

Templates & layouts

PrestaShop template files are based on the [Smarty 3 template engine](#).

All template files must be stored in the theme's `templates/` subfolder. For instance, the default theme has its template files in the following folder: `/themes/classic/templates`.

Directory structure

Templates are then split between various subfolders.

Folder	Description
<code>_partials</code>	Code shared across the whole site like, header, footer or notifications.
<code>catalog</code>	Product page, product/brand/supplier listing, search results and such.
<code>checkout</code>	Cart, delivery options, payment options, order confirmations and such.
<code>cms</code>	All the static content: contact, sitemap, CMS pages and such.
<code>customer</code>	Everything about the customer's account and its data.
<code>errors</code>	All the error templates: not found, server error, forbidden and such.
<code>layouts</code>	The theme layouts: 1, 2 or more columns, full width, everything is possible.

Template files should be written so that a single `.tpl` can generate a whole HTML page – unless they are inside a `_partials` folder or subfolder (see our coding standard, linked from the Prologue chapter of this documentation).

Templates

We make a **clear difference between templates and layout**:

- A template extends a layout.
- The layout holds the global organization of the page.
- A template is specific to a feature: the product page for example.

There are many templates in a PrestaShop theme. The main ones include:

- `index.tpl` for the home page.
- `catalog/product.tpl` for the product page.
- `catalog/listing/product-list.tpl` for any product list page.
- `checkout/cart.tpl` for the detailed cart.
- `checkout/checkout.tpl` for the checkout process.

Specific templates

If you're working on a big store with many languages, you may need to change the layout of the page depending on the language.

For example, you may want a different product page for American customers and Japanese ones. In this case you have to create new `product.tpl` template and place it in the right folder.

When searching for a template, PrestaShop will check many locations to determine which file should be used. It makes it easy to have a different template for a given locale or a specific entity id.

With the Product page, the core will check the following locations (in order) and return the first template found:

- Example with a product with ID = 3 and locale = en-US
 - en-US/catalog/product-3.tpl
 - catalog/product-3.tpl
 - en/US/catalog/listing/product.tpl
 - catalog/listing/product.tpl
- Another example with category template for the category with ID = 9 and locale = en-US.
 - en-US/catalog/listing/category-9.tpl
 - catalog/listing/category-9.tpl
 - en/US/catalog/listing/category.tpl
 - catalog/listing/category.tpl
 - en-US/catalog/listing/product-list.tpl
 - catalog/listing/product-list.tpl

This feature is mostly made for developers working on a custom template for a customer.

Layouts

The layout is the organization of the page, the way in which the parts of your design are arranged. The typical example is the sidebar: is there a sidebar on your category page or is your product listing taking the whole space?

With PrestaShop 1.7, users are given the ability to change the layout of each page independently. As a template developer, it is your role to ensure that your theme is compatible.

The screenshot shows the PrestaShop 1.7 administration interface for Theme & Logo configuration. The 'CHOOSE LAYOUTS' section contains a table with the following data:

Page	Description	Layout
404 error	This page cannot be found	Full Width - No side columns, ideal for distraction-free pages such as product pages.
Best sales	Our best sales	Two Columns, small left column - Two columns with a small left column
Contact us	Use our form to contact us	Two Columns, small left column - Two columns with a small left column
index	Shop powered by PrestaShop	<input checked="" type="checkbox"/> Full Width - No side columns, ideal for distraction-free pages such as product pages. <input type="checkbox"/> Three Columns - One large central column and 2 side columns. <input type="checkbox"/> Two Columns, small left column - Two columns with a small left column <input type="checkbox"/> Two Columns, small right column - Two columns with a small right column <input type="checkbox"/> Two Columns, small left column - Two columns with a small left column
Manufacturers	Manufacturers list	Two Columns, small left column - Two columns with a small left column
New products	Our new products	Two Columns, small left column - Two columns with a small left column
Forgot your password	Enter the e-mail address you use to sign in to receive an e-mail with a new password	Full Width - No side columns, ideal for distraction-free pages such as product pages.
Prices drop	Our special products	Two Columns, small left column - Two columns with a small left column
Sitemap	Lost? Find what you are looking for	Full Width - No side columns, ideal for distraction-free pages such as product pages.
Suppliers	Suppliers list	Full Width - No side columns, ideal for distraction-free pages such as product pages.

What's in a layout file

The layout is the very top level of the template inheritance tree. Basically, it holds the opening and closing `<html>` tags.

Typical layout files look like the following snippet. This one is a full file.

```

<!doctype html>
<html lang="{language.iso_code}">

<head>
  {block name='head'}
  {include file='_partials/head.tpl'}
  {/block}
</head>

<body id="{page.page_name}" class="{page.body_classes|classnames}">

  {hook h='displayAfterBodyOpeningTag'}

  <main>

    <header id="header">
      {block name='header'}
      {include file='_partials/header.tpl'}
      {/block}
    </header>

    <section id="wrapper">
      <div class="container">

        {block name='breadcrumb'}
        {include file='_partials/breadcrumb.tpl'}
        {/block}

        {block name="left_column"}
        <div id="left-column">
          {if $page.page_name == 'product'}
            {hook h='displayLeftColumnProduct'}
          {else}
            {hook h="displayLeftColumn"}
          {/if}
        </div>
        {/block}

        {block name="content_wrapper"}
        <div id="content-wrapper">
          {block name="content"}
          <p>Hello world! This is HTML5 Boilerplate.</p>
          {/block}
        </div>
        {/block}

      </div>
    </section>

    <footer id="footer">
      {block name="footer"}
      {include file="_partials/footer.tpl"}
      {/block}
    </footer>

  </main>

  {hook h='displayBeforeBodyClosingTag'}

  {block name='javascript_bottom'}
  {include file="_partials/javascript.tpl" javascript=$javascript.bottom}
  {/block}

</body>

</html>

```

From there, each part of the theme will do its job and replace content inside these bricks, keeping the same organization.

 Remember to define as many blocks as possible.