Magento 2.2

Frontend Basics

March 13, 2018

Plan

Plan

Plan

Basics

- A **theme** provides the look and feel for an application area (front, admin).

Basics

- A **theme** provides the look and feel for an application area (front, admin).
- Naming convention:
  app/design/<area>/<Vendor>/<theme>

Basics

- A **theme** provides the look and feel for an application area (front, admin).
- Naming convention:
  app/design/<area>/<Vendor>/<theme>
- Default themes:
  - **Magento/backend**: admin theme
  - **Magento/luma**: frontend demo theme
  - **Magento/blank**: base for custom frontend theme creation

Basics

- A theme will usually contain the following files:

Basics

- A theme will usually contain the following files:
  - LESS and CSS files
  - JavaScript files
  - Templates
  - Layout files
  - Images
  - Translations specific to the theme

Basics

- Current theme is displayed in the admin panel
  Content > Configuration
  - Applied Theme: theme used in the selected scope
  - User-Agent Rules: theme to use for specific user agents
- Available themes are listed in Content > Themes

Themes are stored in the **theme** table

Field	Type	Null	Key	Default	Extra
theme_id	int(10) unsigned	NO	PRI	NULL	auto_increment
parent_id	int(11)	YES		NULL	
theme_path	varchar(255)	YES		NULL	
theme_title	varchar(255)	NO		NULL	
preview_image	varchar(255)	YES		NULL	
is_featured	tinyint(1)	NO		0	
area	varchar(255)	NO		NULL	
type	smallint(6)	NO		NULL	
code	text	YES		NULL	

Theme structure

```
theme_dir/
|-- Namespace_Module/
|   |-- web/
|   |   |-- css/
|   |   |   |-- source/
|   |-- layout/
|   |   |-- override/
|   |-- templates/
|-- etc/
|-- i18n/
|-- media/
|-- web/
|   |-- css/
|   |   |-- source/
|   |-- fonts/
|   |-- images/
|   |-- js/
|-- composer.json
|-- registration.php
|-- theme.xml
```

Basics

Themes must implement a file named **registration.php**

```php
<?php
\Magento\Framework\Component\ComponentRegistrar::register(
    \Magento\Framework\Component\ComponentRegistrar::THEME,
    'frontend/MyVendor/mytheme',
    __DIR__
);
```

Where to put the templates, layouts and static files?

Basics

Where to put the templates, layouts and static files?

- Vendor files (Magento or community module)
  - Always override them in your theme

Basics

Where to put the templates, layouts and static files?

- Vendor files (Magento or community module)
  - Always override them in your theme
- Custom layouts/templates
  - In your modules

Where to put the templates, layouts and static files?

- Vendor files (Magento or community module)
  - Always override them in your theme
- Custom layouts/templates
  - In your modules
- Custom CSS/JS files
  - In your modules when the file is a core component of the module
  - In your custom theme otherwise

Theme creation summary

Theme creation summary

- Create a directory in **app/design/<area>/<Vendor>/<theme>**

Theme creation summary

- Create a directory in **app/design/<area>/<Vendor>/<theme>**
- Add a **theme.xml** file, and optionally **etc/view.xml**

Theme creation summary

- Create a directory in **app/design/<area>/<Vendor>/<theme>**
- Add a **theme.xml** file, and optionally **etc/view.xml**
- Add a **registration.php** file

Theme creation summary

- Create a directory in **app/design/<area>/<Vendor>/<theme>**
- Add a **theme.xml** file, and optionally **etc/view.xml**
- Add a **registration.php** file
- Optionally, add a **composer.json** file

Theme creation summary

- Create a directory in **app/design/<area>/<Vendor>/<theme>**
- Add a **theme.xml** file, and optionally **etc/view.xml**
- Add a **registration.php** file
- Optionally, add a **composer.json** file
- Configure the theme in the admin panel

Plan

Images

The file <**theme_dir**>/**etc/view.xml** can be used to associate image properties to a unique identifier.

Images

The file <**theme_dir**>/**etc/view.xml** can be used to associate image properties to a unique identifier.

```
<images module="Magento_Catalog">
    <image id="unique_image_id" type="image">
        <width>100</width> <!-- Image width in px -->
        <height>100</height> <!-- Image height in px -->
    </image>
</images>
```

Images

This identifier can be used in the PHP code to generate images with the appropriate width and height.

```php
<?php
$imageUrl = $this->imageHelper->init($product, 'product_listing_thumbnail')
    ->getUrl();
```

Images

Image attributes:

- **id** [string] Image identifier

Images

Image attributes:

- **id** [string] Image identifier
- **type** [string] Image type (used for product images)
  - image
  - small_image
  - swatch_image
  - swatch_thumb
  - thumbnail

Images

Image elements:

- <**width**> [string] Image width in pixels.

Images

Image elements:

- <**width**> [string] Image width in pixels.
- <**height**> [string] Image height in pixels.

Images

Image elements:

- <**width**> [string] Image width in pixels.
- <**height**> [string] Image height in pixels.
- <**constrain**> [boolean] Prevents the image from being enlarged when set to true (default: true).

Images

Image elements:

- <**width**> [string] Image width in pixels.
- <**height**> [string] Image height in pixels.
- <**constrain**> [boolean] Prevents the image from being enlarged when set to true (default: true).
- <**aspect_ratio**> [boolean] Prevents the aspect ratio of the image from being modifed when set to true (default: true).

Images

Image elements:

- <**width**> [string] Image width in pixels.
- <**height**> [string] Image height in pixels.
- <**constrain**> [boolean] Prevents the image from being enlarged when set to true (default: true).
- <**aspect_ratio**> [boolean] Prevents the aspect ratio of the image from being modifed when set to true (default: true).
- <**frame**> [boolean] If set to true, the transparent background of images is saved.

Images

Image elements:

- <**width**> [string] Image width in pixels.
- <**height**> [string] Image height in pixels.
- <**constrain**> [boolean] Prevents the image from being enlarged when set to true (default: true).
- <**aspect_ratio**> [boolean] Prevents the aspect ratio of the image from being modifed when set to true (default: true).
- <**frame**> [boolean] If set to true, the transparent background of images is saved.
- <**background**> [string] The color for the image background (not applied when transparency is set to true).

Plan

Theme inheritance

- A theme can **extend** another theme.
- Common reasons to extend a theme:
  - To use the parent theme as a basis for customizations.
  - To provide store design updates, like holiday decorations.

Theme inheritance

- A theme can **extend** another theme.
- Common reasons to extend a theme:
  - To use the parent theme as a basis for customizations.
  - To provide store design updates, like holiday decorations.
- The level of theme inheritance is not limited.

Theme inheritance

- A theme can **extend** another theme.
- Common reasons to extend a theme:
  - To use the parent theme as a basis for customizations.
  - To provide store design updates, like holiday decorations.
- The level of theme inheritance is not limited.
- A theme can override files declared in the parent theme(s) or in a module (fallback mechanism).

Theme inheritance

The parent theme is specified in the **parent** node of the **theme.xml** file.

Theme inheritance

The parent theme is specified in the **parent** node of the **theme.xml** file.

```
<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="urn:magento:framework:Config/etc/theme.xsd">
    <title>Theme Name</title>
    <parent>Magento/blank</parent>
    <media>
        <preview_image>media/preview.jpg</preview_image>
    </media>
</theme>
```

Plan

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.
- When a static file is requested with no module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/ (e.g. fr_FR)

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.
- When a static file is requested with no module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/ (e.g. fr_FR)
  - <theme_dir>/web/

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.
- When a static file is requested with no module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/ (e.g. fr_FR)
  - <theme_dir>/web/
  - <parent_dir>/web/i18n/<locale>/

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.
- When a static file is requested with no module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/ (e.g. fr_FR)
  - <theme_dir>/web/
  - <parent_dir>/web/i18n/<locale>/
  - <parent_dir>/web/

Theme inheritance for static files

- **Static files** are styles (less, css), JavaScript, images and fonts.
- When a static file is requested with no module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/ (e.g. fr_FR)
  - <theme_dir>/web/
  - <parent_dir>/web/i18n/<locale>/
  - <parent_dir>/web/
  - lib/web/

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:
  - <theme_dir>/web/i18n/fr_FR/css/source/_theme.less

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:
  - <theme_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <theme_dir>/web/css/source/_theme.less

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:
  - <theme_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <theme_dir>/web/css/source/_theme.less
  - <parent_dir>/web/i18n/fr_FR/css/source/_theme.less

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:
  - <theme_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <theme_dir>/web/css/source/_theme.less
  - <parent_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <parent_dir>/web/css/source/_theme.less

Theme inheritance for static files

- Example with the static file **source/_theme.less** and the locale **fr_FR**:
  - <theme_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <theme_dir>/web/css/source/_theme.less
  - <parent_dir>/web/i18n/fr_FR/css/source/_theme.less
  - <parent_dir>/web/css/source/_theme.less
  - lib/web/css/source/_theme.less

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <theme_dir>/<Namespace_Module>/web/

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <theme_dir>/<Namespace_Module>/web/
  - <parent_dir>/web/i18n/<locale>/<Namespace_Module>/

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <theme_dir>/<Namespace_Module>/web/
  - <parent_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <parent_dir>/<Namespace_Module>/web/

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <theme_dir>/<Namespace_Module>/web/
  - <parent_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <parent_dir>/<Namespace_Module>/web/
  - <module_dir>/view/<area>/web/ (e.g. frontend)

Theme inheritance for static files

- When a static file is requested with a module context, Magento looks in the following directories until the file is found:
  - <theme_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <theme_dir>/<Namespace_Module>/web/
  - <parent_dir>/web/i18n/<locale>/<Namespace_Module>/
  - <parent_dir>/<Namespace_Module>/web/
  - <module_dir>/view/<area>/web/ (e.g. frontend)
  - <module_dir>/view/base/web/

Plan

Theme inheritance for templates

- There is always a module context when a template file is requested.

Theme inheritance for templates

- There is always a module context when a template file is requested.
- The fallback scheme is the following:

Theme inheritance for templates

- There is always a module context when a template file is requested.
- The fallback scheme is the following:
  - <theme_dir>/<Namespace_Module>/templates/

Theme inheritance for templates

- There is always a module context when a template file is requested.
- The fallback scheme is the following:
  - <theme_dir>/<Namespace_Module>/templates/
  - <parent_dir>/<Namespace_Module>/templates/

Theme inheritance for templates

- There is always a module context when a template file is requested.
- The fallback scheme is the following:
  - <theme_dir>/<Namespace_Module>/templates/
  - <parent_dir>/<Namespace_Module>/templates/
  - <module_dir>/view/<area>/templates/ (e.g. frontend)

Theme inheritance for templates

- There is always a module context when a template file is requested.
- The fallback scheme is the following:
  - <theme_dir>/<Namespace_Module>/templates/
  - <parent_dir>/<Namespace_Module>/templates/
  - <module_dir>/view/<area>/templates/ (e.g. frontend)
  - <module_dir>/view/base/templates/

Plan

Theme inheritance for layouts

- There is always a module context when a layout file is requested.

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:
  - <theme_dir>/<Namespace_Module>/layout/

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:
  - <theme_dir>/<Namespace_Module>/layout/
  - <parent_dir>/<Namespace_Module>/layout/

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:
  - <theme_dir>/<Namespace_Module>/layout/
  - <parent_dir>/<Namespace_Module>/layout/
  - <module_dir>/view/<area>/layout/ (e.g. frontend)

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:
  - <theme_dir>/<Namespace_Module>/layout/
  - <parent_dir>/<Namespace_Module>/layout/
  - <module_dir>/view/<area>/layout/ (e.g. frontend)
  - <module_dir>/view/base/layout/

Theme inheritance for layouts

- There is always a module context when a layout file is requested.
- The layouts processing mechanism works differently than other file types.
- Magento parses **all** the files found, in the following order:
  - <theme_dir>/<Namespace_Module>/layout/
  - <parent_dir>/<Namespace_Module>/layout/
  - <module_dir>/view/<area>/layout/ (e.g. frontend)
  - <module_dir>/view/base/layout/

Theme inheritance for layouts

- To completely override a layout file, it must be placed into the <Namespace_Module>/layout/override/ directory of the theme.

Theme inheritance for layouts

- To completely override a layout file, it must be placed into the <Namespace_Module>/layout/override/ directory of the theme.
- Overriding a layout file is *NOT* recommended.

Plan

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.
  - Vendor: Training

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.
  - Vendor: Training
  - Theme: default

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.
  - Vendor: Training
  - Theme: default
  - Path: app/design/frontend/Training/default

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.
  - Vendor: Training
  - Theme: default
  - Path: app/design/frontend/Training/default
- Files to create:
  - registration.php
  - theme.xml
  - etc/view.xml

Exercice

- Exercice: create a custom frontend theme that extends the Magento/blank theme.
  - Vendor: Training
  - Theme: default
  - Path: app/design/frontend/Training/default
- Files to create:
  - registration.php
  - theme.xml
  - etc/view.xml
- Select your theme in the admin configuration.

Exercice

registration.php file

```php
<?php
\Magento\Framework\Component\ComponentRegistrar::register(
    \Magento\Framework\Component\ComponentRegistrar::THEME,
    'frontend/Training/default',
    __DIR__
);
```

Exercice

theme.xml file

```
<theme xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="urn:magento:framework:Config/etc/theme.xsd">
    <title>Training</title>
    <parent>Magento/blank</parent>
    <media>
        <preview_image>media/preview.jpg</preview_image>
    </media>
</theme>
```

Plan

Plan

Basics

- **Grunt** is a task runner that can be used to compile .less files.

Basics

- **Grunt** is a task runner that can be used to compile .less files.
- Grunt is managed by the file **Gruntfile.js**

Basics

- **Grunt** is a task runner that can be used to compile .less files.
- Grunt is managed by the file **Gruntfile.js**
- To use Grunt, Magento must be set in **developer** or **default** mode.

Theme config in dev/tools/grunt/configs/themes.js

```
module.exports = {
    blank: {
        area: 'frontend',
        name: 'Magento/blank',
        locale: 'en_US',
        files: [
            'css/styles-m',
            'css/styles-l',
            'css/email',
            'css/email-inline'
        ],
        dsl: 'less'
    }
},
// ...
```

Good practice: put your themes in a config file, e.g. dev/tools/grunt/configs/themes.smile.js

```
module.exports = {
    training: {
        area: 'frontend',
        name: 'MyVendor/mytheme',
        locale: 'fr_FR',
        files: [
            'css/styles-m',
            'css/styles-l',
            'css/email',
            'css/email-inline'
        ],
        dsl: 'less'
    }
},
// ...
```

Plan

Commands

Available commands:

- **grunt clean:<theme>**

  Removes the theme related static files in pub/static/ and var/ directories

Commands

Available commands:

- **grunt clean:<theme>**

  Removes the theme related static files in pub/static/ and var/ directories

- **grunt exec:<theme>**

  Creates symlinks to the source files to the pub/static/frontend/<Vendor>/<theme>/<locale> directory

Commands

Available commands:

- **grunt clean:<theme>**

  Removes the theme related static files in pub/static/ and var/ directories

- **grunt exec:<theme>**

  Creates symlinks to the source files to the pub/static/frontend/<Vendor>/<theme>/<locale> directory

- **grunt less:<theme>**

  Compiles .css files using the symlinks published with grunt:exec

Commands

Available commands:

- **grunt clean:<theme>**

  Removes the theme related static files in pub/static/ and var/ directories

- **grunt exec:<theme>**

  Creates symlinks to the source files to the pub/static/frontend/<Vendor>/<theme>/<locale> directory

- **grunt less:<theme>**

  Compiles .css files using the symlinks published with grunt:exec

- **grunt watch**

  Tracks the changes in the source files and recompiles .css files in real time

Commands

When to use commands:

- After adding files from your theme:

```
grunt exec:<theme>
grunt less:<theme>
```

Commands

When to use commands:

- After adding files from your theme:

```
grunt exec:<theme>
grunt less:<theme>
```

- After removing files from your theme:

```
grunt clean:<theme>
grunt exec:<theme>
grunt less:<theme>
```

Commands

When to use commands:

- After adding files from your theme:

```
grunt exec:<theme>
grunt less:<theme>
```

- After removing files from your theme:

```
grunt clean:<theme>
grunt exec:<theme>
grunt less:<theme>
```

- After updading any .less file:

```
grunt less:<theme>
or grunt:watch
```

Plan

3 CSS Preprocessing
- Less Syntax
- Magento Import
- Customizing Styles

Plan

Less Syntax

- Import syntax: `@import (keyword) "filename";`

Less Syntax

- Import syntax: `@import (keyword) "filename";`
- Allows to import additional .less files

Less Syntax

- Import syntax: `@import (keyword) "filename";`
- Allows to import additional .less files
- May be treated differently depending on the file extension
  - .css: treated as CSS

Less Syntax

- Import syntax: `@import (keyword) "filename";`
- Allows to import additional .less files
- May be treated differently depending on the file extension
  - .css: treated as CSS
  - Any other extension: treated as Less and imported

Less Syntax

- Import syntax: `@import (keyword) "filename";`
- Allows to import additional .less files
- May be treated differently depending on the file extension
  - .css: treated as CSS
  - Any other extension: treated as Less and imported
  - No extension: .less automatically appended, treated as Less and imported

```
@import "foo";       // foo.less is imported
@import "foo.less";  // foo.less is imported
@import "foo.php";   // foo.php imported as a less file
@import "foo.css";   // statement left in place, as-is
```

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it
- less: treat the file as a Less file, no matter what the file extension

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it
- less: treat the file as a Less file, no matter what the file extension
- css: treat the file as a CSS file, no matter what the file extension

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it
- less: treat the file as a Less file, no matter what the file extension
- css: treat the file as a CSS file, no matter what the file extension
- once: only include the file once (this is default behavior)

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it
- less: treat the file as a Less file, no matter what the file extension
- css: treat the file as a CSS file, no matter what the file extension
- once: only include the file once (this is default behavior)
- multiple: include the file multiple times

Less Syntax

Import options (keyword):

- reference: use a Less file but do not output it
- inline: include the source file in the output but do not process it
- less: treat the file as a Less file, no matter what the file extension
- css: treat the file as a CSS file, no matter what the file extension
- once: only include the file once (this is default behavior)
- multiple: include the file multiple times
- optional: continue compiling when file is not found

Less Syntax

Nested rules

```
#header {
    color: black;
    .navigation {
        font-size: 12px;
    }
    .logo {
        width: 300px;
    }
}
```

Compiles to:

```
#header {
    color: black;
}
#header .navigation {
    font-size: 12px;
}
#header .logo {
    width: 300px;
}
```

Less Syntax

Variables

```
@nice-blue: #5B83AD;
@light-blue: @nice-blue + #111;

#header {
    color: @light-blue;
}
```

Compiles to:

```
#header {
    color: #6c94be;
}
```

Less Syntax

Extend

```
nav ul {
    &:extend(.inline);
    background: blue;
}

.inline {
    color: red;
}
```

Compiles to:

```
nav ul {
    background: blue;
}

.inline,
nav ul {
    color: red;
}
```

Mixins

```
.my-hover-mixin() {
    &:hover {
        border: 1px solid red;
    }
}

button {
    .my-hover-mixin();
}
```

Compiles to:

```
button:hover {
    border: 1px solid red;
}
```

Mixins with parameters

```
.border-radius(@radius) {
    -webkit-border-radius: @radius;
    -moz-border-radius: @radius;
    border-radius: @radius;
}

#header {
    .border-radius(4px);
}

.button {
    .border-radius(6px);
}
```

Compiles to:

```
#header {
    -webkit-border-radius: 4px;
    -moz-border-radius: 4px;
    border-radius: 4px;
}

.button {
    -webkit-border-radius: 6px;
    -moz-border-radius: 6px;
    border-radius: 6px;
}
```

Mixins guards

```
.mixin (@a) when (lightness(@a) >= 50%) {
    background-color: black;
}

.mixin (@a) when (lightness(@a) < 50%) {
    background-color: white;
}

.mixin (@a) {
    color: @a;
}

.class1 {
    .mixin(#ddd)
}
.class2 {
    .mixin(#555)
}
```

Compiles to:

```
.class1 {
    background-color: black;
    color: #ddd;
}

.class2 {
    background-color: white;
    color: #555;
}
```

Loops

```
.generate-columns(4);

.generate-columns(@n, @i: 1) when (@i =< @n) {
    .column-@{i} {
        width: (@i * 100% / @n);
    }
    .generate-columns(@n, (@i + 1));
}
```

Compiles to:

```
.column-1 {
    width: 25%;
}
.column-2 {
    width: 50%;
}
.column-3 {
    width: 75%;
}
.column-4 {
    width: 100%;
}
```

Plan

Magento Import

Magento specific directive: @magento_import

Magento Import

Magento specific directive: @magento_import

- Allows including multiple files by a name pattern.

Magento Import

Magento specific directive: @magento_import

- Allows including multiple files by a name pattern.
- Used to include files with the same name from multiple locations (modules, themes).

Magento Import

Magento specific directive: @magento_import

- Allows including multiple files by a name pattern.
- Used to include files with the same name from multiple locations (modules, themes).
- Can only be used in the root .less files of a theme

Magento Import

Magento specific directive: @magento_import

- Allows including multiple files by a name pattern.
- Used to include files with the same name from multiple locations (modules, themes).
- Can only be used in the root .less files of a theme
- It MUST be commented out with two slashes

Magento Import

@magento_import example in vendor/magento/theme-frontend-blank/web/css/styles-l.less

```
@import 'source/_reset';
@import '_styles';

//
// Custom Magento LESS import directives
// ----------------------------------------

//@magento_import 'source/_module.less'; // Theme modules
//@magento_import 'source/_widgets.less'; // Theme widgets
//@magento_import 'source/_extend.less'; // Extend for minor customization
```

Magento Import

```
//@magento_import 'source/_widgets.less';
```

Compiles to:

```
@import '../Magento_Catalog/css/source/_widgets.less';
@import '../Magento_Cms/css/source/_widgets.less';
@import '../Magento_Reports/css/source/_widgets.less';
@import '../Magento_Sales/css/source/_widgets.less';
```

Plan

Customizing Styles

How Magento stylesheet files are organized

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles
- <theme_dir>/web/css: theme-specific styles

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles
- <theme_dir>/web/css: theme-specific styles
  - print.less: used to generate styles for the printed version of store pages

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles
- <theme_dir>/web/css: theme-specific styles
  - print.less: used to generate styles for the printed version of store pages
  - _styles.less: composite file that includes all LESS files used in the theme

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles
- <theme_dir>/web/css: theme-specific styles
  - print.less: used to generate styles for the printed version of store pages
  - _styles.less: composite file that includes all LESS files used in the theme
  - styles-m.less: used to generate mobile-specific styles, includes _styles.less

Customizing Styles

How Magento stylesheet files are organized

- <theme_dir>/<Vendor_Module>/web/css: module-specific styles
- <theme_dir>/web/css: theme-specific styles
  - print.less: used to generate styles for the printed version of store pages
  - _styles.less: composite file that includes all LESS files used in the theme
  - styles-m.less: used to generate mobile-specific styles, includes _styles.less
  - styles-l.less: used to generate desktop-specific styles, includes _styles.less

Customizing Styles

Simplest way to extend parent styles:

Customizing Styles

Simplest way to extend parent styles:

- Create a <theme_dir>/web/css/source/_extend.less file.

Customizing Styles

Simplest way to extend parent styles:

- Create a <theme_dir>/web/css/source/_extend.less file.
- Write your LESS code in this file.

```
<theme_dir>/
|-- web/
|   |-- css/
|   |   |-- source/
|   |   |   |-- _extend.less
```

Customizing Styles

- You can also extend any .less file from the source folder.

Customizing Styles

- You can also extend any .less file from the source folder.
- For example, to extend the buttons.less file:

```
<theme_dir>/
|-- web/
|   |-- css/
|   |   |-- source/
|   |   |   |-- _buttons_extend.less
```

Customizing Styles

Simplest way to override parent styles:

Customizing Styles

Simplest way to override parent styles:

- Create a <theme_dir>/web/css/source/_theme.less file.

Customizing Styles

Simplest way to override parent styles:

- Create a <theme_dir>/web/css/source/_theme.less file.
- Declare the variables you want to override in this file.

```
<theme_dir>/
|-- web/
|   |-- css/
|   |   |-- source/
|   |   |   |-- _theme.less
```

Customizing Styles

Simplest way to override parent styles:

- Create a <theme_dir>/web/css/source/_theme.less file.
- Declare the variables you want to override in this file.

```
<theme_dir>/
|-- web/
|   |-- css/
|   |   |-- source/
|   |   |   |-- _theme.less
```

While not recommended, you can override any .less file from the source folder.

Customizing Styles

For complex themes, you can add your own @import declarations to use project-specific .less files.

Customizing Styles

For complex themes, you can add your own @import declarations to use project-specific .less files.

```
<theme_dir>/
|-- web/
|   |-- css/
|   |   |-- layout/
|   |   |   |-- _header.less
|   |   |   |-- _footer.less
|   |   |-- module/
|   |   |   |-- _catalog-product.less
|   |   |   |-- _checkout-cart.less
```

Plan

4 JavaScript
- RequireJS
- Replace a component
- Extend a component
- Mixins

Plan

RequireJS

- Magento 2 uses the RequireJS library to manage dependencies between javascript components.

RequireJS

- Magento 2 uses the RequireJS library to manage dependencies between javascript components.
- Two steps to define a JS component:

RequireJS

- Magento 2 uses the RequireJS library to manage dependencies between javascript components.
- Two steps to define a JS component:
  - Declare the component in a javascript file, using the **define** function:

```
define([
    'jquery' // Components to use
], function ($) {
    // Code of your own component
});
```

RequireJS

- Magento 2 uses the RequireJS library to manage dependencies between javascript components.
- Two steps to define a JS component:
  - Declare the component in a javascript file, using the **define** function:

```
define([
    'jquery' // Components to use
], function ($) {
    // Code of your own component
});
```

  - Let RequireJS know about the component by adding the path to its file in a RequireJS config file, named **requirejs-config.js**.

- RequireJS config files can be defined in any theme or module, at the following locations:

RequireJS

- RequireJS config files can be defined in any theme or module, at the following locations:
  - Modules: <module_dir>/view/<area>/requirejs-config.js

RequireJS

- RequireJS config files can be defined in any theme or module, at the following locations:
  - Modules: <module_dir>/view/<area>/requirejs-config.js
  - Themes: <theme_dir>/requirejs-config.js

RequireJS

- RequireJS config files can be defined in any theme or module, at the following locations:
  - Modules: <module_dir>/view/<area>/requirejs-config.js
  - Themes: <theme_dir>/requirejs-config.js
- For every area, all requirejs-config.js files are merged into a single file.

RequireJS

- RequireJS config files can be defined in any theme or module, at the following locations:
  - Modules: <module_dir>/view/<area>/requirejs-config.js
  - Themes: <theme_dir>/requirejs-config.js
- For every area, all requirejs-config.js files are merged into a single file.
- This file is written to the pub/static/requirejs directory.

Example in the Catalog module (in view/base/requirejs-config.js):

```
var config = {
    map: {
        '*': {
            categoryForm:       'Magento_Catalog/catalog/category/form',
            newCategoryDialog:  'Magento_Catalog/js/new-category-dialog',
            categoryTree:       'Magento_Catalog/js/category-tree',
            productGallery:     'Magento_Catalog/js/product-gallery',
            baseImage:          'Magento_Catalog/catalog/base-image-uploader',
            productAttributes:  'Magento_Catalog/catalog/product-attributes'
        }
    },
    deps: [
        'Magento_Catalog/catalog/product'
    ]
};
```

- The **map** variable contains the paths to the JS components.

- The **map** variable contains the paths to the JS components.
- **Values** of the object represent the path to the component.
  'Magento_Catalog/catalog/category/form' is the path to
  <module_catalog_dir>/view/adminhtml/web/catalog/category/form.js

- The **map** variable contains the paths to the JS components.
- **Values** of the object represent the path to the component.
  'Magento_Catalog/catalog/category/form' is the path to
  <module_catalog_dir>/view/adminhtml/web/catalog/category/form.js
- **Keys** of the object represent aliases that can be used instead of the path when requiring a component.

RequireJS

- To load a component, use the **require** function:

```
require(["Magento_ConfigurableProduct/js/configurable"], function (Configurable) {
});
```

- Components are **lazy-loaded** by RequireJS.

RequireJS

- To load a component, use the **require** function:

```
require(["Magento_ConfigurableProduct/js/configurable"], function (Configurable) {
});
```

- Components are **lazy-loaded** by RequireJS.
- A component will be loaded:
  - When it is used with the **require** function

RequireJS

- To load a component, use the **require** function:

```
require(["Magento_ConfigurableProduct/js/configurable"], function (Configurable) {
});
```

- Components are **lazy-loaded** by RequireJS.
- A component will be loaded:
  - When it is used with the **require** function
  - When a component that depends on it is loaded

Plan

4 JavaScript
- RequireJS
- Replace a component
- Extend a component
- Mixins

Replace a component

- To **replace** an existing component with your own component, use the following configuration:

```
var config = {
    "map": {
        "*": {
            "<default_component>": "<custom_component>"
        }
    }
}
```

Replace a component

- To **replace** an existing component with your own component, use the following configuration:

```
var config = {
    "map": {
        "*": {
            "<default_component>": "<custom_component>"
        }
    }
}
```

- Where <default_component> is the name of the component to replace, and <custom_component> is the path to your own component.

Replace a component

Example: replacing the navigation-menu.js component, in the requirejs-config.js file of your module/theme:

```
var config = {
    "map": {
        "*": {
            "menu": "js/navigation-menu",
            "mage/backend/menu": "js/navigation/menu"
        }
    }
};
```

Plan

Extend a component

- If a core component is a jQuery widget or an UI component, your custom component can **extend** it.

Extend a component

- If a core component is a jQuery widget or an UI component, your custom component can **extend** it.
- This will not replace the default component.

Extend a component

- If a core component is a jQuery widget or an UI component, your custom component can **extend** it.
- This will not replace the default component.
- This mechanism enables you to create a **new** component that extends another one.

Extend a component

Extending an UI component:

```
define([
    'Magento_Ui/js/grid/filters/filters'
], function (Filters) {
    return Filters.extend({
        // You can redeclare functions of the parent component or add your own
    });
});
```

Extend a component

Extending a jQuery widget:

```
define([
    'jquery',
    'jquery/ui',
    'Magento_ConfigurableProduct/js/configurable'
], function($){
    $.widget('myproject.configurable', $.mage.configurable, {
        // Code you want to override
    });

    return $.myproject.configurable;
});
```

Plan

Mixins

- You can both **replace** a core component and **extend** it by using **JavaScript mixins**.

Mixins

- You can both **replace** a core component and **extend** it by using **JavaScript mixins**.
- **Replace**: your component will be used instead of the core component.

Mixins

- You can both **replace** a core component and **extend** it by using **JavaScript mixins**.
- **Replace**: your component will be used instead of the core component.
- **Extend**: your component will contain only the new logic.

Mixins

- You can both **replace** a core component and **extend** it by using **JavaScript mixins**.
- **Replace**: your component will be used instead of the core component.
- **Extend**: your component will contain only the new logic.
- A mixin is a function that receives the return value of the core component.

Mixins

- Mixin declaration in requirejs-config.js file:

```
var config = {
    config: {
        mixins: {
            'Magento_Checkout/js/view/payment': {
                'Namespace_Module/js/view/payment-mixin': true
            }
        }
    }
};
```

Mixins

- Mixin declaration in requirejs-config.js file:

```
var config = {
    config: {
        mixins: {
            'Magento_Checkout/js/view/payment': {
                'Namespace_Module/js/view/payment-mixin': true
            }
        }
    }
};
```

- Mixin path:
  <module_dir>/view/frontend/web/js/view/payment-mixin.js

Mixins

Mixin implementation:

```
define(
    [],
function () {
    'use strict';
    return function (target) {
        return target.extend({
            // Add new functions or override existing functions
        });
    };
});
```

Plan

Plan

Responsive Web Design

- The blank and luma themes implement the following breakpoints:

Responsive Web Design

- The blank and luma themes implement the following breakpoints:
  - 320px
  - 480px
  - 640px
  - 768px
  - 1024px
  - 1440px
- It is possible to override breakpoints by overriding the _responsive.less file.

Responsive Web Design

- The blank and luma themes use the following scripts to relocate page elements by breakpoint:

Responsive Web Design

- The blank and luma themes use the following scripts to relocate page elements by breakpoint:
  - responsive.js
  - menu.js
  - matchMedia.js, used by responsive.js and menu.js

Responsive Web Design

- The blank and luma themes use the following scripts to relocate page elements by breakpoint:
  - responsive.js
  - menu.js
  - matchMedia.js, used by responsive.js and menu.js
- You can use these files to add responsive behavior in your custom theme.

Plan

To prevent XSS issues, escape using the following block methods:

- **escapeHtml**

To prevent XSS issues, escape using the following block methods:

- **escapeHtml**
- **escapeQuote**

To prevent XSS issues, escape using the following block methods:

- **escapeHtml**
- **escapeQuote**
- **escapeUrl**

To prevent XSS issues, escape using the following block methods:

- **escapeHtml**
- **escapeQuote**
- **escapeUrl**
- **escapeXssInUrl**

Situations where escaping is NOT necessary

- If a method indicates that the contents are already escaped

```
echo $block->getTitleHtml()
```

Situations where escaping is NOT necessary

- If a method indicates that the contents are already escaped

```
echo $block->getTitleHtml()
```

- Type casting and PHP function **count()**

```
echo (int) $var
```

Situations where escaping is NOT necessary

- If a method indicates that the contents are already escaped

```
echo $block->getTitleHtml()
```

- Type casting and PHP function **count()**

```
echo (int) $var
```

- Hardcoded string variables

```
echo 'some text'
```

Example of a safe template file:

```
<?= $block->getTitleHtml() ?>
<?= $block->escapeHtml($block->getTitle()) ?>
<?= (int) $block->getId() ?>
<?= count($var) ?>
<?= __('Some text') ?>
<a href="<?= $block->escapeXssInUrl($block->getUrl()) ?>">
    <?= $block->getAnchorTextHtml() ?>
</a>
```

Plan