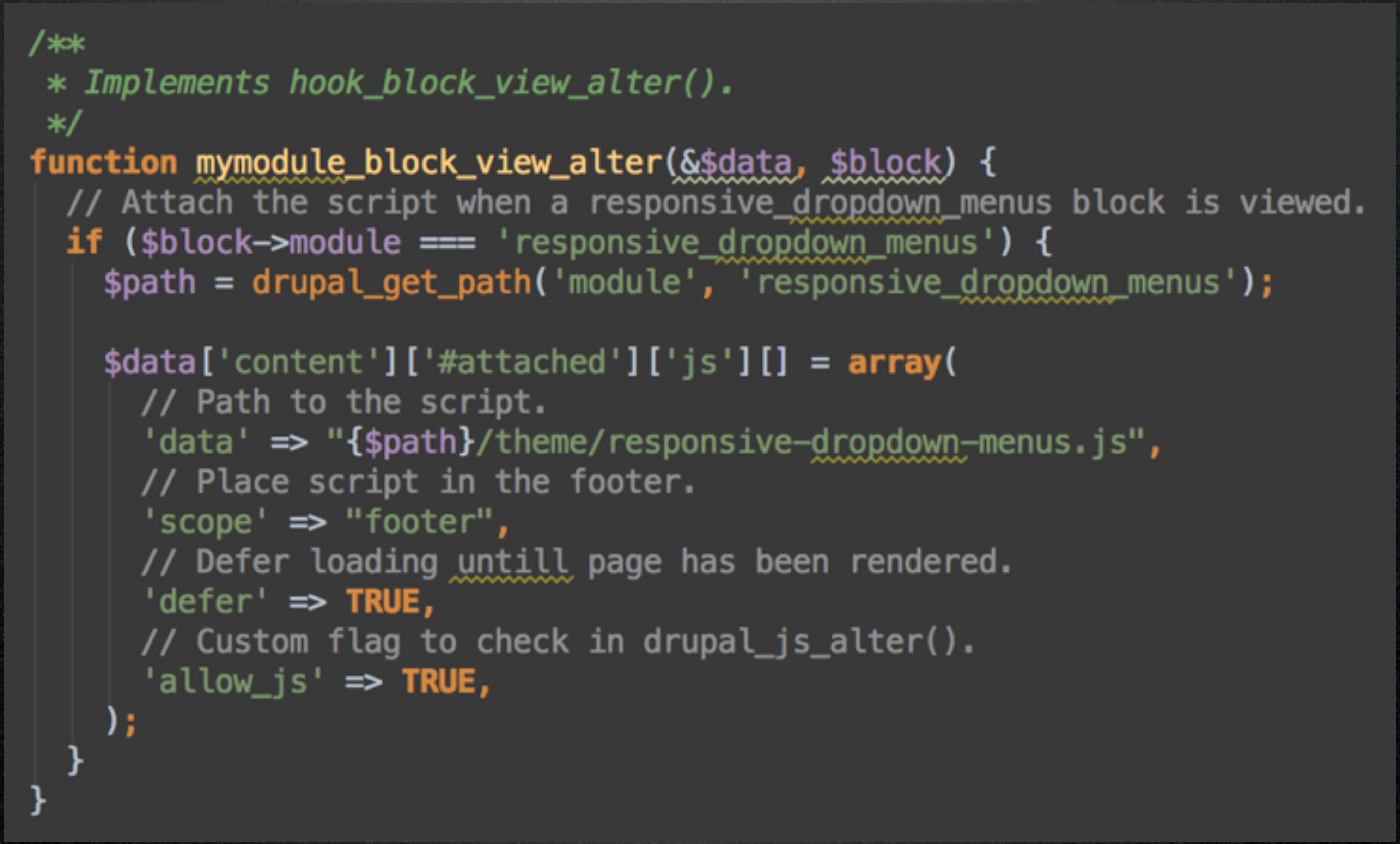
How?

- Combine images used in css into 1 sprite.
- Remove unused javascript/css
- Turn on javascript/css aggregation

Remove unused javascript/css

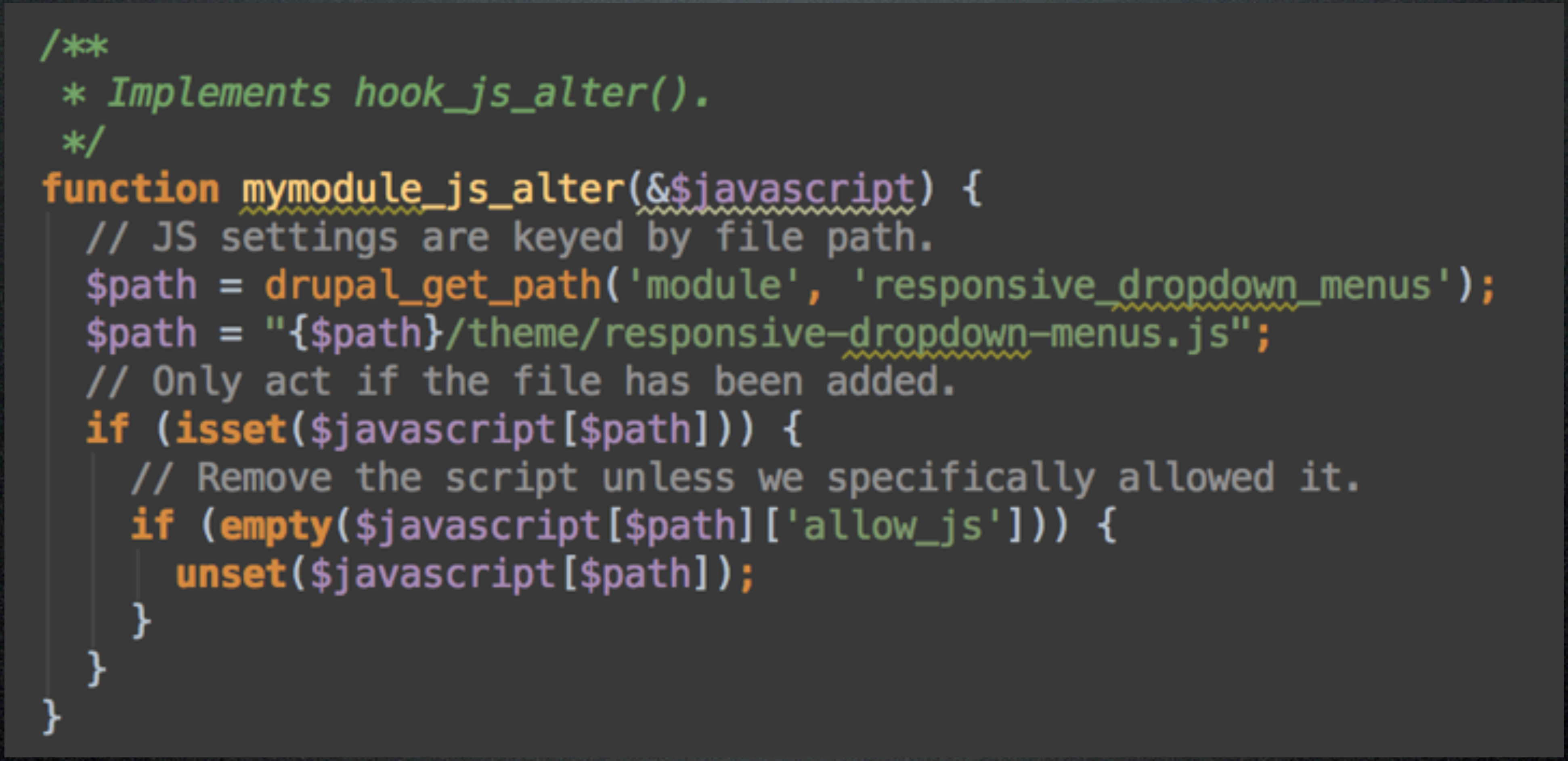


```
stylesheets[all][] = theme/responsive-dropdown-menus.css
scripts[] = theme/responsive-dropdown-menus.js
```



```
/**
 * Implements hook_block_view_alter().
 */
function mymodule_block_view_alter(&$data, $block) {
  // Attach the script when a responsive_dropdown_menus block is viewed.
  if ($block->module === 'responsive_dropdown_menus') {
    $path = drupal_get_path('module', 'responsive_dropdown_menus');

    $data['content']['#attached']['js'][] = array(
      // Path to the script.
      'data' => "{$path}/theme/responsive-dropdown-menus.js",
      // Place script in the footer.
      'scope' => "footer",
      // Defer loading untill page has been rendered.
      'defer' => TRUE,
      // Custom flag to check in drupal_js_alter().
      'allow_js' => TRUE,
    );
  }
}
```

Remove unused javascript/css



```php
/**
 * Implements hook_js_alter().
 */
function mymodule_js_alter(&$javascript) {
  // JS settings are keyed by file path.
  $path = drupal_get_path('module', 'responsive_dropdown_menus');
  $path = "{$path}/theme/responsive-dropdown-menus.js";
  // Only act if the file has been added.
  if (isset($javascript[$path])) {
    // Remove the script unless we specifically allowed it.
    if (empty($javascript[$path]['allow_js'])) {
      unset($javascript[$path]);
    }
  }
}
```

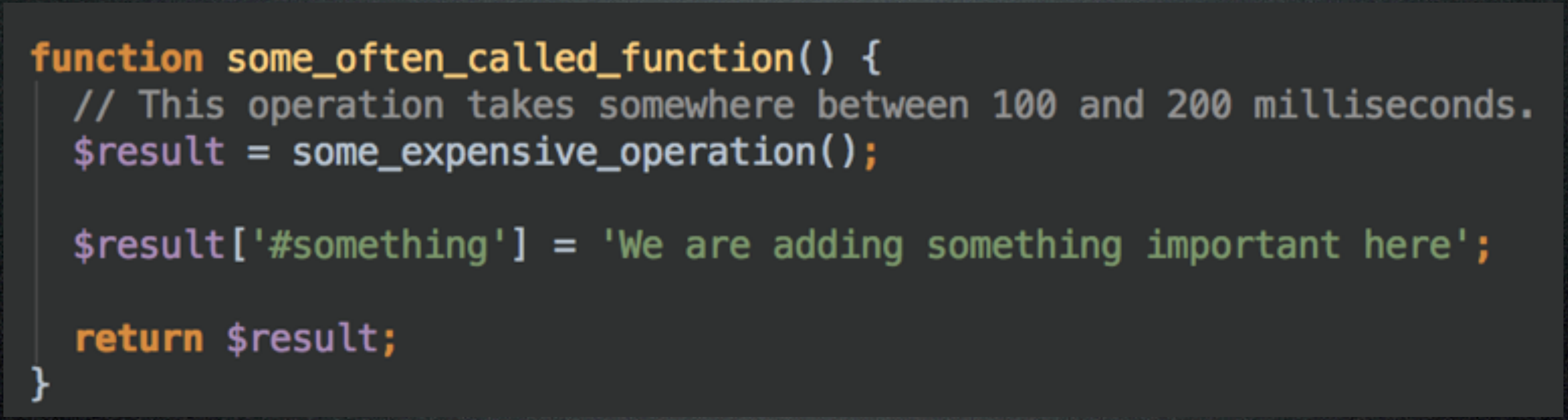
Measurable performance - Optimise code

Cache expensive operations

- static $variable;
- $variable = &drupal_static();

Reduce database queries

Measurable performance - Cache expensive operations

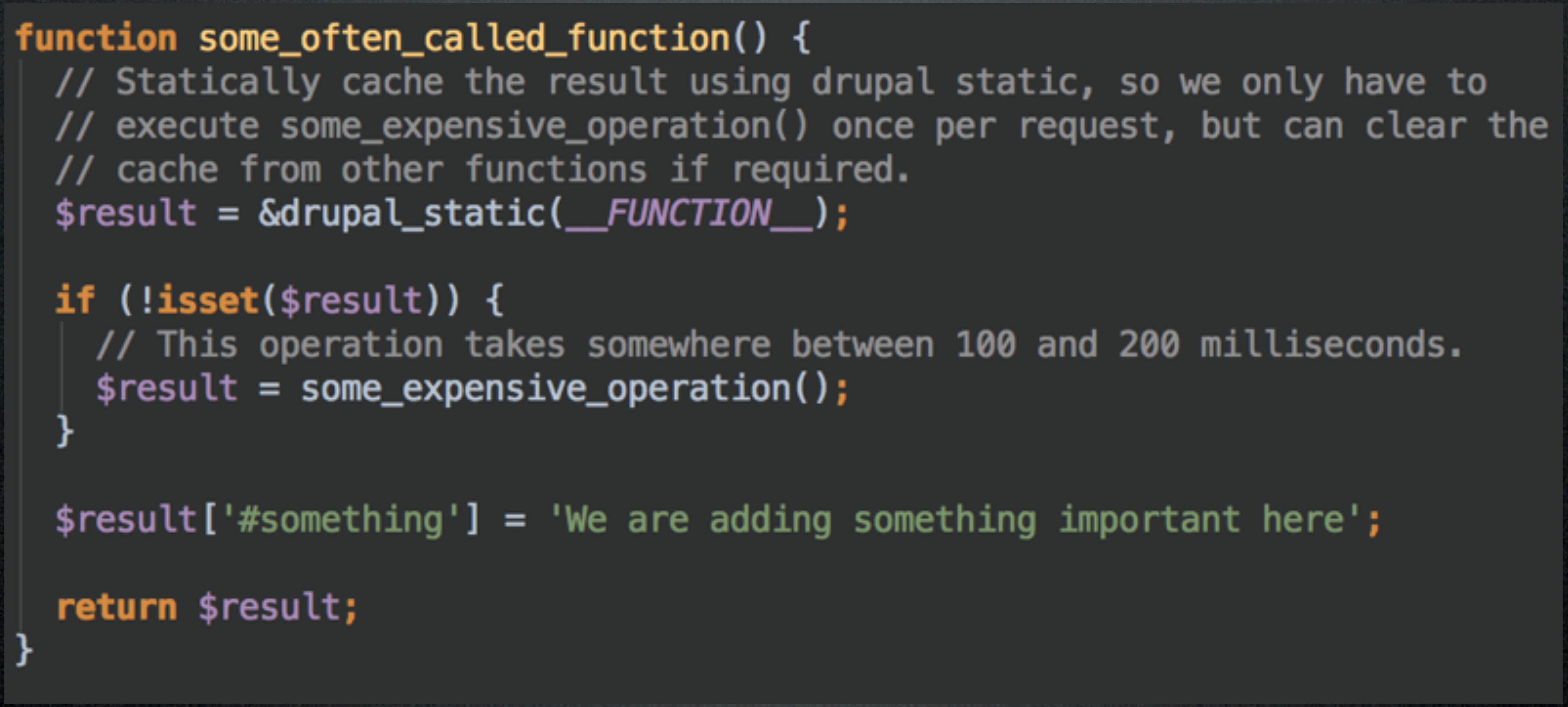


```
function some_often_called_function() {
  // This operation takes somewhere between 100 and 200 milliseconds.
  $result = some_expensive_operation();

  $result['#something'] = 'We are adding something important here';

  return $result;
}
```

Measurable performance - Cache expensive operations

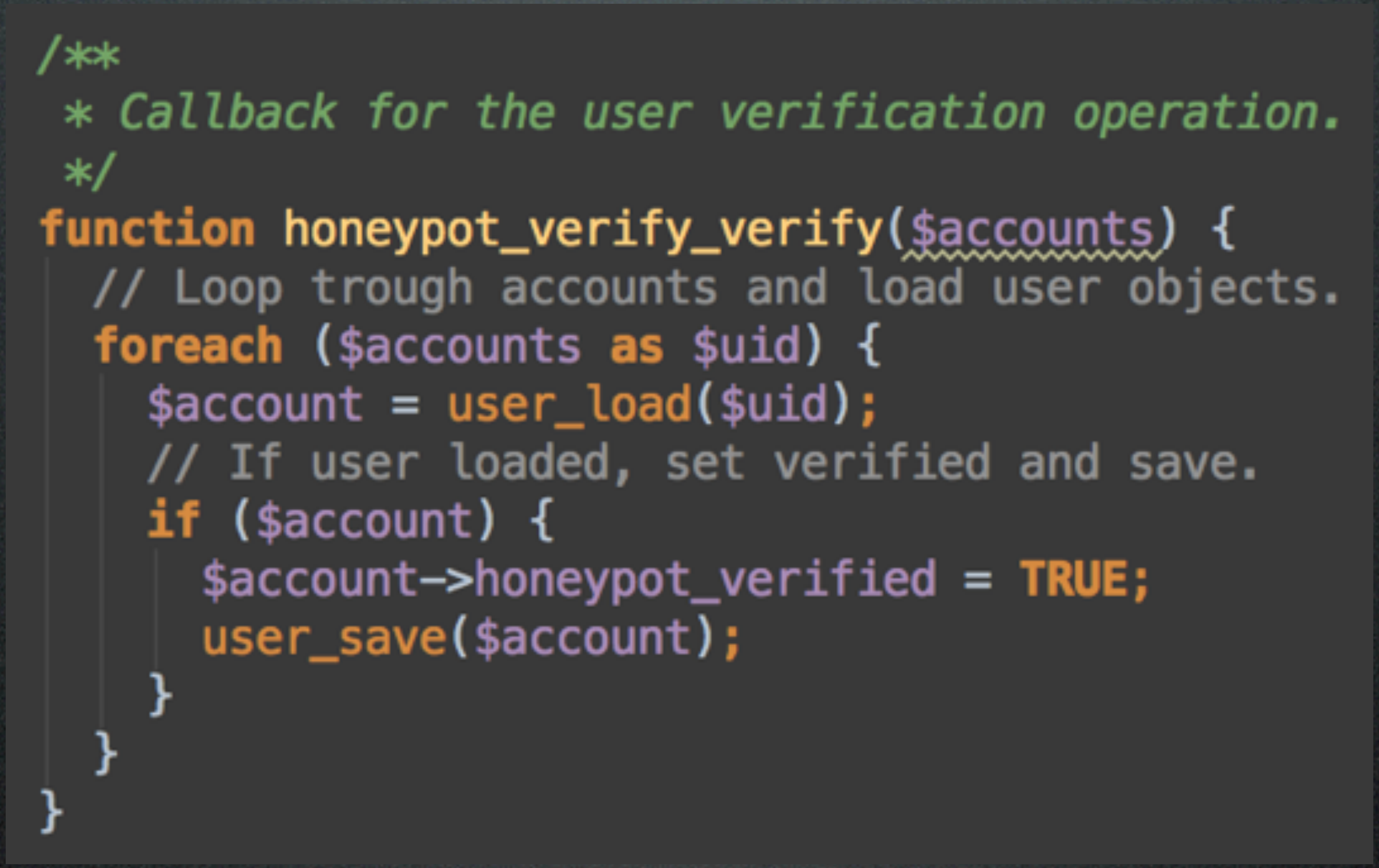
```php
function some_often_called_function() {
  // Statically cache the result, so we only have to execute
  // some_expensive_operation() once per request.
  static $result;

  if (!isset($result)) {
    // This operation takes somewhere between 100 and 200 milliseconds.
    $result = some_expensive_operation();
  }

  $result['#something'] = 'We are adding something important here';

  return $result;
}
```

Measurable performance - Cache expensive operations



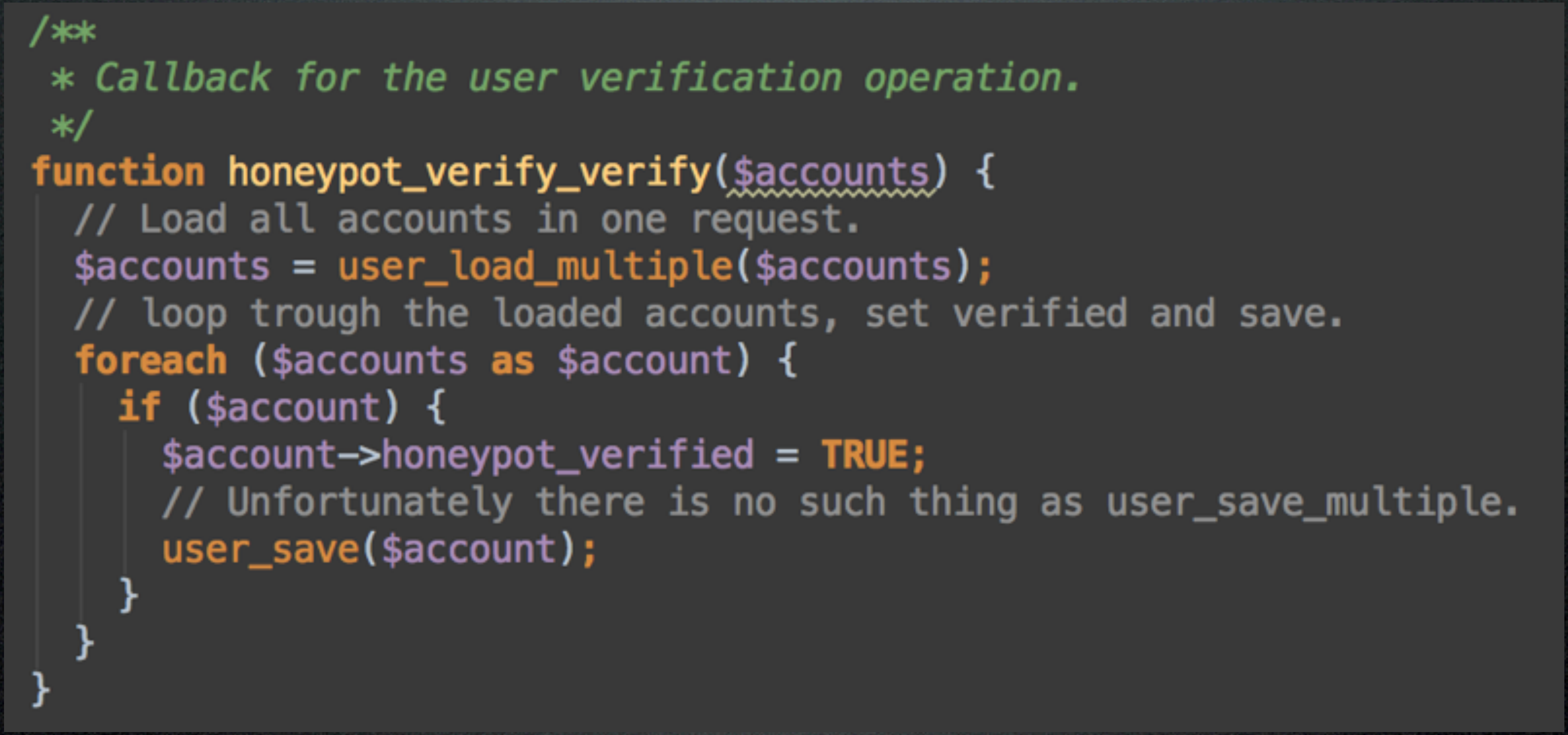
```php
function some_often_called_function() {
  // Statically cache the result using drupal static, so we only have to
  // execute some_expensive_operation() once per request, but can clear the
  // cache from other functions if required.
  $result = &drupal_static(__FUNCTION__);

  if (!isset($result)) {
    // This operation takes somewhere between 100 and 200 milliseconds.
    $result = some_expensive_operation();
  }

  $result['#something'] = 'We are adding something important here';

  return $result;
}
```

Measurable performance - Reduce database queries



```
/**
 * Callback for the user verification operation.
 */
function honeypot_verify_verify($accounts) {
  // Loop trough accounts and load user objects.
  foreach ($accounts as $uid) {
    $account = user_load($uid);
    // If user loaded, set verified and save.
    if ($account) {
      $account->honeypot_verified = TRUE;
      user_save($account);
    }
  }
}
```

Measurable performance - Reduce database queries


```php
/**
 * Callback for the user verification operation.
 */
function honeypot_verify_verify($accounts) {
  // Load all accounts in one request.
  $accounts = user_load_multiple($accounts);
  // loop trough the loaded accounts, set verified and save.
  foreach ($accounts as $account) {
    if ($account) {
      $account->honeypot_verified = TRUE;
      // Unfortunately there is no such thing as user_save_multiple.
      user_save($account);
    }
  }
}
```

Measurable performance - Optimise configuration

Disable/uninstall

- Unused modules
- Development modules (devel, devel_themer, etc.)
- UI modules (views_ui, field_ui, *_ui, etc.)

Sometimes 3 specific modules beat 1 multipurpose

Measurable performance - Optimise configuration

Views

- Remove unneeded relationships
- Prevent using DISTINCT
- Prevent COUNTing all rows
- Enable query caching
- Enable block caching

Log using syslog instead of dblog

Measurable performance - Queue expensive operations

- Sending e-mails
- Generating PDF files
- Aggregating votes/reviews/etc.

Example:

https://www.computerminds.co.uk/drupal-code/drupal-queues

Measurable performance - Offloading the database

Change caching backend

- Memcache(d)
- Redis

Change search engine

- Solr
- Elasticsearch

Measurable performance - Caching

Drupal

- page cache
- Entity cache
    - Render cache

Supporting software

- Opcode caching
- Varnish

Leverage browser caching

- Browser caching (ETags, Expires, Vary, etc.)

Measurable performance - Caching

Why mention caching last?

- Only masks problems
- Easy to get wrong.
- Session breaks caching (Ignore for D8)
- Highly administered sites don't cache well (Ignore for D8)

Perceived performance

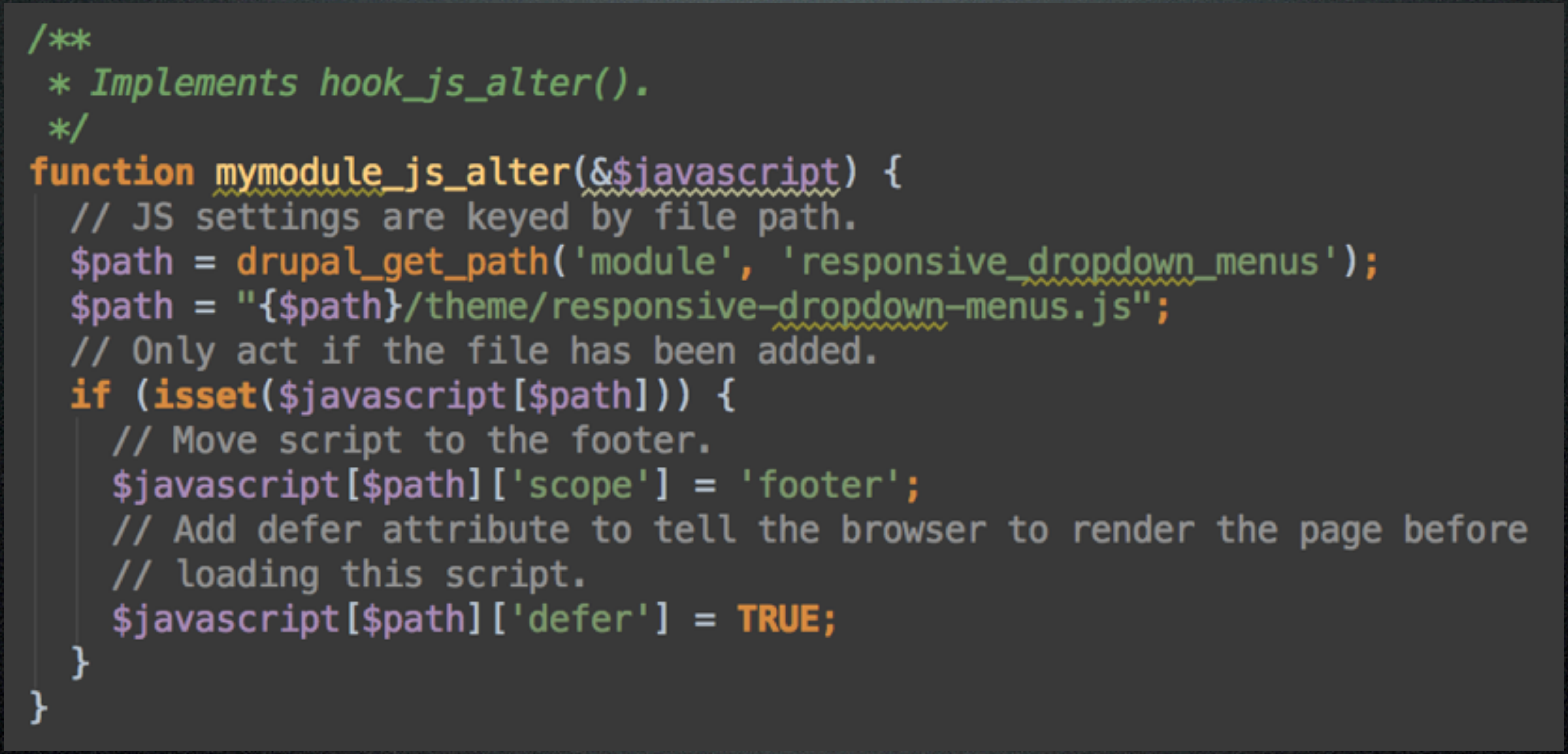
Rendering of useful content

- Relocate (and defer) javascript
- Lazyloading

Activity feedback

- Throbber
- Progress bar

Perceived performance - Relocate javascript



```php
/**
 * Implements hook_js_alter().
 */
function mymodule_js_alter(&$javascript) {
  // JS settings are keyed by file path.
  $path = drupal_get_path('module', 'responsive_dropdown_menus');
  $path = "{$path}/theme/responsive-dropdown-menus.js";
  // Only act if the file has been added.
  if (isset($javascript[$path])) {
    // Move script to the footer.
    $javascript[$path]['scope'] = 'footer';
    // Add defer attribute to tell the browser to render the page before
    // loading this script.
    $javascript[$path]['defer'] = TRUE;
  }
}
```

What to take away?

from this talk

What to take away?

- Performance is context dependent
- Focus on your goals
- Think about performance while developing
  - But don’t optimise too early

Thank you,
any questions?