

Inheritance allows you to maintain a consistent design throughout the website and manage content shared by multiple pages in a single location. Portal engine pages can use two different types of content inheritance:

- Page nesting inside ancestor pages
- Inheriting the entire page template from the parent page (shared templates)

Using page nesting

Nested pages display their own content inside other pages. You can nest pages within ancestor pages in the content tree.



What are ancestor pages?

Ancestors include all pages under which a given page is stored, from the root of the site's content tree down to the page's direct parent.

For example, the */Company/Offices/London Office* page has the following ancestors:

- Website root page
- /Company
- /Company/Offices

Nesting allows you to organize your website's content tree in the following way:

- Pages that provide shared content on the upper levels
- Individual content pages stored as subpages

Creating pages that support nesting

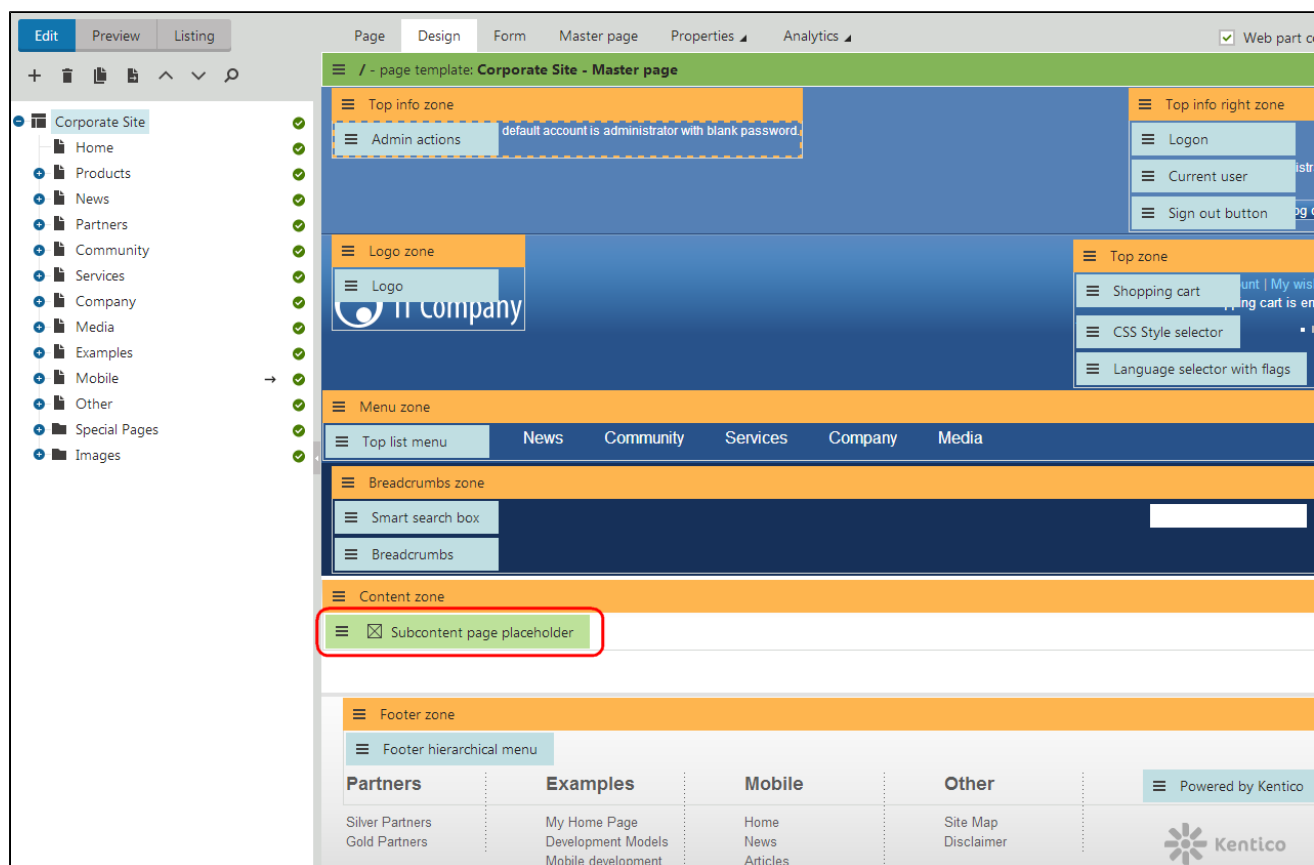


Note: The following steps are required for all pages serving as [master pages](#), but you can also set up nesting for any other pages.

To allow subpages to nest within a page:

1. Open the **Pages** application.
2. Select the page in the content tree.
3. Open the **Design** tab.
4. [Add](#) the **Page placeholder** web part.

The page placeholder specifies the position of nested pages within the content. When displaying the nested pages, the system loads everything around the page placeholder as fixed content.




Setting default content for page placeholders


By default, the placeholder does not display anything on the page where it is placed. You can configure the placeholder to show one of the website's [re-usable page templates](#) as default content:

1. **Configure** the page placeholder web part (double-click).
2. **Select** a template in the **Default page template** property.
3. Click **OK**.

The page displays the selected template inside the area occupied by the page placeholder. Nested pages ignore the placeholder's default template and display their own content instead.

 The **Use template on all subpages** and **Page to display** properties force the placeholder to always display the default template or a specific page. Such placeholders cannot be used for standard page nesting — all nested subpages display the specified content instead of their own.

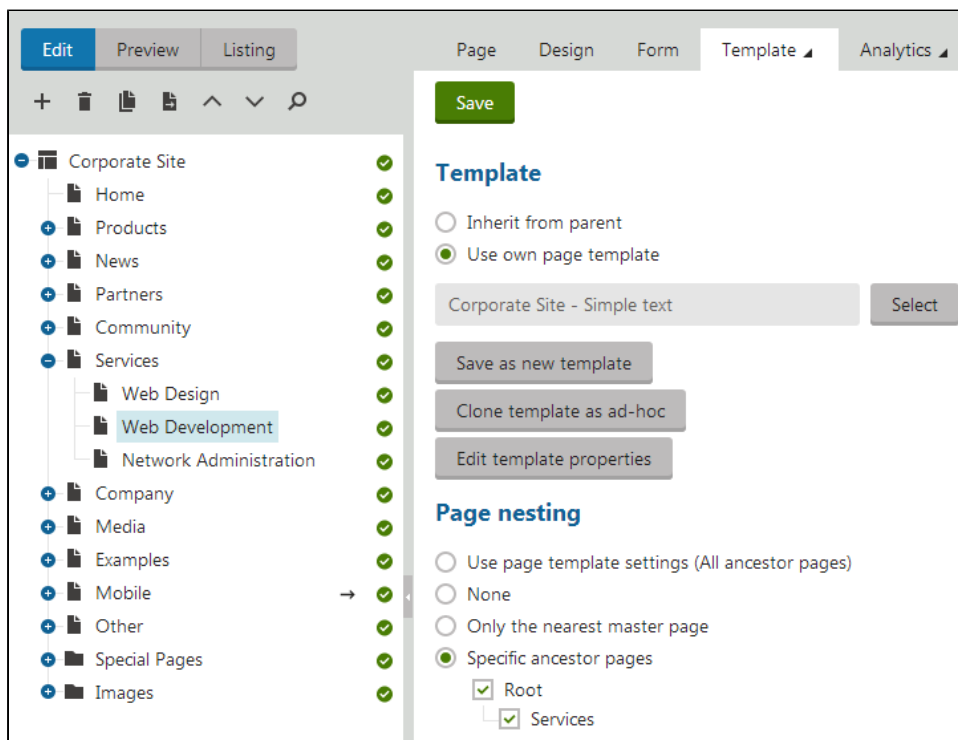
Configuring pages to nest within ancestors

 **Tip:** You can use page nesting on any number of levels. Nested pages can also contain their own page placeholders for displaying subpages.

1. Open the **Pages** application.
2. Select the subpage page in the content tree.
3. Open the **Properties -> Template** tab.
4. Choose one of the following **Page nesting** options for the page:

Page nesting type	Description
-------------------	-------------

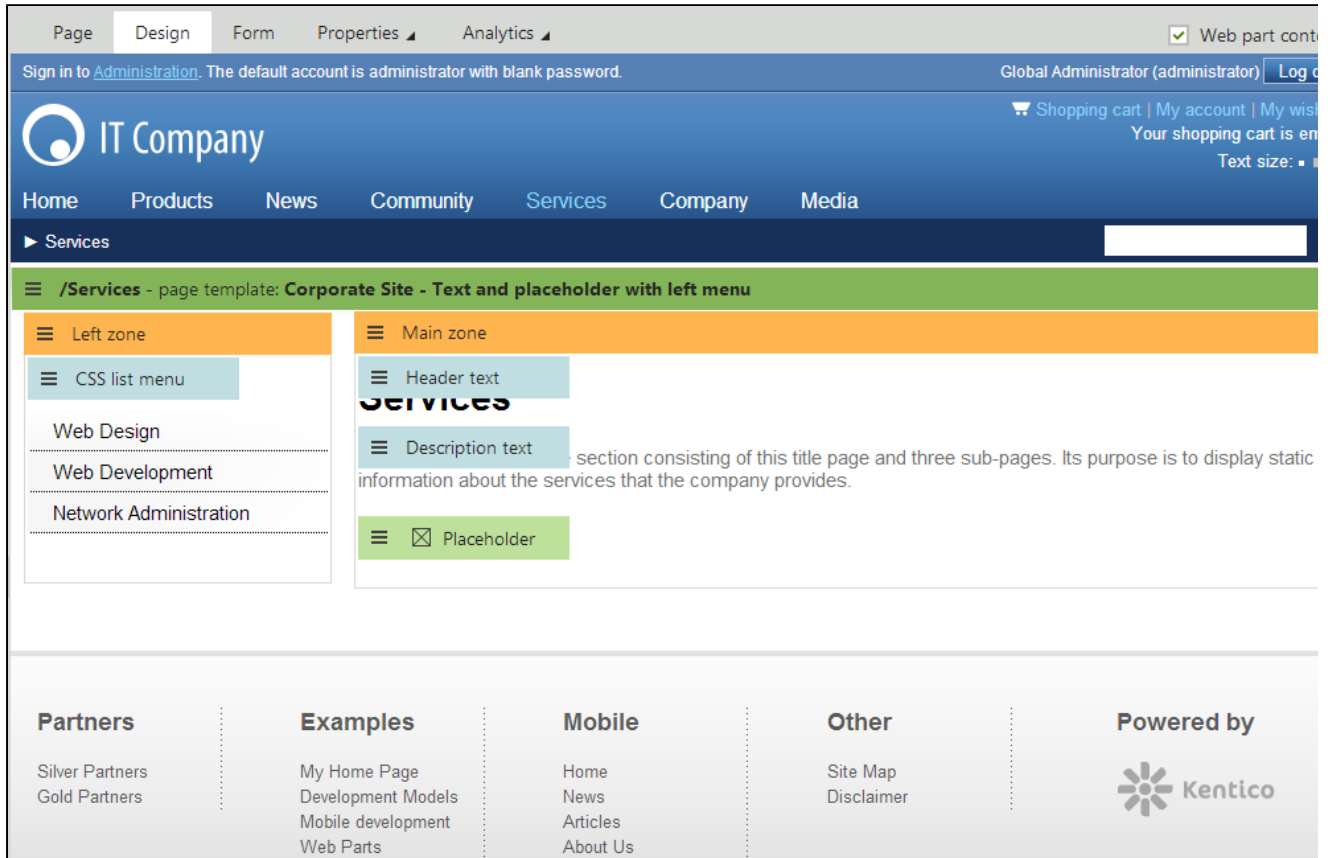
Use page template settings	<p>The page nesting settings are determined by the configuration of the page's template. This option allows you to manage the settings for pages with shared templates.</p> <p>To modify the page template's nesting settings:</p> <ol style="list-style-type: none"> Click Edit template properties. Set the Page nesting options on the General tab. Click Save.
None	The page behaves as a standalone page without any nesting.
Only the nearest master page	The page nests only within the website's master page . If your website uses multiple master pages, the page nests within the closest master page in the content tree hierarchy.
Specific ancestor pages	Allows you to enable or disable nesting within any ancestor pages (regardless of the master page structure). Specific pages are represented by the checkboxes below.



5. Click **Save**.

When visitors view the nested page, the system loads the content of the selected ancestors and displays it around the content of the page itself (according to the positions of the page placeholder web parts).

On the **Design** tab of nested pages, the editable template section appears inside the content of ancestor pages (in the position of the page placeholder on the previous nesting level). You can edit the [page layout](#) and [web parts](#) of the nested page's template as usual. You cannot modify the content from the ancestors where the page is nested.



The screenshot shows the Kentico Design tab for a corporate site template. The top navigation bar includes links for Home, Products, News, Community, Services, Company, and Media. The main content area is titled '/Services - page template: Corporate Site - Text and placeholder with left menu'. It features a left zone with a CSS list menu containing 'Web Design', 'Web Development', and 'Network Administration'. The main zone contains a header text area with the word 'SERVICES', a description text area with a paragraph about static information, and a placeholder area. The footer includes sections for Partners, Examples, Mobile, Other, and Powered by Kentico.



Important

You can only edit the page template on the **Design** tab if all pages where the given page is nested contain a *Page placeholder*.

If you configure a subpage to "nest" inside a page without a page placeholder, the subpage only displays the content of the given ancestor and cannot be modified in any way.

Inheriting page header content for nested pages

You can add custom `<head>` content to pages, such as links to external CSS or JavaScript files.

1. In the **Pages** application, open the page's **Properties -> Template** tab.
2. Click **Edit template properties**.
3. Switch to the **Header** tab.
4. Type in the required content.
5. Click **Save**.

The system inserts the content into the `<head>` element in the output code of all pages that use the given template.



Note: Custom `<head>` content is only rendered on pages whose page type has the **Behaves as Page (menu item) type** property enabled.

Nested pages can inherit the header content from their ancestors. The inheritance depends on the following options, which you can enable or disable on the **Header** tab of each page template:

- **Allow descendant templates to inherit the header**
- **Inherit headers from the templates of ancestor pages**

The header inheritance follows the page nesting settings – pages can only inherit head content from the ancestor pages where they are nested. Pages that do not use nesting cannot inherit head content.

Inheriting the page template of the parent page

One way to display content from parent pages is to inherit the entire [page template](#). Pages with an inherited template are mostly identical to the parent, but you can modify them in the following ways:

- Set different content inside the page's editable web parts (**Editable text** and **Editable image**)
- Use web parts that display different content based on the page's type and location in the content tree (path)
- Hide web parts on subpages
- Add web parts that appear only on particular page types

A typical scenario where you can use page template inheritance is a parent page of the Page (menu item) type with multiple child pages that aren't set to behave as Page (menu item) types. By inheriting the template, users can create the child pages without worrying about the page design. You can use one of the page viewer web parts to dynamically display a list of these child pages on the parent page. Each child page then displays specific detailed information when viewed.

You can create [new pages](#) with an inherited template by choosing the **Use parent page template** option in the template selection dialog. Page types that aren't set to behave as Page (menu item) types automatically inherit the template of the parent page by default.

To configure an existing page to inherit the parent page's template:

1. In the **Pages** application, select the page in the content tree.
2. Open the **Properties -> Template** tab.
3. Select **Inherit from parent** in the **Template** section.
4. Click **Save**.

The child page uses the same page template as the parent. **Any changes that you make** to the page template's [layout](#) or [web part configuration](#) **affect both the parent page and all pages that inherit the template.**

✓ **Tip:** The page template inheritance can continue throughout the content tree — pages may inherit from parent pages that already use an inherited template (*Parent -> Child -> Descendants...*).

Hiding web parts on subpages

You can disable content inheritance for individual instances of web parts. This allows you to add web parts to pages without affecting the design of subpages that inherit content. Applies to both types of content inheritance — [page nesting](#) and pages with [inherited templates](#).

1. In the **Pages** application, open the **Design** tab of the page containing the web part.
2. **Configure** the web part (double-click).
3. Expand the **Visibility** property category.
4. Select the **Hide on subpages** checkbox.
5. Click **Save & Close**.

Users now cannot see the web part on sub-pages in the content tree.

✓ Example

You can find an example on the **News** page on the sample Corporate site. The page hides the *Header text*, *Description text*, *News filter* and *Universal pager* web parts on child pages. As a result, the text sections are only visible on the list of news items — not on the detail pages of individual news items, which inherit the page template from the parent News page.

Making web parts visible only for specific page types

You can configure web part instances to appear only on specific [types of pages](#). This allows you to set up different content inheritance for different page types (applies to both [page nesting](#) and pages with [inherited templates](#)). You can also use this feature for pages that simply share page templates, without any kind of inheritance.

1. In the **Pages** application, open the **Design** tab of the page containing the web part.
2. **Configure** the web part (double-click).
3. Expand the **Visibility** property category.
4. Click **Select** next to the **Show for page types** property.
5. Choose the allowed page types and click **Select**.
6. Click **Save & Close** in both dialogs.

The web part is now only visible when viewing pages of the selected types.



Example

You can find an example on the **/Community/Blogs/Andrew-Jones-Blog** page on the sample Corporate site. The page displays the *Header text*, *Description text* and *Blogs filter* web parts only on *CMS.Blog* and *CMS.BlogMonth* pages. The web parts are hidden when viewing individual *CMS.BlogPost* pages, which inherit the template.

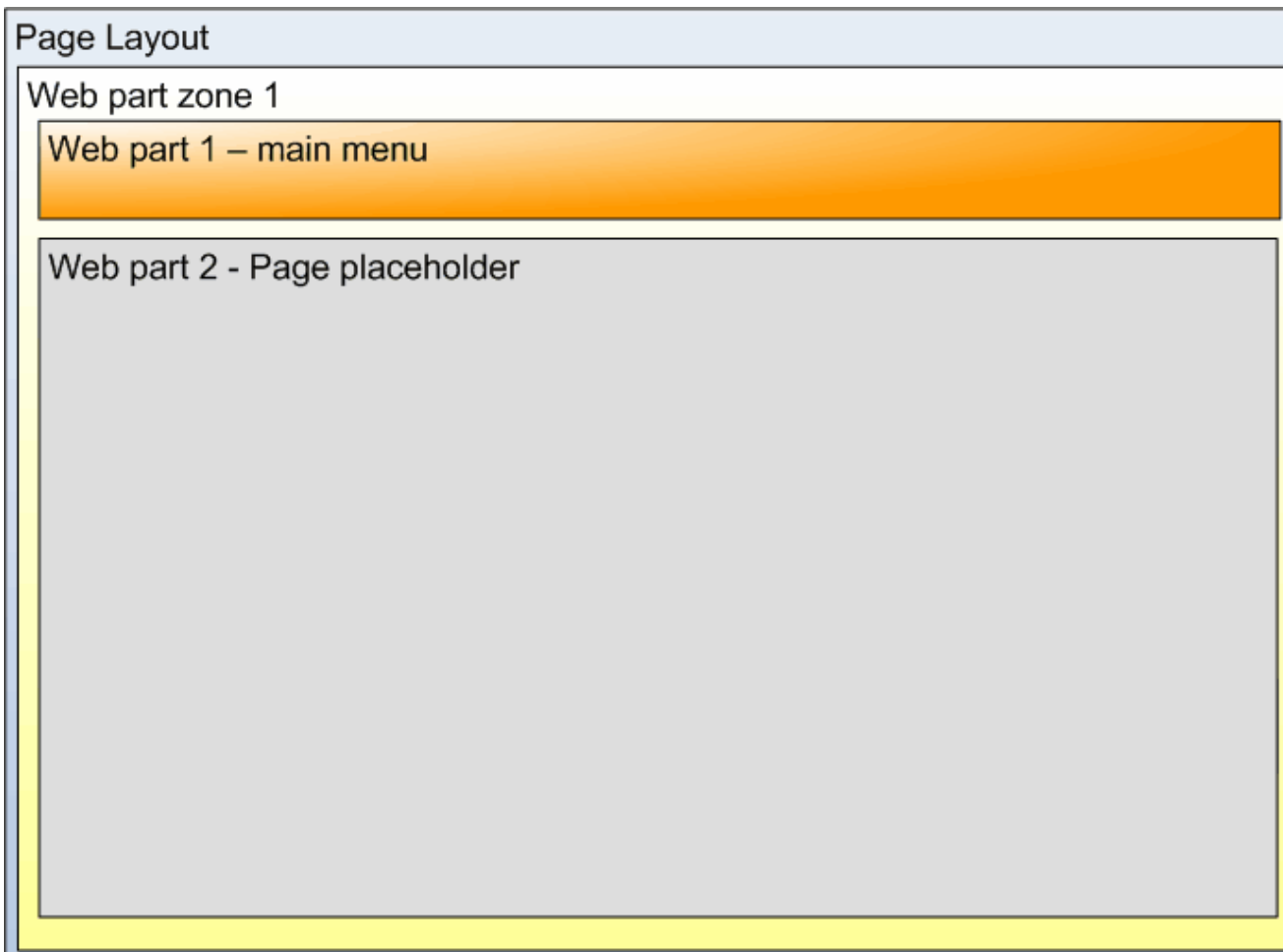
Sample inheritance scenario

The following example shows how the system processes a request for a nested product page: **/Products/Notebooks/Dell-XPS-15z**

This sample scenario demonstrates content inheritance through both page nesting (*Page placeholder* web parts) and page template sharing. The portal engine loads the templates in the following order:

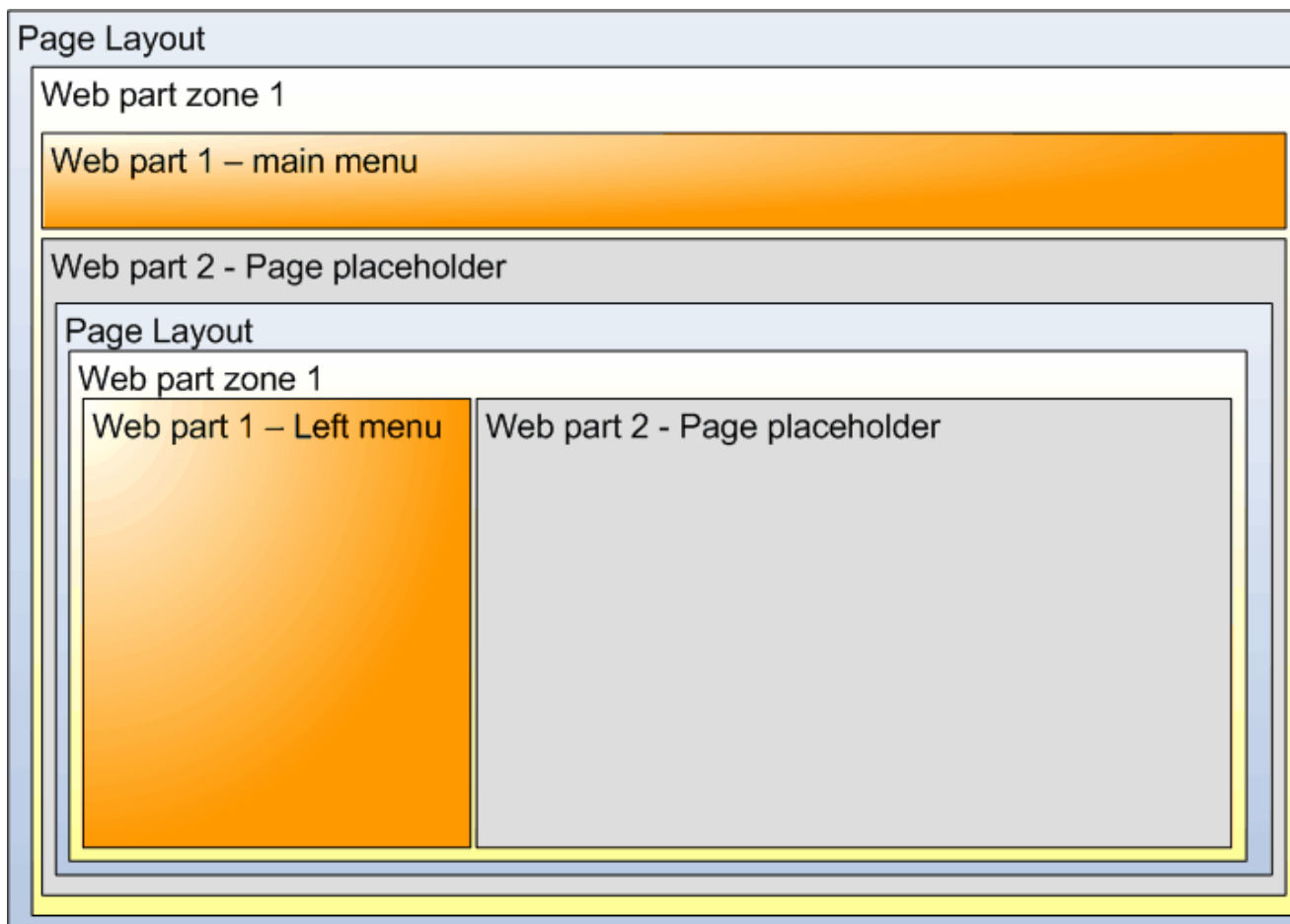
1. / (root)

Website master template



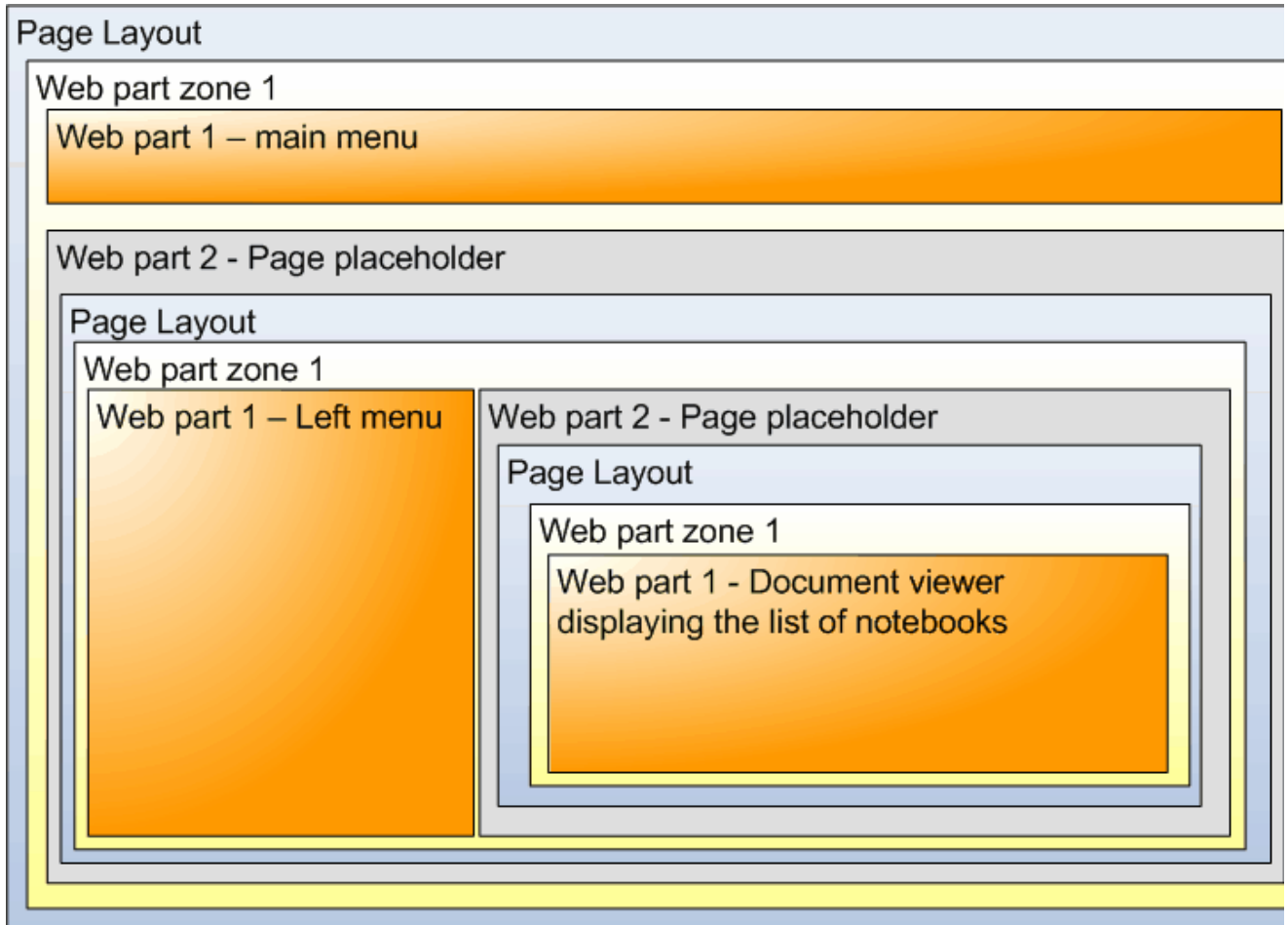
2. /Products

Products page template



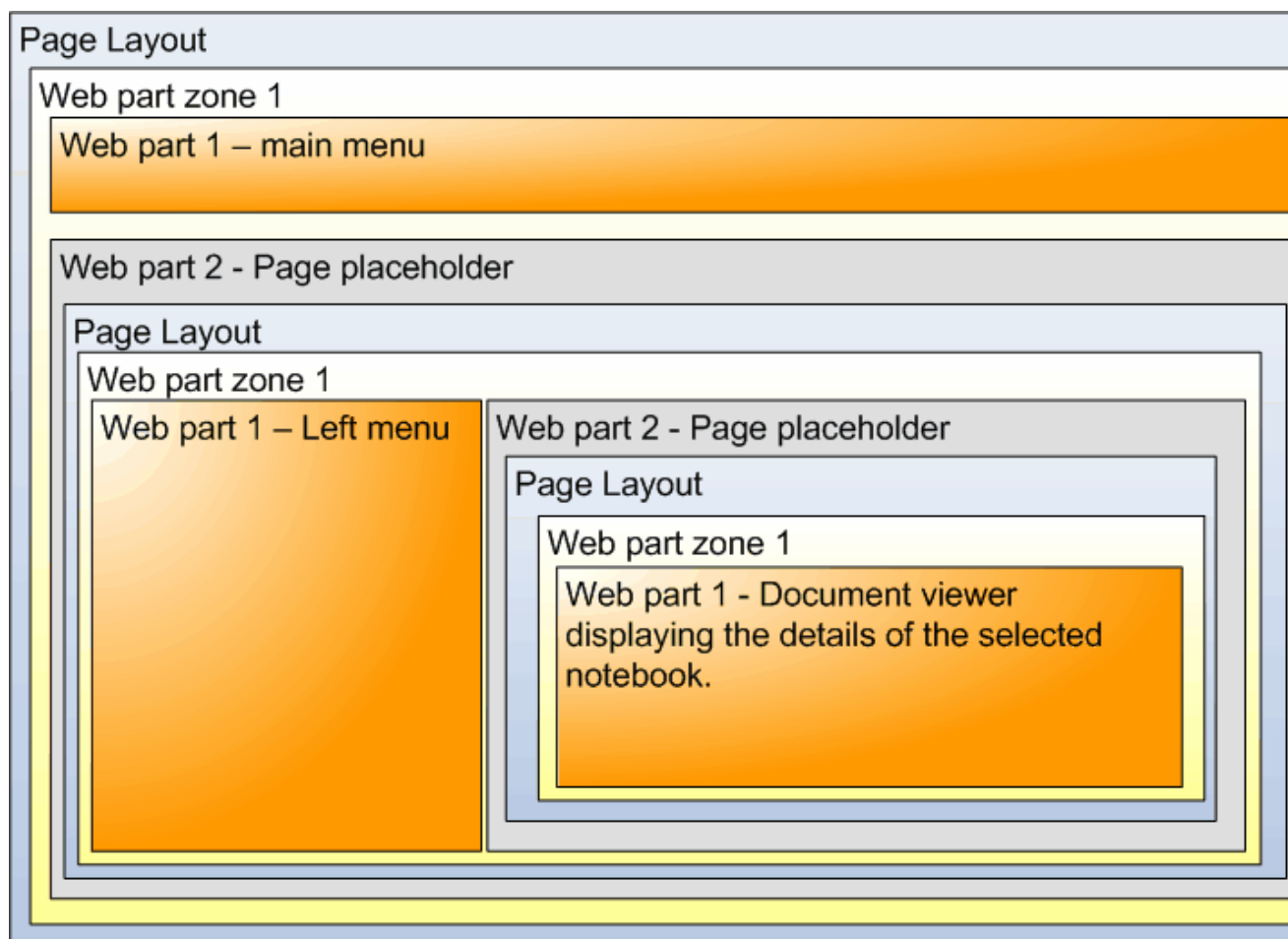
3. /Products/Notebooks

Laptops page template



4. /Products/Notebooks/Dell-XPS-15z

Laptops page template - inherits the page template from the parent page.



Note

A page viewer web part (e.g. **Repeater**) automatically displays the detail of the selected product instead of the product list.

See [Loading and displaying data on websites](#) for more information.