odoo

# Create Themes for Website

By Fabien Pinckaers - Founder & CEO, Odoo

odoo

# Topics

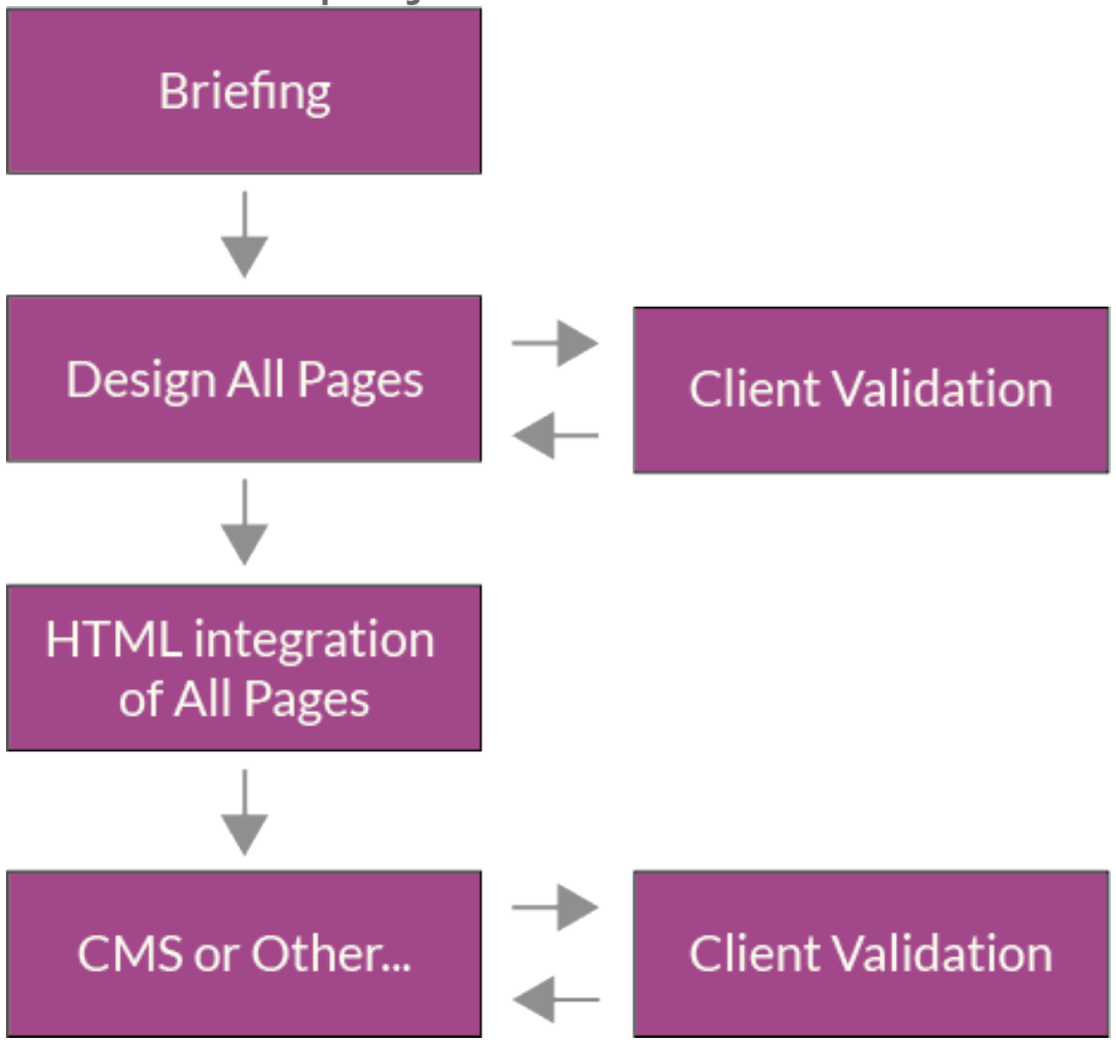
1. Introduction
    - Classical workflow
    - Odoo's CMS workflow
2. Tutorial
    - Starting with a single page
    - Snippets
    - Options
    - Custom Css
3. Examples

odoo

# Introduction

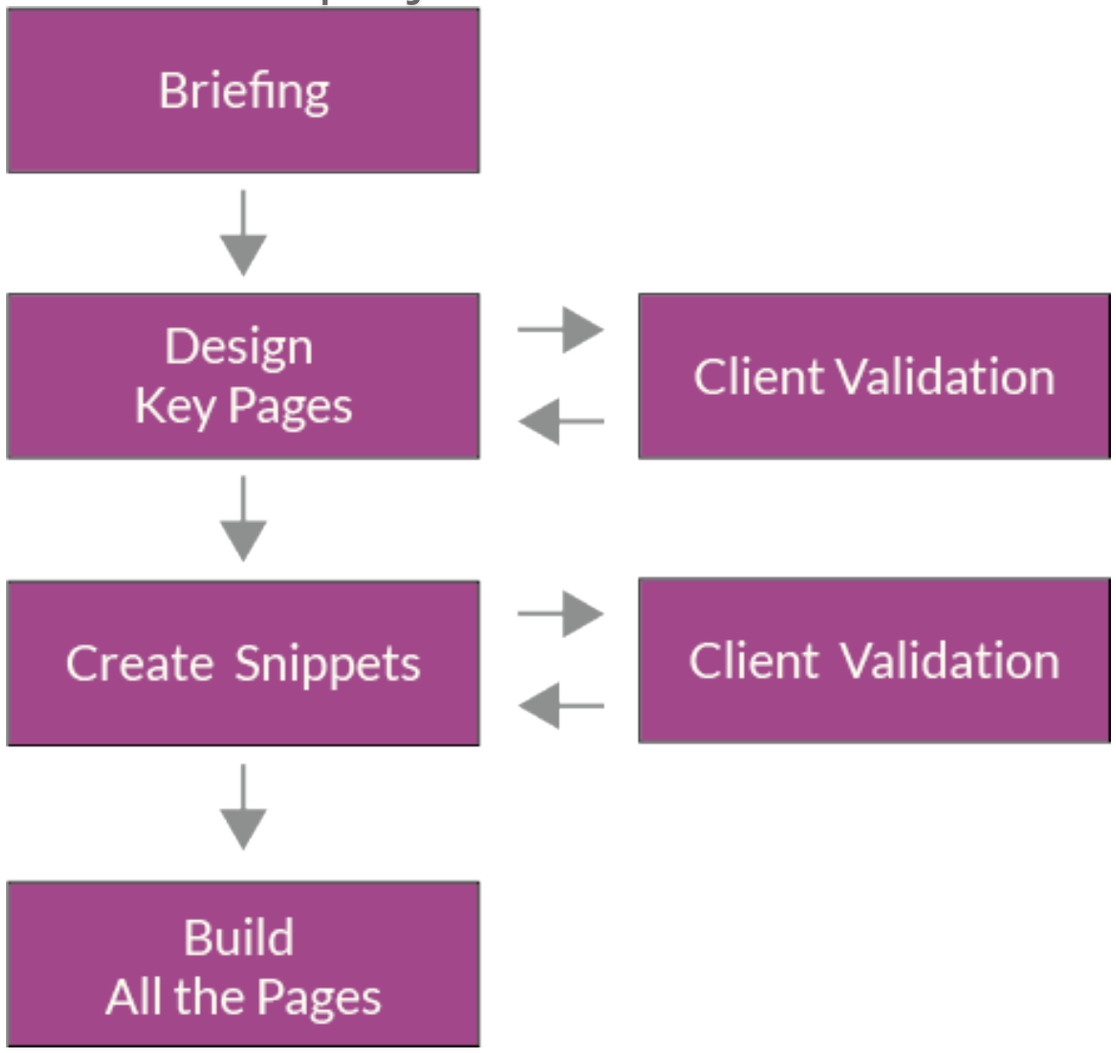
# Classical workflow

Start a new project.



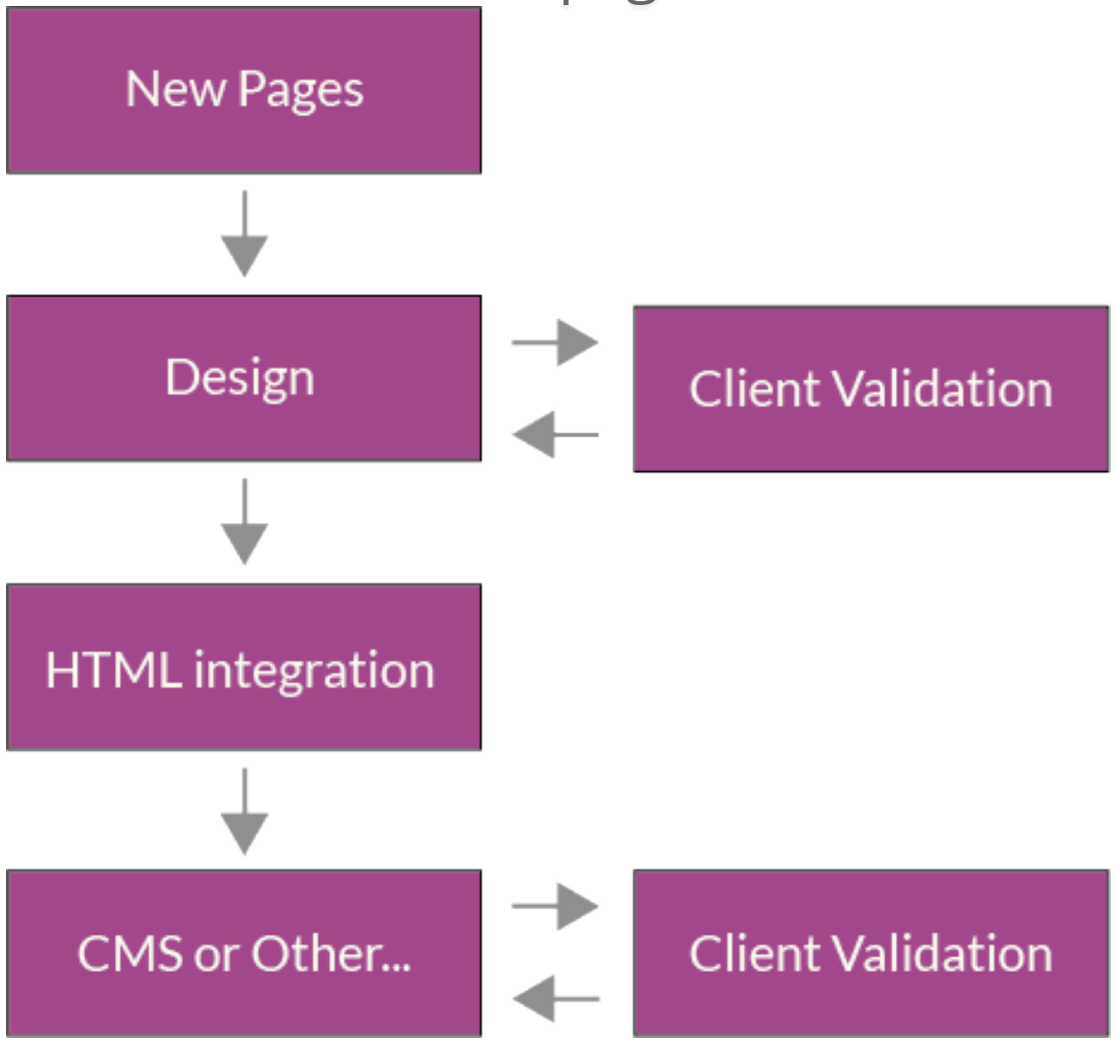
# Working with Odoo CMS

Start a new project.



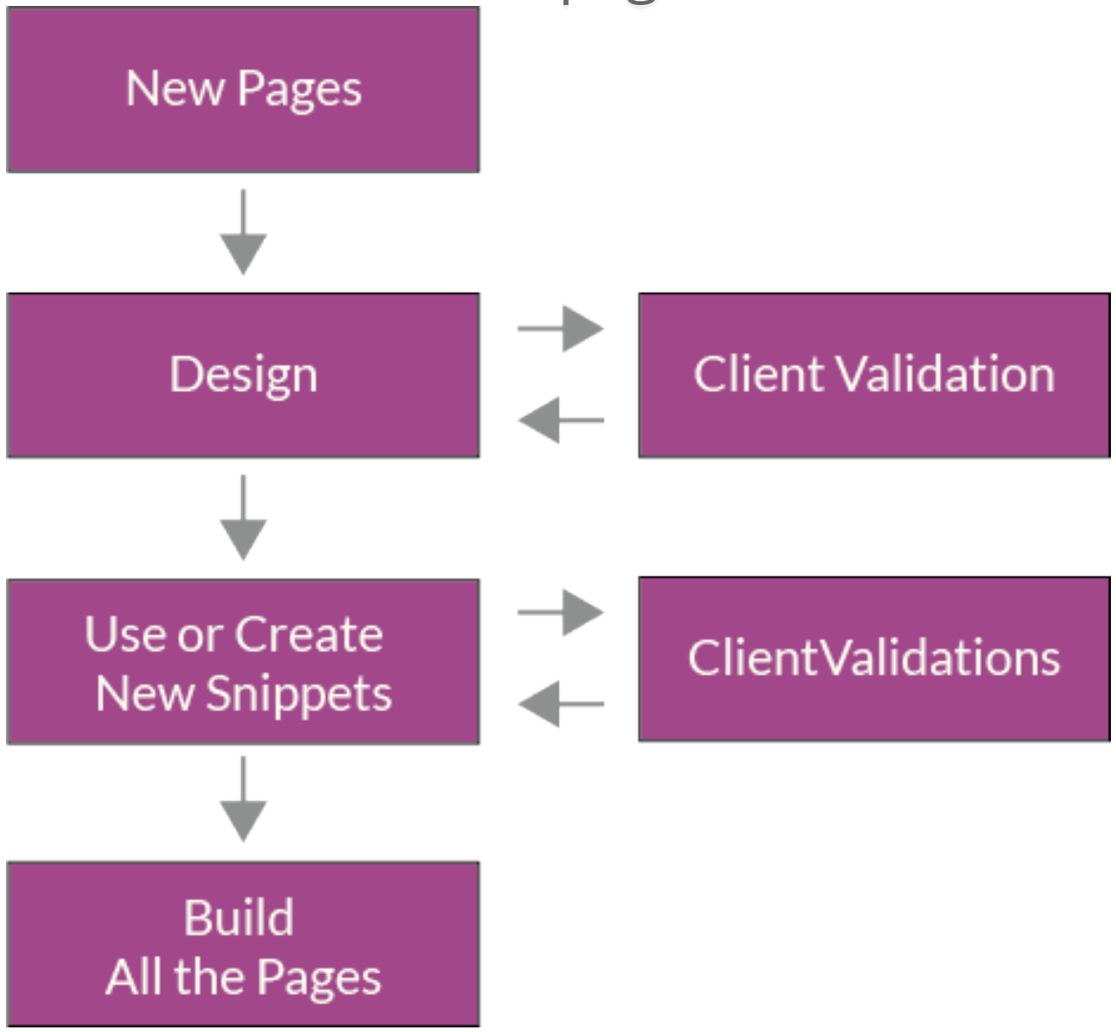
# Classical workflow

Add new features or pages.



# Working with Odoo CMS

Add new features or pages.



odoo

# Tutorial

# A Theme?

**A Theme is an Odoo's MODULE**

(Without Python Logic)

# Structure of a Theme

- My Theme
  - static
    - src
      - js, css, font, xml, img
  - views
    - my_theme.xml
    - snippets.xml
    - options.xml
    - pages.xml (static pages)
    - [...]

odoo

# Simple HTML page

# Starting with an HTML page

Let's start with the homepage.

*views/pages.xml*

```
[...]
  <template id="website.homepage" name="Homepage" page="True">
   <html>
    <head>
     <title>Title</title>
    </head>
    <body>
     <h1> Hello, World!
     </h1>
    </body>
   </html>
  </template>
[...]
```

# Starting with an HTML page

Add the Odoo context : ( with Bootstrap front-end framework, Edition bar, Snippets, etc. )

*views/pages.xml*

```
[...]
  <template id="website.homepage" name="Homepage" page="True">
   <t t-call="website.layout">
    <h1> Hello, World!</h1>
   </t>
  </template>
[...]
```

# Starting with an HTML page

It's possible to create all the pages like this way.

*views/pages.xml*

```
[...]>
  <template id="website.homepage" name="Homepage" page="True">
   <t t-call="website.layout">
   <div id="wrap" class="oe_structure">
     <!--  Your HTML code here -->
    </div>
   </t>
[...]
```

Adding the class "oe_structure" allows you to use this cool feature: **Snippets**.

odoo

# Snippets

# Build with snippets

But instead of creating all the pages this way, we think about using "Building Blocks".

We call them "**Snippets**".

- Block of html code usable everywhere.
- Draggable in your page.
- Can contain Javascript or/and Css logics.

# A very Simple Snippet

Structure of a snippet.

*views/snippets.xml*

```
[...]
  <template id="my_theme_snippets" inherit_id="website.snippets" name="My Theme snippets">
   <xpath expr="//div[@id='snippet_structure']" position="inside">
    <div>
     <!-- Thumbnail -->
     <div class="oe_snippet_thumbnail">
      <img class="oe_snippet_thumbnail_img"
           src="/my_theme/static/src/img/snippet/snippet_thumbs.png"/>
      <span class="oe_snippet_thumbnail_title">My Snippet</span>
     </div>
     <!-- Snippet Body -->
     <section class="oe_snippet_body mt_simple_snippet">
      <div class="container">
       <hr />
       <h1 class="text-center mb32 mt32">This is a simple snippet</h1>
       <hr />
      </div>
     </section>
    </div>
   </xpath>
  </template>
[...]
```

# Customize my Snippet

We can customize this simple snippet with Sass/Css.

*static/src/css/my_theme.sass*

```
// Pure compass imports
@import "compass/css3"
@import "bootstrap"

// Create CSS only for snippet
@import "my_theme-snippet.sass"
```

*static/src/css/my_theme-snippet.sass*

```
@font-face
  font-family: 'BebasNeue'
  src: url('/my_theme/static/src/font/BebasNeue Bold.ttf')
  src: local("BebasNeue Book"),
       url('/my_theme/static/src/font/BebasNeue Bold.ttf') format("truetype")

.mt_simple_snippet
  h1
    font-family: 'BebasNeue'
    font-size: 5em
```

# Customize my Snippet

To insert this new Css we need to extend the theme template and replace the default bootstrap by our new Css.

*views/my_theme.xml*

```
[...]
  <template id="theme" inherit_id="website.theme" name="My Theme Assets">
   <xpath expr="//link[@id='bootstrap_css']" position="replace">
    <link rel="stylesheet" type="text/css" href="/my_theme/static/src/css/my_theme.css" />
   </xpath>
  </template>
[...]
```

# Snippet & Javascript

It's possible to add **javascript logic** when a snippet has been dropped or appears in the page.

*static/src/js/snippet.js*

```
(function() {
  'use strict';
  var website = openerp.website;
  website.openerp_website = {};
  website.snippet.animationRegistry.my_snippet = website.snippet.Animation.extend({

    selector : ".mt_simple_snippet",

    start: function(){
      var h1 = this.$el.find("h1");
      var h1_width = h1.width();
      h1.css('width',0);
      h1.animate( { width: h1_width }, 3000 );
    },

  });
})();
```

# Snippet & Javascript

Just inherits from "website.assets_frontend" template to enable it.

*views/my_theme.xml*

```
[...]
  <template id="assets_frontend" inherit_id="website.assets_frontend"
            name="My Theme Front End Assets">
   <xpath expr="." position="inside">
    <script src="/my_theme/static/src/js/snippet.js"></script>
   </xpath>
  </template>
[...]
```

# Organize my snippets

You can create a new snippet section or insert your snippet into an already present section.

New section *views/snippet.xml*

```
[...]
  <template id="snippets" inherit_id="website.snippets" name="My Simple Snippet">
   <xpath expr="//ul[@class='nav navbar-nav nav-tabs']" position="inside">
    <li>
      <a href="#snippet_my_snippet" data-toggle="tab">My Snippet</a>
    </li>
   </xpath>
  </template>
[...]
```

Insert into "Structure" section.

```
 [...]
 <template id="my_theme_snippets" inherit_id="website.snippets" name="My Theme snippets">
   <xpath expr="//div[@id='snippet_structure']" position="inside">
    <div><!-- Your snippet --></div>
   </xpath>
 </template>
 [...]
```

odoo

# Options

# Add Options

We can add options for every snippets or blocks.

In our case, we add 2 options ( patterns background) for the snippet created before.

*views/options.xml*

```
[...]
  <template id="my_theme_snippet_option" name="My Snippet Options"
            inherit_id="website.snippet_options">
  <xpath expr="." position="inside">
    <div data-snippet-option-id="my_theme_snippet_option"
         data-selector=".mt_simple_snippet"
         data-selector-children=".oe_structure">
    <li class="dropdown-submenu">
            <a tabindex="-1" href="#">Pattern</a>
            <ul class="dropdown-menu">
                <li data-value="tweed"><a>Tweed</a></li>
                <li data-value="sprinkles"><a>Sprinkles</a></li>
            </ul>
        </li>
    </div>
    </xpath>
[...]
```

# Add Options

In fact, it adds a class-name to the data-selector.

And now, simply create the Css to have the desired result.

*static/src/css/my_theme-options.sass*

```
.mt_simple_snippet
  &.tweed
    background-image: url(/my_theme/static/src/img/backgrounds/patterns/tweed2.png)
    h1
      color: rgb(255,255,255)
    hr
      border-top: 1px solid dashed rgba(255,255,255,.8)
  &.sprinkles
    background-image: url(/my_theme/static/src/img/backgrounds/patterns/sprinkles.png)
    h1
      color: rgb(120,120,120)
      +text-shadow(0 0 5px rgba(0,0,0,.3))
    hr
      border-top: 1px solid solid rgba(0,0,0,.8)
```

# Custom Css

We can override Bootstrap variables to create your theme.

*static/src/css/my_theme.sass*

```
// Override Bootstrap variables
$brand-primary:         rgb(120,120,120)
$brand-success:         rgb(94,148,162)

// Pure compass imports
@import "compass/css3"
@import "bootstrap"

// Create CSS only for snippet
@import "my_theme-snippet.sass"

// Create CSS only for options
@import "my_theme-options.sass"
```

odoo

# Summary

# Summary

- Infinite customizations
- Easy to understand
- Template inherits
- Bootstrap based
- Only imagination is your limit
- Robust Odoo back-end behind

**... and so much things will come :)**

odoo

# Example

odoo

# Thank you

And we are hiring a webdesigner. Contact cde@odoo.com for more informations.