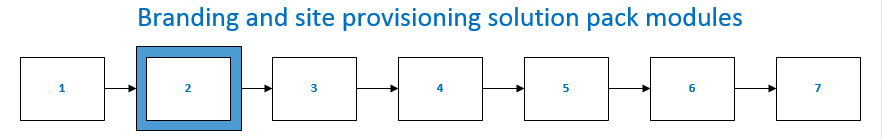
Module 2: Inside SharePoint pages

# About SharePoint pages and the SharePoint page model

This document introduces the SharePoint 2013 and SharePoint Online page model. The SharePoint page model describes the files and logical relationships that SharePoint uses to render SharePoint pages in the web browser.



This module:

* **Defines** key terms and concepts related to the page model.
* Provides an **overview of the pieces of the page model**, including master pages, content pages, and the regions of a SharePoint page.
* Inventories **page files** that are available out-of-the-box in SharePoint 2013 and SharePoint Online, including master pages, Web Part pages, and wiki pages.
* Describes how SharePoint **renders a page**.

# Key terms and concepts

Before you learn about the details of the SharePoint page model, it’s helpful to get acquainted with a few terms and concepts.

**Table 1. Key terms and concepts to know when working with the SharePoint page model**

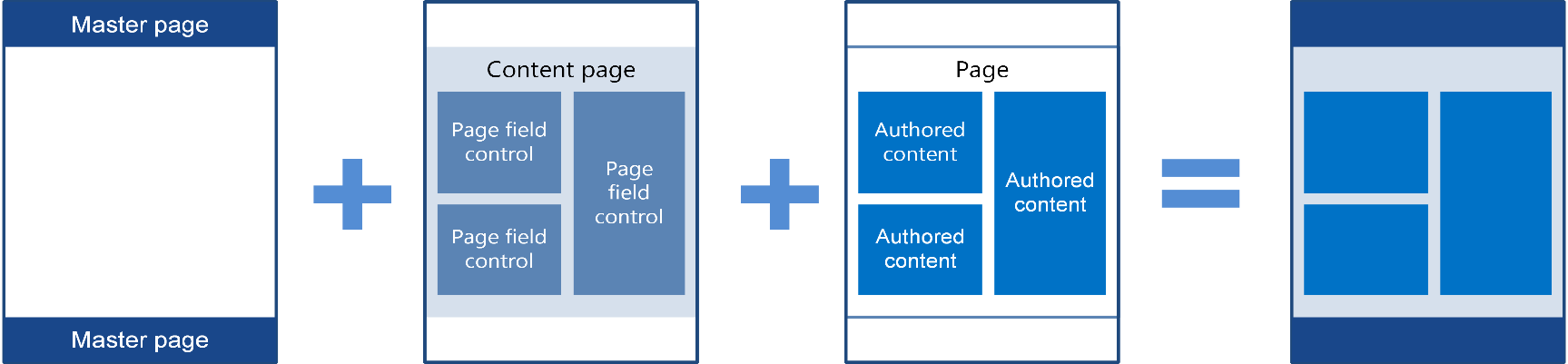
|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Definition** | **Where is it? How can I access it?** | **Guidance** |
| Collaboration site | See Team site. |  |  |
| Content placeholder | An entry in a master page that reserves a space for controls or content that can be programmatically replaced later. | All SharePoint .master pages | Content placeholders are the building blocks of SharePoint .master pages. |
| Master page | A page that standardizes the behavior and presentation the left navigation and top navigation regions of a SharePoint page. | SharePoint file system  Master Page Gallery |  |
| [Master Page Gallery](http://msdn.microsoft.com/en-us/library/dn205271.aspx) | The Master Page Gallery is a special document library in SharePoint 2013 where all branding elements—master pages, page layouts, JavaScript files, CSS, and images—are stored by default. Every site has its own Master Page Gallery. | Settings > Site Settings > Master Pages and Page Layouts > | The Master Page Gallery contains catalogs that store branding assets such as master pages and CSS files.  **TIP** When creating custom branding elements, store custom assets in the default Master Page Gallery file structure. |
| [MDS/Minimal Download Strategy](http://msdn.microsoft.com/en-us/library/dn456544.aspx) | Reduces the amount of data that the browser must download when users navigate from one SharePoint page to another. | Site Settings | When MDS is active, SharePoint passes all page requests through /\_layouts/15/start.aspx and checks for visual differences between new page requests and the previously loaded page.  [Optimize page performance in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/dn449110.aspx) |
| Navigation | Functionality that enables users to move around the information architecture of a SharePoint site. Navigation elements in SharePoint include search, tree controls, buttons, the ribbon, hyperlinks, tabs, menus, and taxonomy. |  | [Navigation](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.navigation.aspx) class  [NavigationNode](http://msdn.microsoft.com/EN-US/library/office/microsoft.sharepoint.client.navigationnode.aspx) class |
| Oslo.master | A default master page in SharePoint 2013. | SharePoint file system  Master Page Gallery | Unlike the seattle.master master page, the current navigation is in the same position as the top navigation area. |
| Page content control | A control on a publishing site where a Web Part can be added. |  |  |
| Page Layout | A template applied to a Publishing page that enforces the consistent presentation of content. | SharePoint file system  Master Page Gallery | [How to: Create a page layout in SharePoint 2013](http://msdn.microsoft.com/en-us/library/jj822368.aspx) |
| [Page model](http://msdn.microsoft.com/en-us/library/jj191506.aspx) | The files, content, and interactions that result in a SharePoint page rendered to users in a browser. |  |  |
| Publishing page | An .aspx page in a Publishing site |  | [PublishingPage](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.publishing.publishingpage.aspx) class |
| Publishing site | A SharePoint site that can access publishing sites and pages, which include page layouts, taxonomy, managed navigation, and other web content management and enterprise content management features. |  | [PublishingWeb](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.publishing.publishingweb.aspx) class  [What’s new in SharePoint 2013 sites?](http://msdn.microsoft.com/en-us/library/office/jj163942.aspx) |
| Seattle.master | A default master page in SharePoint 2013. | SharePoint file system  Master Page Gallery |  |
| Team site | A site designed for users to collaborate on documents, wikis, ideas, processes, and so on. |  |  |
| Text layout | Defines the content areas that appear on a Wiki page. |  |  |
| Text layout control | A wiki page control that can contain text, images, Web Parts, and App Parts. |  |  |
| Top-level site | The default, top-level site provided by the server. |  | [Create a SharePoint site](http://office.microsoft.com/en-us/sharepoint-designer-help/create-a-sharepoint-site-HA010131447.aspx) |
| Web Part | Server-side controls that run inside the context of site pages. |  | [Custom actions and property bag entries from a SharePoint app](http://blogs.msdn.com/b/vesku/archive/2013/10/02/ftc-to-cam-custom-actions-and-property-bag-entries.aspx) (Vesa Juvonen) |
| Web Part page | A content page made up of Web part zones, which can contain Web Parts. Web Parts are represented on Web Part pages by [WebPartDefinition](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.webparts.webpartdefinition.aspx) objects. |  | [Microsoft.SharePoint.Client.WebParts](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.webparts.aspx) namespace |
| Web Part zone | A region on a page where a Web Part can be added. |  |  |
| Wiki page | A content page that uses the Enterprise Wiki site template. |  | See the [ProvisionWikiPages](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-5db977e8) sample to learn how to provision a wiki page. |

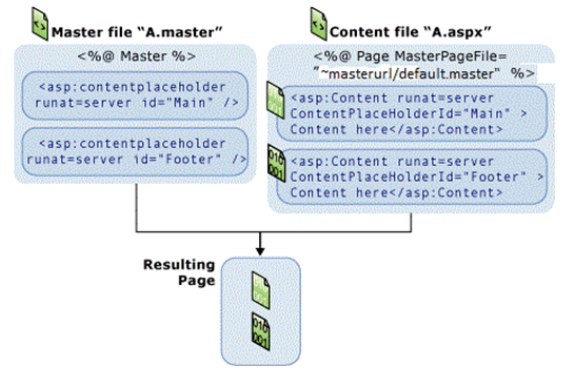
# Page model overview

A rendered SharePoint page is the combination of a SharePoint master page, a content page, and authored content.

Figure 1 shows how the files that SharePoint uses to make a page and the content that users add to a page are combined into the SharePoint page that users see in their web browser.

**Figure 1. Master page + content page + authored content on a page = rendered page**





In Figure 1, the master page lays out the content defined in the content page, and populates the page returned to the user in the browser with authored content.

## SharePoint master pages

Master pages provide the common layout and interface of SharePoint pages. A master page defines the chrome—the elements of a page that consistently appear as users navigate through a site.

A master page is an ASP.NET file with a .master extension. It includes a **“<%@ Master”** directive, and defines the top-level HTML elements such as HTML, Head, and Form. It firsts lists controls and assemblies, and then declares a Document Type Definition of DOCTYPE, which tells the browser how to render the HTML. SharePoint 2013 is tuned to work best with the XHTML 1.0 and HTML5 DOCTYPES.

### Out-of-the-box master pages

SharePoint includes several .master pages by default. For the SKUs and site types where they apply, these master pages provide the default structure and chrome of a SharePoint page—specifically, on the top and left sides of the page.

Table 2 lists default SharePoint 2013 and SharePoint Online master pages, how and where they’re available.

**Table 2. Out-of-the-box master pages**

|  |  |
| --- | --- |
| **Master page** | **How and where it’s used** |
| Custom.master | System pages, such as forms and views. Used by all SharePoint 2013 and SharePoint Online SKUs. |
| Default.master | Site pages in publishing sites. Included in all SharePoint 2013 and SharePoint Online SKUs. Available when the publishing feature is activated. |
| Application.master | Some system pages, such as scope.aspx and keyword.aspx. Included in all SharePoint 2013 and SharePoint Online SKUs. |
| Minimal.master | Available default master page option in all SharePoint 2013 SKUs. |
| Seattle.master | Available default master page option in all SharePoint 2013 and SharePoint Online SKUs. |
| Oslo.master | Available default master page option in all SharePoint 2013 and SharePoint Online SKUs. |
| Kyoto.master | A master page available in SharePoint Online. |
| Berlin.master | A master page available in SharePoint Online. |
| Lyon.master | A master page available in SharePoint Online. |
| Mysite15.master | OneDrive for Business sites (previously: MySites, personal sites, or OneDrive Pro sites). |

Each default SharePoint master page includes controls that are required for common web programming technologies such as HTML, CSS, and JavaScript, to function in SharePoint.

### Content placeholders

Content placeholders hold the place for information defined in content pages. Content placeholders correspond to regions of a page.

Each region of a .master page is defined by between a few and hundreds of content placeholders.

SharePoint master pages use a mix of ASP.NET (“<asp:”) and SharePoint (“<SharePoint:”) declarations. The text after the colon in a declaration defines the control’s functionality; for example, **SharePoint:PlaceholderGlobalNavigation** injects the global navigation of a SharePoint page into the relevant HTML tags on that page.

Content controls in a master page bind content placeholders to content with the **ContentPlaceHolderID**.

* + - 1. System master pages and site master pages

SharePoint includes two types of master pages: *system master pages* and *site master pages*.

* System master pages are applied to all form pages and view pages on a site.
* Site master pages are used by all pages in a Publishing site.

You can tell which kind of master page a site is using by opening the .master page and viewing the **Page** directive, as follows:

* *~masterurl/default.master = system master page*
* *~masterurl/custom.master=site master page*

You can use CSOM code to set master page properties—mainly by writing code against the [Web](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.web.aspx) object. Change the system master page using its [MasterUrl](http://msdn.microsoft.com/EN-US/library/office/microsoft.sharepoint.client.web.masterurl.aspx) property, and change the site master page by using the object’s [CustomMasterUrl](http://msdn.microsoft.com/EN-US/library/office/microsoft.sharepoint.client.web.custommasterurl.aspx) property.

SharePoint Team sites and OneDrive for Business sites use system master pages (default.master), while SharePoint Publishing sites and SharePoint Team sites with the Publishing feature enabled point to the site master page in the **Page** directive.

* + - 1. Dynamic tokens and the SharePoint page URL

Content placeholders often include dynamic tokens, which are important pieces of code that form part of a SharePoint page URL. SharePoint parses URL strings according to the rules of protocols, such as HTTP, that define how hypertext information is transferred between the server and a SharePoint page. Usually, a content placeholder that points to a CSS or theme control will use a relative URL, which in the SharePoint server-side object model is represented as *~SPUrl*.

SharePoint uses dynamic tokens to bind the master page to the content page, which is defined in <asp:content> declaration of .master page code. Table 3 lists dynamic tokens that are found in SharePoint master pages, and either the CSOM properties that replace them when the page is processed, or the form of the URL string that SharePoint renders for that content placeholder.

**Table 3. Dynamic tokens in master pages replaced by property values**

|  |  |
| --- | --- |
| **Dynamic token** | **Replaced with** |
| ~masterurl/default.master | SPWeb.MasterUrl |
| ~masterurl/custom.master | SPWeb.CustomMasterUrl |
| ~site/<xyz>.master | http://<siteColl>/<subsite1>/<subsite2>/<xyz>.master |
| ~sitecollection/<abc>.master | http://<siteColl>/<abc>.master |

**Note**  The dynamic tokens in content placeholders correspond to server-side API properties and methods. When using remote provisioning, write code in CSOM or REST.

To learn more about dynamic tokens and SharePoint URLs, see [URLs and tokens in SharePoint 2013](http://msdn.microsoft.com/en-us/library/ms431831.aspx). Apps for SharePoint use some tokens that apply to site URLs.

### Web part pages

Web Part pages can contain structured and unstructured information. They are made up of Web Part zones. Web Parts placed in Web Part zones can display data from lists, search results, and queries, and can present custom views of data from multiple sources.

A Web Part page contains most of the same regions as a standard SharePoint Team site. The Title bar can contain a title, caption, description, company logo, or other image. The Web Part Page adds the following elements:

* A Web Part Page menu that can be used to add or modify Web Parts, design the page layout, and switch between personal and shared views.
* A tool pane used to find and add Web Parts and edit properties related to Web Parts and the Web Part page.

### Wiki pages

Compared to Web Part pages, wiki pages are less structured. Because of their semi-structured to unstructured form, they make it easy for users to create content and collaborate with each other. By default, SharePoint displays a wiki page the first time you view a new Team site.

Enterprise wiki functionality is available in all versions of SharePoint. The Enterprise Wiki template makes it possible to create and use page layouts with wiki pages. When you edit a wiki page, Web Parts, text, and other content is displayed in the text layout. The text layout arranges content areas on a wiki page.

* + - 1. Using CSOM to create and manage HTML content on a wiki page

You can use the remote provisioning pattern to create a wiki page. The [WikiPageCreationInformation](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.utilities.wikipagecreationinformation.aspx) class provides methods you can use to create the wiki page, while the [WikiHtmlContent](http://msdn.microsoft.com/EN-US/library/office/microsoft.sharepoint.client.utilities.wikipagecreationinformation.wikihtmlcontent.aspx) property gets and sets HTML content on the page. The [Utility](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.utilities.utility.aspx) class includes a [CreateWikiPageInContextWeb](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.utilities.utility.createwikipageincontextweb.aspx) method, which SharePoint uses to create the wiki page in the client runtime context using parameters from the [WikiPageCreationInformation](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.utilities.wikipagecreationinformation.aspx) class.

See the [ProvisionWikiPages](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-5db977e8) sample to learn how to provision a wiki page using the remote provisioning pattern.

### Page layouts

The page layout is the content page of choice for Publishing sites. Page layouts are templates that define different kinds of pages in a SharePoint site, such as articles, by customizing the structure of the body of the page. Just as the Web Part page is a template that exists to arrange Web Part zones and Web Parts on a page, page layouts exist to arrange *fields* on a page. The field controls defined in a page layout will contain content that an author creates, and the structure of that content will be based on the page layout.

**Note** Page layouts can include Web Part zones.

* + - 1. Page field controls

Designers can apply styles to page field controls. This gives designers control over how CSS is applied to each field and rendered, yet allows users to create and manage content in each page field.

* + - 1. Content types in page layouts

In SharePoint, content types are reusable collections of metadata (also known as columns) and behavior that define specific items and documents. For example, you may want to create a kind of content that looks and behaves the way you think an online magazine article would: content types make it possible for you to do that. You may also want to create other unique kinds of content, but reuse and share characteristics of one content type in others.

To learn more about content types in detail, see the [Introduction to content types](http://msdn.microsoft.com/en-us/library/office/ms472236(v=office.14).aspx), [Columns](http://msdn.microsoft.com/en-us/library/office/ms196085(v=office.14).aspx), and [Custom information in content types](http://msdn.microsoft.com/en-us/library/office/ms468437(v=office.14).aspx) articles on MSDN.

Every page layout is based on exactly one content type. Every content type is assigned a unique [Content Type ID](http://msdn.microsoft.com/en-us/library/office/aa543822(v=office.14).aspx), which ensures its uniqueness and enables it to be recursive.

**Important** Currently, you can use the remote provisioning pattern to apply out-of-the-box page layouts to a SharePoint site. While you can provision custom content types on a site using CSOM code via custom apps for SharePoint code, and setting custom **ContentTypeId** via CSOM is supported in SharePoint Online, setting the **ContentTypeId** for a custom content type via remote provisioning on on-premises SharePoint sites is not currently supported.

[How to: Create a page layout in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/jj822368.aspx)

## Page processing model

SharePoint is a template-based page rendering system. It combines master pages, content pages, and authored content to render pages. The page rendering system is known as the [page processing model](http://msdn.microsoft.com/en-us/library/office/ms498550(v=office.14).aspx).

Master pages are used by all page instances in the site to which they are applied, and content pages are used by all instances of the page that are based on that content page.

The page processing model interprets and runs all the requests that user agents such as web browsers make to the server. For example, consider a user requesting a page called *contoso.aspx*. To complete the request, the ASP.NET engine retrieves two pages: the content page associated with *contoso.aspx*, and the master page that the file provider associated with the SharePoint site. The engine also retrieves the field controls and Web Parts from fields and renders them on the page.

While the page processing model in the solution pack is specific to Publishing pages, the page processing logic for Team sites and OneDrive for Business sites is very similar.

### Processing a Web Part page

When a SharePoint user loads a Web Part page, SharePoint gets it by parsing the path to its template, page content, and context. It also sets the Web Parts associated with the Web Part page, assigns a [WebPartCollection](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.webparts.webpartcollection(v=vs.110).aspx) to the page, and populates the Web Part page and its Web Parts with content.

### Processing a Wiki page

When a SharePoint user loads a wiki page (either using the Enterprise Wiki template on a Team site or a Publishing site), SharePoint gets it by parsing the path to its template, page content, and context. It also sets the text layout control associated with the wiki page, and populates the enterprise wiki page and its text layout with content.

See the [ProvisionWikiPages](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-5db977e8) sample in this solution pack to learn how to provision a wiki page using the remote provisioning pattern.

### Minimal download strategy and <AjaxDelta> controls

In SharePoint, the minimal download strategy (MDS) feature manages which specific content on a master page to refresh before the page renders. When MDS is enabled, the content associated with content placeholders wrapped in <SharePoint:AjaxDelta> tags on the master page, the content associated with those placeholders refreshes before the page renders. Conversely, content placeholders not wrapped in <SharePoint:AjaxDelta> tags does not render when MDS is enabled.

You can enable or disable MDS through central site administration or by using the SharePoint client-side object model (CSOM). You can activate the feature using the [EnableMinimalDownload](http://msdn.microsoft.com/en-US/library/office/microsoft.sharepoint.client.web.enableminimaldownload.aspx) property. To learn more about MDS and to see sample CSOM code that enabled it, see [Minimal Download Strategy overview](http://msdn.microsoft.com/library/office/dn456544.aspx). For a more thorough look at how to optimize a master page to work well with MDS, see [Modify SharePoint components for MDS](http://msdn.microsoft.com/en-us/library/office/dn456543.aspx).

MDS is enabled by default on SharePoint Team sites, and disabled by default on SharePoint Publishing sites and SharePoint Team sites with Publishing enabled.

[Optimize page performance in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/dn449110.aspx)

### Creating a custom master page based on seattle.master

While you can use remote provisioning to provision site branding elements such as themes to a site, and use CSS or JavaScript to show or hide regions or page controls, there may be times when customizing a master page provides an additional level of control over page structure.

When creating a custom master page, never edit and save a default master page (e.g., seattle.master) using its default name. Instead, make a copy of the default master page you want to modify, and rename it.

**Important** Because of the potential long-term impact of ongoing support costs and maintenance, Microsoft recommends not changing the structure of a new master page. You can make changes to the master page that support branding that don’t affect the structure, such as changing colors in the header, adding a color background to specific regions of a page, or showing and hiding a site logo. If the default .master page you’re using does not include a structural element, such as a footer, that you want to include on your page, use a different out-of-the-box master page.

To help maintain consistency in a custom master page, follow the existing coding pattern. For example, in regions of the page that use tables, reinforce the coding pattern by using tables. In areas where <DIV> tags or HTML5 are used, match any custom code with <DIV> tags or HTML5. In the long run, this will make any custom master pages you find that you do have to create easier to maintain, and therefore, less expensive.

## Conclusion

In this module, you learned about the SharePoint 2013 pages and the page model at a high level.

Module 3 introduces the tools options available to SharePoint designers and developers for creating branding assets, converts them into formats that SharePoint recognizes and uses, and demonstrates how to use the remote provisioning pattern to apply branding elements to a SharePoint site.