

Questions:

1. What is a Master Page?
2. What are the 2 important parts of a master page?
3. Can Master Pages be nested?
4. What is the file extension for a Master Page?
5. How do you identify a Master Page?
6. Can a Master Page have more than one ContentPlaceHolder?
7. What is a ContentPlaceHolder?
8. How do you bind a Content Page to a Master Page?
9. Can the content page contain any other markup outside of the Content control?
10. What are the advantages of using Master Pages?
11. What are the 3 levels at which content pages can be attached to Master Page?
12. What is @MasterType directive used for?
13. Are controls on the master page accessible to content page code?
14. At what stage of page processing master page and content page are merged?
15. Can you dynamically assign a Master Page?
16. Can you access controls on the Master Page without using FindControl() method?
17. Can you access controls on the Master Page without using FindControl() method?

Questions and Answers:

1. What is a Master Page?

Answer: ASP.NET master pages allow you to create a consistent layout for the pages in your application. A single master page defines the look and feel and standard behavior that you want for all of the pages (or a group of pages) in your application. You can then create individual content pages that contain the content you want to display. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page.

2. What are the 2 important parts of a master page?

Answer: The following are the 2 important parts of a master page

- 1) The Master Page itself
- 2) One or more Content Pages

3. Can Master Pages be nested?

Answer: Yes, Master Pages be nested.

4. What is the file extension for a Master Page?

Answer: .master

5. How do you identify a Master Page?

Answer: The master page is identified by a special @ Master directive that replaces the @ Page directive that is used for ordinary .aspx pages.

6. Can a Master Page have more than one ContentPlaceHolder?

Answer: Yes, a Master Page can have more than one ContentPlaceHolder

7. What is a ContentPlaceHolder?

Answer: ContentPlaceHolder is a region where replaceable content will appear.

8. How do you bind a Content Page to a Master Page?

Answer: MasterPageFile attribute of a content page's @ Page directive is used to bind a Content Page to a Master Page.

9. Can the content page contain any other markup outside of the Content control?

Answer: No.

10. What are the advantages of using Master Pages?

Answer:

- a) They allow you to centralize the common functionality of your pages so that you can make updates in just one place.
- b) They make it easy to create one set of controls and code and apply the results to a set of pages. For example, you can use controls on the master page to create a menu that applies to all pages.
- c) They give you fine-grained control over the layout of the final page by allowing you to control how the placeholder controls are rendered.

- d) They provide an object model that allows you to customize the master page from individual content pages.

11. What are the 3 levels at which content pages can be attached to Master Page?

Answer:

- I. At the page level - You can use a page directive in each content page to bind it to a master page
- II. At the application level - By making a setting in the pages element of the application's configuration file (Web.config), you can specify that all ASP.NET pages (.aspx files) in the application automatically bind to a master page.
- III. At the folder level - This strategy is like binding at the application level, except that you make the setting in a Web.config file in one folder only. The master-page bindings then apply to the ASP.NET pages in that folder.

12. What is @MasterType directive used for?

Answer: @MasterType directive is used to create a strongly typed reference to the master page.

13. Are controls on the master page accessible to content page code?

Answer: Yes, controls on the master page are accessible to content page code.

14. At what stage of page processing master page and content page are merged?

Answer: During the initialization stage of page processing, master page and content page are merged.

15. Can you dynamically assign a Master Page?

Answer: Yes, you can assign a master page dynamically during the PreInit stage using the Page class MasterPageFile property as shown in the code sample below.

```
void Page_PreInit(Object sender, EventArgs e)
{
    this.MasterPageFile = "~/MasterPage.master";
}
```

16. Can you access nonpublic properties and nonpublic methods of a master page inside a content page?

Answer: No, the properties and methods of a master page must be public in order to access them on the content page.

17. Can you access controls on the Master Page without using FindControl() method?

Answer: Yes, by casting the Master to your MasterPage as shown in the below code sample.

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
    MyMasterPage MMP = this.Master;
```

```
    MMP.MyTextBox.Text = "Text Box Found";
```

```
}
```

Deccansoft