

HTML5 | Introduction

Introduction: HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is an abbreviation of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text document within the tag which defines the structure of web pages. HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

Features:

- It has introduced new multimedia features which supports both audio and video controls by using <audio> and <video> tags.
- There are new graphics elements including vector graphics and tags.
- Enrich semantic content by including <header> <footer>, <article>, <section> and <figure> are added.
- Drag and Drop- The user can grab an object and drag it further dropping it to a new location.
- Geo-location services- It helps to locate the geographical location of a client.
- Web storage facility which provides web application methods to store data on the web browser.
- Uses SQL database to store data offline.
- Allows drawing various shapes like triangle, rectangle, circle, etc.
- Capable of handling incorrect syntax.
- Easy DOCTYPE declaration i.e., <!doctype html>
- Easy character encoding i.e., <meta charset="UTF-8">

Removed elements from HTML 5: There are many elements which are depreciated from HTML 5 are listed below:

Removed Elements	Use Instead Elements
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS

Removed Elements	Use Instead Elements
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<isindex>	
<strike>	CSS, <s> or
<tt>	CSS

New Added Elements in HTML 5:

- **<article>**: The <article> tag is used to represent an article. More specifically, the content within the <article> tag is independent from the other content of the site (even though it can be related).
- **<aside>**: The <aside> tag is used to describe the main object of the web page in a shorter way like a highlighter. It basically identifies the content that is related to the primary content of the web page but does not constitute the main intent of the primary page. The <aside> tag contains mainly author information, links, related content and so on.
- **<figcaption>**: The <figcaption> tag in HTML is used to set a caption to the figure element in a document.
- **<figure>**: The <figure> tag in HTML is used to add self-contained content like illustrations, diagrams, photos or codes listing in a document. It is related to main flow, but it can be used in any position of a document and the figure goes with the flow of the document and if it is removed it should not affect the flow of the document.
- **<header>**: It contains the section heading as well as other content, such as a navigation links, table of contents, etc.
- **<footer>**: The <footer> tag in HTML is used to define a footer of HTML document. This section contains the footer information (author information, copyright information, carriers etc.). The footer tag is used within body tag. The <footer> tag is new in the HTML 5. The footer elements require a start tag as well as an end tag.
- **<main>**: Delineates the main content of the body of a document or web app.
- **<mark>**: The <mark> tag in HTML is used to define the marked text. It is used to highlight the part of the text in the paragraph.

- **<nav>**: The <nav> tag is used to declaring the navigational section in HTML documents. Websites typically have sections dedicated to navigational links, which enables user to navigate the site. These links can be placed inside a nav tag.
- **<section>**: It demarcates a thematic grouping of content.
- **<details>**: The <details> tag is used for the content/information which is initially hidden but could be displayed if the user wishes to see it. This tag is used to create interactive widget which user can open or close it. The content of details tag is visible when open the set attributes.
- **<summary>**: The <summary> tag in HTML is used to define a summary for the <details> element. The <summary> element is used along with the <details> element and provides a summary visible to the user. When the summary is clicked by the user, the content placed inside the <details> element becomes visible which was previously hidden. The <summary> tag was added in HTML 5. The <summary> tag requires both starting and ending tag.
- **<time>**: The <time> tag is used to display the human-readable data/time. It can also be used to encode dates and times in a machine-readable form. The main advantage for users is that they can offer to add birthday reminders or scheduled events in their calendars and search engines can produce smarter search results.
- **<bdi>**: The <bdi> tag refers to the Bi-Directional Isolation. It differentiates a text from other text that may be formatted in different direction. This tag is used when a user generated text with an unknown direction.
- **<wbr>**: The <wbr> tag in HTML stands for word break opportunity and is used to define the position within the text which is treated as a line break by the browser. It is mostly used when the used word is too long and there are chances that the browser may break lines at the wrong place for fitting the text.
- **<datalist>**: The <datalist> tag is used to provide autocomplete feature in the HTML files. It can be used with input tag, so that users can easily fill the data in the forms using select the data.
- **<keygen>**: The <keygen> tag in HTML is used to specify a key-pair generator field in a form. The purpose of <keygen> element is to provide a secure way to authenticate users. When a form is submitted then two keys are generated, private key and public key. The private key stored locally, and the public key is sent to the server. The public key is used to generate client certificate to authenticate user in future.
- **<output>**: The <output> tag in HTML is used to represent the result of a calculation performed by the client-side script such as JavaScript.
- **<progress>**: It is used to represent the progress of a task. It also defines how much work is done and how much is left to download a task. It is not used to represent the disk space or relevant query.
- **<svg>**: It is the Scalable Vector Graphics.
- **<canvas>**: The <canvas> tag in HTML is used to draw graphics on web page using JavaScript. It can be used to draw paths, boxes, texts, gradient and adding images. By default, it does not contain border and text.
- **<audio>**: It defines the music or audio content.
- **<embed>**: Defines containers for external applications (usually a video player).
- **<source>**: It defines the sources for <video> and <audio>.
- **<track>**: It defines the tracks for <video> and <audio>.
- **<video>**: It defines the video content.

Advantages:

- All browsers supported.
- More device friendly.
- Easy to use and implement.

- HTML 5 in integration with CSS, JavaScript, etc. can help build beautiful websites.

Disadvantages:

- Long codes have to be written which is time consuming.
- Only modern browsers support it.

Supported Browsers: It is supported by all modern browsers.

Below examples illustrate the HTML 5 content.

Example 1:

html

```
<!DOCTYPE html>
<html>

<head>
  <title>HTML 5</title>
  <style>
    h1 {
      font-size:50px;
    }
  </style>
</head>

<body>
  <h1>PREPARED BY NEELAM</h1>
</body>

</html>
```

Output:

PREPARED BY NEELAM

Introduction to CSS3

What is CSS?

CSS, also known as Cascading Style Sheets, is a game-changing language that revolutionized the web design process. With its simple design, CSS makes it easy to enhance the presentation of web pages, by applying styles independently of the HTML that makes up each page. Essentially, CSS is responsible for the aesthetic appeal of a website, dictating everything from font choice to color scheme and spacing. With this language, developers and designers can create visually appealing websites that truly stand out.

Unlike HTML, which uses tags, CSS uses rulesets. Despite its powerful control over the presentation of an HTML document, CSS is easy to learn and understand. In fact, it offers many benefits that make it a highly practical choice for web development:

- **Time-saving:** With CSS, you can write styles once and apply them to multiple HTML pages. This means you won't have to repeat the same style changes on every page.
- **Easy maintenance:** CSS simplifies the process of updating your website's style. Making a global change is as simple as updating the style sheet, and all the relevant elements across all webpages will be updated automatically.
- **SEO-friendly:** CSS is considered a clean coding technique that search engines can easily read and understand. This means your website's content will be more accessible to search engines, leading to higher rankings and more traffic.
- **Advanced styling features:** Compared to HTML attributes, CSS offers a much wider array of attributes, allowing for more creative and sophisticated designs.
- **Offline browsing:** CSS enables web applications to be stored locally with the help of an offline cache. This allows users to view websites offline, without needing to be connected to the internet.

CSS3 is the latest version of CSS, which offers even more advanced features and increased efficiency. With CSS3, developers can create highly personalized and user-friendly websites that truly stand out from the crowd.

CSS Syntax

A CSS consists of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. These style rules are made up of three essential parts, each of which contributes to the overall aesthetic of your website:

- **Selectors** determine which HTML tags the style rules will be applied to. This could be any tag, from a simple <h1> heading to a complex <table> element.
- **Properties** represent the attributes of HTML tags. Essentially, all the attributes that you'd typically use in HTML can be converted into CSS properties. These properties can range from simple ones like *color* to more complex ones like *border*.
- **Values** are assigned to properties and dictate how they will be displayed on the web page. For example, a *color* property can have a value of *red* or *#F1F1F1*, among others.

Syntax:

```
selector { property: value }
```

CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces. Example : In the following example all p elements will be center-aligned, with a blue text color:

```
p {  
color: blue;  
text-align: center;  
}
```

Types of CSS

There are three types of CSS:

- Inline CSS
- Internal or Embedded CSS
- External CSS

Inline CSS

Inline CSS is used to apply styling to a single HTML element. This type of CSS is added directly to the HTML element using the "style" attribute. Inline CSS is easy to use and can be added quickly, but it should be used sparingly as it can make the HTML code cluttered and difficult to read. Inline CSS is typically used for small changes, such as changing the color of a single element.

Example:

```
<!DOCTYPE html>

<html>

  <head>

    <title>Inline CSS</title>

  </head>

  <body>

    <p style = "color:#009900; font-size:50px;
      font-style:italic; text-align:center;">
      GeeksForGeeks
    </p>

  </body>

</html>
```

Internal CSS

Internal CSS is used to apply styling to a single HTML document. This type of CSS is added to the head section of the HTML document using the "style" tag. Internal CSS is more organized than inline CSS, but it can still be difficult to maintain if the HTML document is large. Internal CSS is typically used for medium-sized web pages, as it allows for more complex styling than inline CSS. Internal style sheets increase page load times, that's why many developers avoid such practice

```
<!DOCTYPE html>

<html>

<head>

<style>

body {

  background-color: black;

}

h1 {

  color: red;

  padding: 50px;

}

</style>

</head>
```

```
<body>
<h2>CSS types</h2>
<p>Cascading Style sheet types: inline, external and internal</p>
</body>
</html>
```

External CSS

External CSS contains a separate CSS file that contains only style properties with the help of tag attributes (For example class, id, heading, ... etc). CSS property is written in a separate file with a .css extension and should be linked to the HTML document using a link tag. This means that for each element, style can be set only once and that will be applied across web pages. External CSS is the most organized type of CSS, as it allows for easy maintenance and updating of the styling across multiple HTML documents. External CSS is typically used for large web pages or websites.

```
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<p>This text is styled by the 'styles.css' file.</p>
</body>
```

- **link** tag is used to link the external style sheet with the html webpage.
- **href** attribute is used to specify the location of the external style sheet file.

Conclusion

In conclusion, Cascading Style Sheets (CSS) is a language that revolutionized the web design process by making it easier to enhance the presentation of web pages by applying styles independently of the HTML that makes up each page. With its powerful control over the presentation of an HTML document, CSS is easy to learn and understand, offers many benefits such as time-saving, easy maintenance, SEO-friendly, advanced styling features, and offline browsing. CSS3, the latest version of CSS, offers even more advanced features and increased efficiency, enabling developers to create highly personalized and user-friendly websites that truly stand out from the crowd. CSS syntax is made up of style rules that consist of selectors, properties, and values. CSS also has three types: inline, internal or embedded, and external, each with its own unique advantages and disadvantages. By understanding CSS and its various types, developers and designers can create visually appealing websites that are easy to maintain and update.

How to Create a Website using HTML And CSS

In this section, let's create a full-fledged website using only HTML and CSS. Most of the users have a question today – Can you create a website just using HTML and CSS?

It is quite possible to create a good-looking website with the help of only HTML and CSS. HTML stands for Hypertext markup language and provides the skeleton for our website. However, CSS (Cascading Style Sheet) allows the skeleton to be more good-looking. Let us use seven steps to create a good-looking website from scratch.

Step 1: Create a Layout

First create a basic structure of your website as a rough sketch. There are a lot of free online services that will help you design your website. Nonetheless, you must have a basic structure of the website ready.



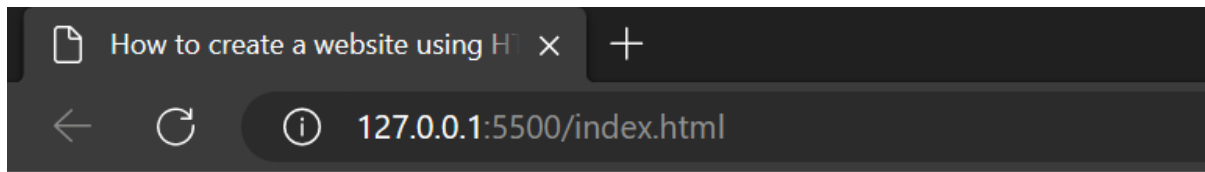
Step 2: Set up the boiler code

Create a new project folder and create an empty index.html file inside the folder. Here, add the boilerplate code to the HTML file.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>How to create a website using HTML and CSS</title>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<h1>Test</h1>
</body>
```


</html>

Before starting the actual content add some test content in your HTML file, and run it on the browser to test if the code is working fine.



Test

Step 3: Create major elements in the layout

Create section elements in the HTML file.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>How to create a website using HTML and CSS</title>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<header>

</header>
<main>
<section id="intro">

</section>

<section id="about">

</section>

<section id="contact">

</section>
```

```
</main>
```

```
<footer>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Read More: [Browser compatibility with CSS Gradients](#)

Step 4: Create the HTML content

In the previous step, you had created the elements in the layout. In this step, fill in the HTML content. Note that, in this example, let us fill the content with dummy text only.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>How to create a website using HTML and CSS</title>
```

```
<link rel="stylesheet" href="css/style.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="#intro">Home</a></li>
```

```
<li><a href="#about">About</a></li>
```

```
<li><a href="#contact">Contact</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<main>
```

```
<section id="intro">
```

```
<div class="Container">
```

```

```

```
<h2>My name is Doggo</h2>
```

```
</div>
```

```
</section>
```

```
<section id="about">
```

```
<div class="container">
```

```
<h1>About Me</h1>
```

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint, similique?</p>
```

```
<ul>
```

```
<li>Btech Qualified</li>
```

```
<li>Software Engineer</li>
```

```
<li>GATE AIR 01</li>
```

```
</ul>
```

```
</div>
```

```
</section>
```

```
<section id="contact">
```

```
<div class="container">
```

```
<h1>Contact me</h1>
```

```
<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nam, laudantium.</p>
```

```
<ul>
```

```
<li>Email ID</li>
```

```
<li>Insta ID</li>
```

```
<li>Facebook ID</li>
```

```
</ul>
```

```
</div>
```

```
</section>
```

```
</main>
```

```
<footer>
```

```
<p>© Copyright 2022 Doggo Co LTD.</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Now, if you reload the page, you are going to get an output something like this. You are now going to give this webpage some CSS in the next step to make it good-looking.

- [Home](#)
- [About](#)
- [Contact](#)



My name is Doggo

About Me

Lorem ipsum dolor sit amet consectetur adipisicing elit. Sint, similique?

- Btech Qualified
- Software Engineer
- GATE AIR 01

Contact me

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nam, laudantium.

- Email ID
- Insta ID
- Facebook ID

© Copyright 2022 Doggo Co LTd.

Step 5: Create CSS for the layout

Before adding the depth in the CSS, let us first add some basic CSS to make our webpage look somewhat similar to the layout that we designed in the first step.

Moreover, we linked our HTML file to a CSS file in the second step while writing our boilerplate code. Add the basic layout CSS in the linked CSS file. In this step, we are going to focus on height, width, padding, margin, and display of the sections and images, to make them adjustable according to the webpage.

```
*{  
padding: 0;  
margin: 0;  
}
```

```
header{  
height: 45px;  
}
```

```
header nav ul{  
display: flex;  
margin-left: 80%;  
}
```

```
header nav ul li{  
padding-left: 10%;  
}
```

```
section{  
height: 100vh;  
border: 1px solid grey;  
display: flex;  
justify-content: center;  
align-items: center;  
}
```

```
.Container{  
margin-top: 10%  
}
```

```
.Container img{  
height: 300px;  
}
```

```
.Container h2{  
margin-top: 3%;  
}
```

```
footer {  
line-height: 40px;
```

```
display: flex;
justify-content: center;
}
```

Learn More: [What are CSS Breakpoints and Media Query Breakpoints](#)

Step 6: Create CSS to style individual elements

In this step let us style individual content. Let us focus on properties like font, border, colors, and more.

```
*{
padding: 0;
margin: 0;
}

header{
height: 45px;
}

header nav ul{
display: flex;
margin-left: 70%;
list-style: none;
}

header nav ul li{
padding-left: 10%;
}

header a{
text-decoration: none;
color: brown;
}

section{
height: 100vh;
border: 1px solid grey;
display: flex;
```

```
justify-content: center;
align-items: center;
}
```

```
.Container img{
height: 300px;
border-radius: 50%;
}
```

```
.Container h2{
margin-top: 2%;
font-size: 3em;
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
}
```

```
.Container p, ul{
margin-top: 2%;
font-size: 1.5rem;
}
```

```
footer {
line-height: 40px;
display: flex;
justify-content: center;
font-size: 1rem;
}
```

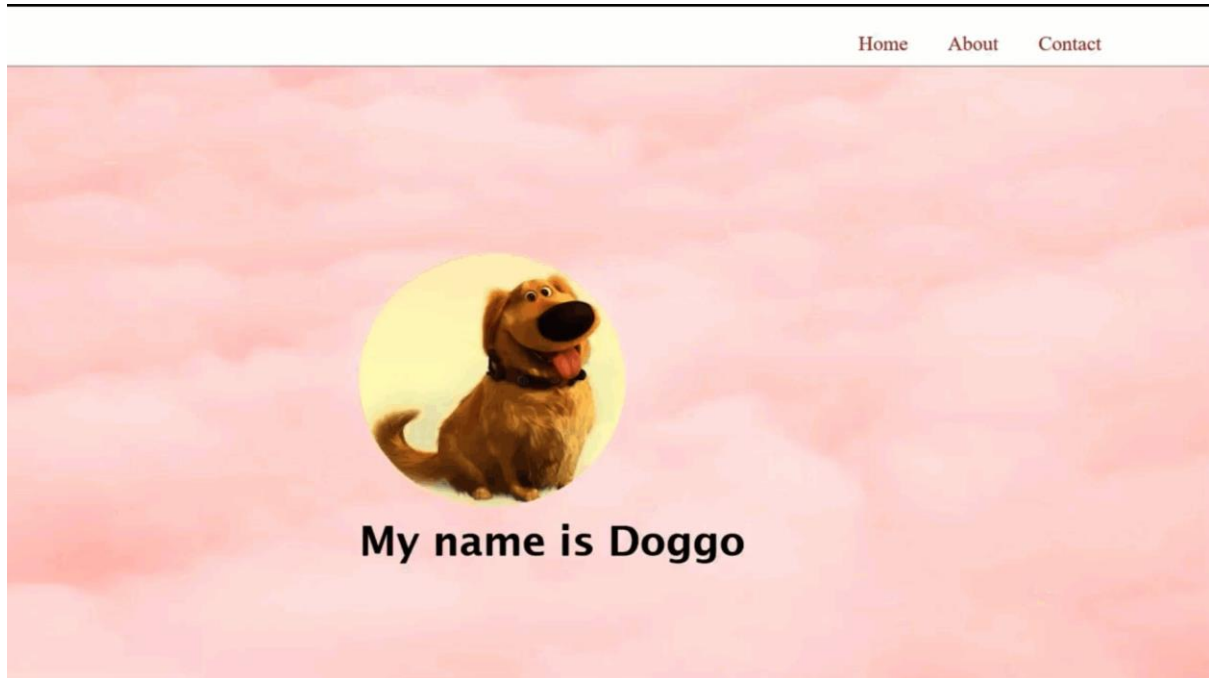
Step 7: Add background color and style

In this step, let us add some finishing touches, and our website is ready. Let us add a background image and background colors to the sections.

```
#intro {
background-image: url(pinkbg.jpg);
background-repeat: round;
}
#about{
background-color: bisque;
}
```

```
#contact{  
background-color: blanchedalmond;  
}
```

After completing the entire code of our website, it will look something like this. Note that you can add more CSS to make it further good-looking.



[Talk to an Expert](#)

How to optimize HTML code for Website?

View the Source Code of an HTML Document

To view the source code of any webpage:

- Navigate to the webpage
- Right-click on the page
- Select "view page source"

Alternatively, you may use a keyboard shortcut **CTRL + U** or **CMD + U** to inspect the source code of any HTML document.

The source code of an HTML document will look something like this.


```
view-source:https://www.browserstack.com
Line wrap
1 <!doctype html><html lang="en-US" class="no-js"><head> <meta charset="UTF-8"> <title>Most Reliable App & Cross Browse
2   var stage_env_cookie_prefix = '';
3   function getCookieVal(e){for(var t=(stage_env_cookie_prefix+e).trim()+"=",a=document.cookie.split(";"),i=0;i<a.length
4 </script> <link onload="this.media == 'none'?this.media='all':"" rel="stylesheet" href="https://browserstack.wpenginep
5 /** Objects and variables defined so they are always defined when required */
6   var BrowserStackEnterprise = false, existingRows = 0, js_theme_path = {'templateUrl': 'https://browserstack.wpengine
7   if (typeof userDetails === 'undefined') {
8     var userDetails = { location: {city: '', country: '', ip: '', region: '', longitude: '', latitude: ''} };
9   }
10 </script><script>
11 window.addEventListener("error", function(e, url) {
12   var eleArray = ["SCRIPT"];
13   if (eleArray.indexOf(e.target.tagName) != -1) {
14     if ((/bstack_/ .test(e.target.src))) {
15       var headerJs=function(){function e(){document.getElementById("header-habitat").innerHTML.length>0&&t()}var t=fu
16     }
17   }
18 }, true);
19 </script> <script type="application/ld+json">
```

Nest HTML Elements

Nesting in HTML is to apply several HTML tags to a single content. In nesting, one element can be placed inside other elements. Another benefit of nesting in HTML includes improving the readability of your code for you and other developers.

Nesting in HTML will look something like this.

```
index.html > ...
1 <html>
2   <head>
3     <body>
4       <div class="parent">
5         <div class="child">
6           <p>
7             <span>
8               <b>This is a bold text trying to display nesting</b>
9             </span>
10          </p>
11        </div>
12      </div>
13    </body>
14  </head>
15 </html>
```

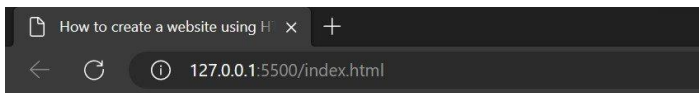
How to optimize CSS code for Website

Style Images with CSS

In this section, let's learn how to style images with CSS such as adding a border to an image, adjusting its dimensions, and further specific CSS to our images in the webpage. First, add an image element in the HTML file.

```

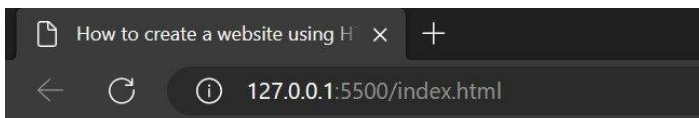
```



Now, it's time to add CSS to the image to make it look good.

```
img{  
height: 300px;  
border-radius: 50%;  
border: 12px dotted rgb(255, 85, 0);  
}
```

This CSS will apply to all the images of our HTML document.



Read More: [How to position Text Over an Image using CSS](#)

Styling Classes With CSS

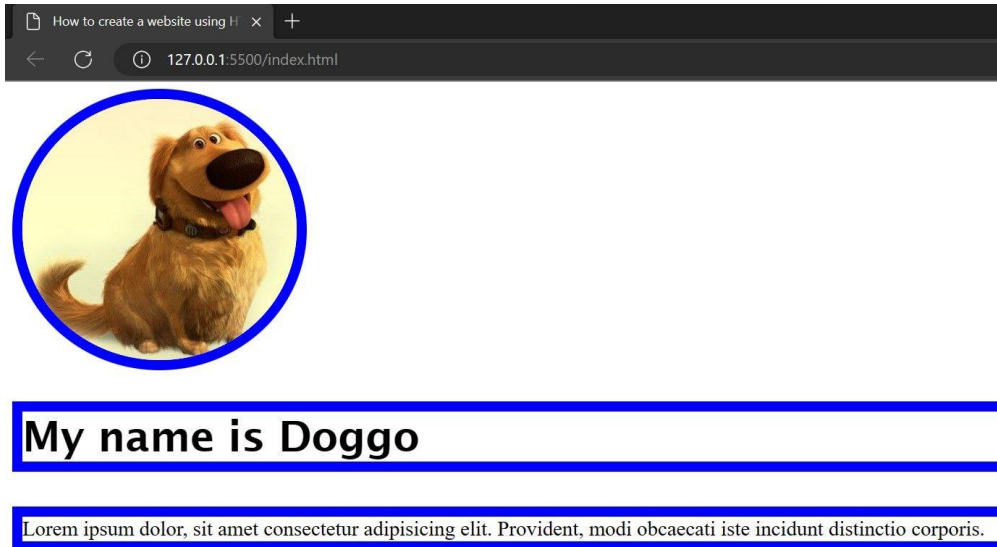
Now, let's see how to create classes with the help of CSS. here, we shall learn how to apply CSS rules only to the HTML elements that have specific classes. First, let's create an HTML element that has some class, and then we shall apply CSS to the entire class. Applying CSS to the entire class will allow us to style all the elements that have the particular class.

```
  
<h2 class="blueBorder">My name is Doggo</h2>  
<p class="blueBorder">Lorem10</p>
```

In this example, we took three different elements having the same class. Once we apply CSS to the class, it will be applied to all the elements belonging to the same class.

```
.blueBorder{  
border: 12px solid blue;  
}
```

To use the class as a selector while creating a CSS rule, we use the ‘.’ symbol before writing the class name in the CSS file.



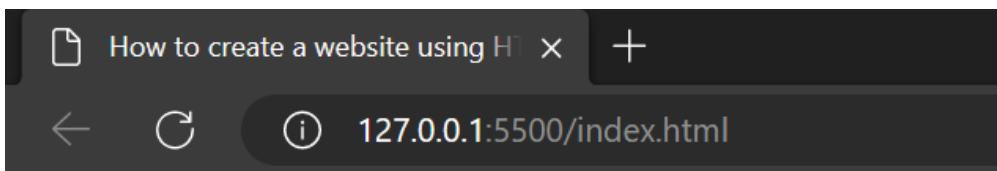
Styling IDs with CSS

Using IDs as selectors while creating CSS rules is similar to using classes as selectors. However, IDs are unique in the HTML document, and no two elements can carry the same IDs. IDs are generally applied to elements that are present only once in the HTML document, such as the navbar, logo, and more.

```
<h2 id="BrowserStack">Using ID as a CSS selector</h2>
```

Now, let's create a CSS rule, using ID as a selector.

```
#BrowserStack{  
font-size: 2rem;  
color: blueviolet;  
background-color: bisque;  
}
```



Using ID as a CSS selector

Creating Pseudo-classes With CSS

Pseudo-classes are classes that are activated when certain keywords are added to a selector that specifies a certain state. For example, the pseudo-class :hover is used to change the state of the element when a user hovers the pointer over it.

Let's take an example when we hover over the previous image, it changes the border color of the image.

```
img:hover{  
border: 12px dotted blue;  
}
```

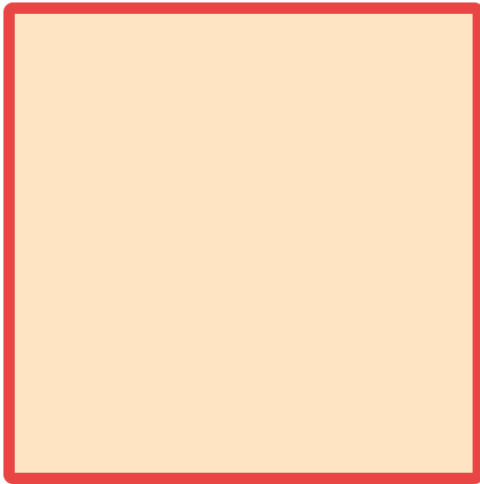
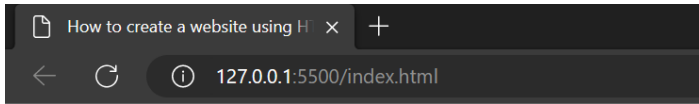
[Home](#) [About](#) [Contact](#)



Styling HTML <div> element with CSS

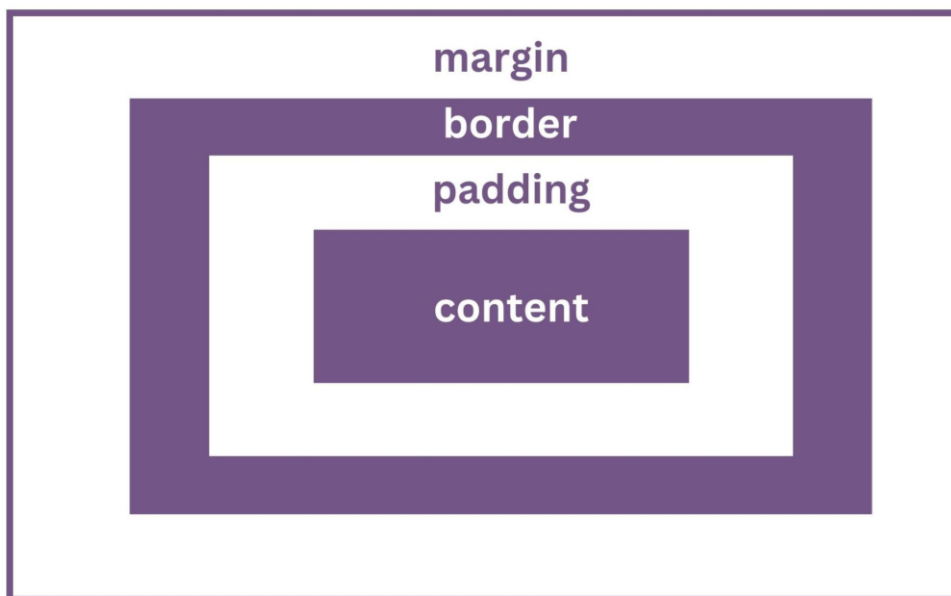
The div tag is often used to specify the container for HTML elements. It can also be used to structure the layout of the webpage. Now, let's understand how to style the div element and its children elements.

```
div{  
background-color: bisque;  
border: 10px solid rgb(232, 69, 69);  
border-radius: 2%;  
height: 50vh;  
width: 50vh;  
}
```



How to adjust the Content, Padding, Border, & Margins of an HTML Element With CSS

Before understanding how to adjust the content, padding, border, and margins of an HTML element, let's understand the CSS box model. It is a box that wraps around every HTML element in the DOM.



CSS box model

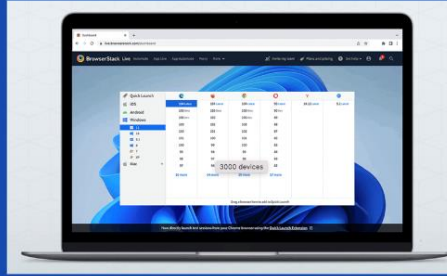
- **Content box:** It is the space where the content of the HTML element appears, such as images, text, and more.
- **Padding:** It is the transparent area around the content of the element.
- **Border:** It is the box surrounding the padding box. By default, the value of the border for every HTML element is zero; however, increasing the border value increases the space between the padding and the margin box.
- **Margin:** It is the transparent area outside the border box.

Let's take the example of the previous image to understand how to adjust these values.

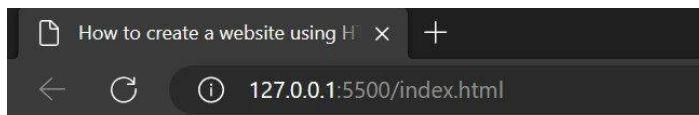
Test on Real Devices for Real Results

Skip the simulators and test directly on real devices to guarantee your application's performance across all browsers.

Contact Sales



```
img{  
height: 300px;  
border-radius: 50%;  
border: 12px dotted rgb(255, 85, 0);  
padding: 10px 10px 20px 20px;  
margin: 20px 20px 15px 10px;  
}
```

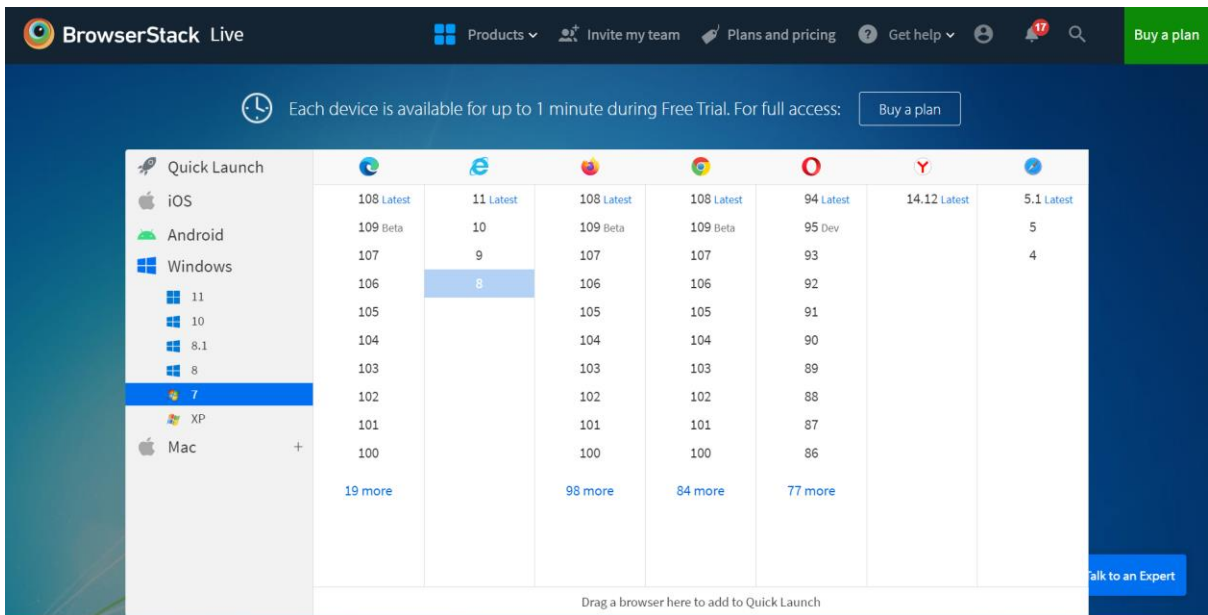


How to test a Website on Real Devices with BrowserStack Live?

BrowserStack allows you to test your website on different browser-device combinations and check if your website is working as expected under real user conditions. Here's how you can test your website on Real Devices & Browsers using BrowserStack Live:

[Sign up for a free BrowserStack account](#)

- **Step 1: Launch [BrowserStack Live](#)**
- **Step 2: Select the Browser-OS-Device combination** on which you want to test your website.

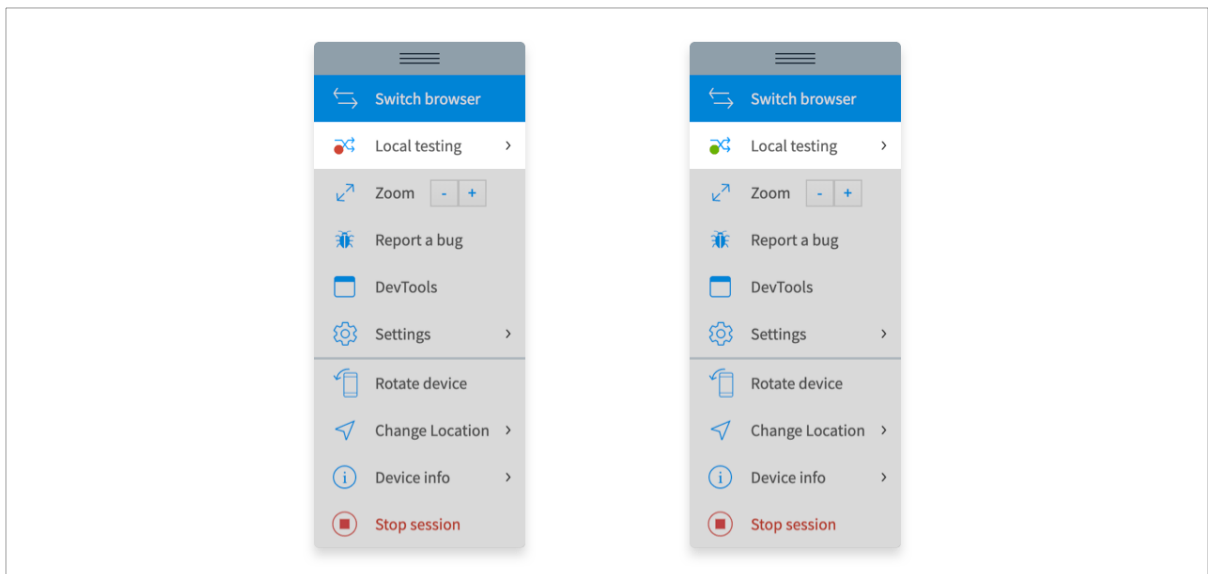


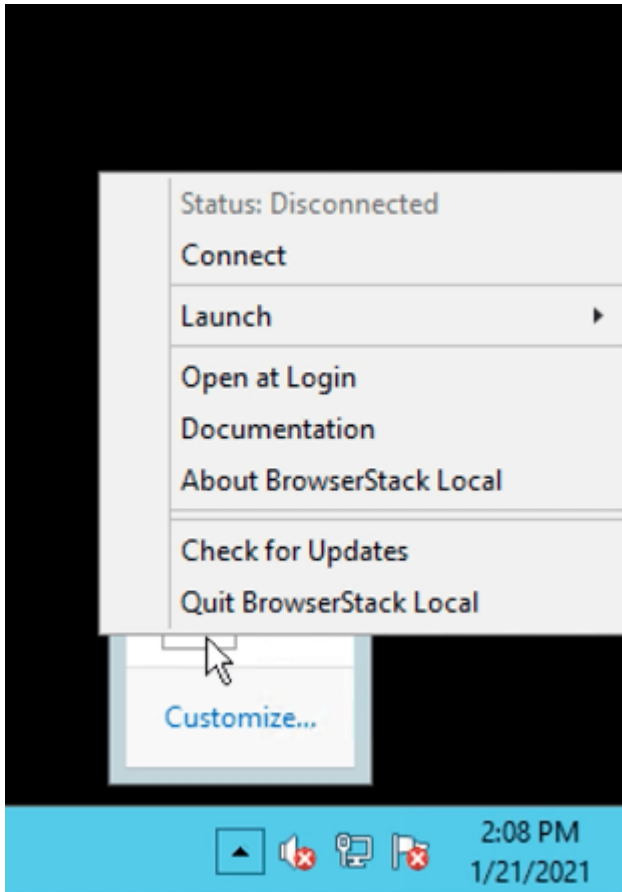
- **Step 3: Enable local testing**

BrowserStack allows you test your local website. To enable BrowserStack local testing, start a session on BrowserStack Live and look for the green indicator on the 'Local Testing' icon in the toolbar tray.

Read More: [Overcoming Challenges of Local Testing](#)

If the icon is red, look for the BrowserStack local app, download it, and launch a live session from the toolbar.

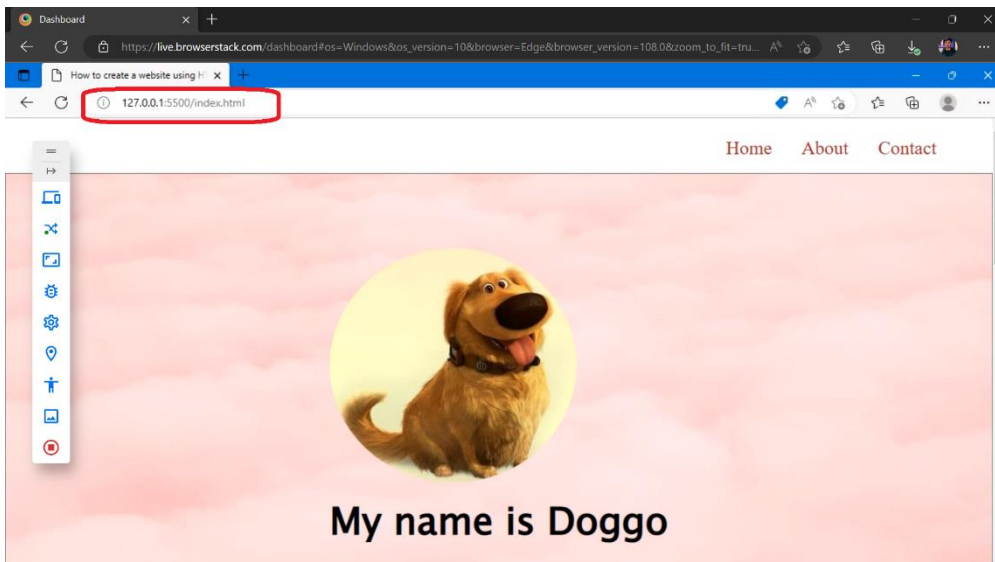




Read More: [What is IP Whitelisting and How it Elevates Local Testing](#)

- **Step 4: Run the test**

Now, the final step is to enter the local host URL of your website, and it will display your website on the Live Session.



PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf released the first version of PHP way back in 1994. Initially, PHP was supposed to be an abbreviation for "Personal Home Page", but it now stands for the recursive initialism "PHP: Hypertext Preprocessor".

Lerdorf began PHP development in 1993 by writing several Common Gateway Interface (CGI) programs in C, which he used to maintain in his personal homepage. Later on, He extended them to work with web forms and to communicate with databases. This implementation of PHP was "Personal Home Page/Forms Interpreter" or PHP/FI.

Today, PHP is the world's most popular server-side programming language for building web applications. Over the years, it has gone through successive revisions and versions.

PHP Versions

PHP was developed by Rasmus Lerdorf in 1994 as a simple set of CGI binaries written in C. He called this suite of scripts "Personal Home Page Tools". It can be regarded as PHP version 1.0.

- In April 1996, Rasmus introduced PHP/FI. Included built-in support for DBM, mSQL, and Postgres95 databases, cookies, user-defined function support. PHP/FI was given the **version 2.0** status.
- PHP: Hypertext Preprocessor – **PHP 3.0** version came about when Zeev Suraski and Andi Gutmans rewrote the PHP parser and acquired the present-day acronym. It provided a mature interface for multiple databases, protocols and APIs, object-oriented programming support, and consistent language syntax.
- **PHP 4.0** was released in May 2000 powered by Zend Engine. It had support for many web servers, HTTP sessions, output buffering, secure ways of handling user input and several new language constructs.
- **PHP 5.0** was released in July 2004. It is mainly driven by its core, the Zend Engine 2.0 with a new object model and dozens of other new features. PHP's development team includes dozens of developers and others working on PHP-related and supporting projects such as PEAR, PECL, and documentation.
- **PHP 7.0** was released in Dec 2015. This was originally dubbed PHP next generation (phpng). Developers reworked Zend Engine is called Zend Engine 3. Some of the important features of PHP 7 include its improved performance, reduced memory usage, Return and Scalar Type Declarations and Anonymous Classes.
- **PHP 8.0** was released on 26 November 2020. This is a major version having many significant improvements from its previous versions. One standout feature is Just-in-time compilation (JIT) that can provide substantial performance improvements. The latest version of PHP is 8.2.8, released on July 4th, 2023.

PHP Application Areas

PHP is one of the most widely used language over the web. Here are some of the application areas of PHP –

- PHP is a server-side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. Although it is especially suited to web development, you can also build desktop standalone applications as PHP also has a command-line interface. You can use PHP-GTK extension to build GUI applications in PHP.
- PHP is widely used for building web applications, but you are not limited to output only HTML. PHP's output abilities include rich file types, such as images or PDF files, encrypting data, and sending emails. You can also output easily any text, such as JSON or XML.
- PHP is a cross-platform language, capable of running on all major operating system platforms and with most of the web server programs such as Apache, IIS, lighttpd and nginx. PHP also supports other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM, etc.

Here are some more important features of PHP –

- PHP performs system functions. It can create, open, read, write, and close the files.
- PHP can handle forms. It can gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.

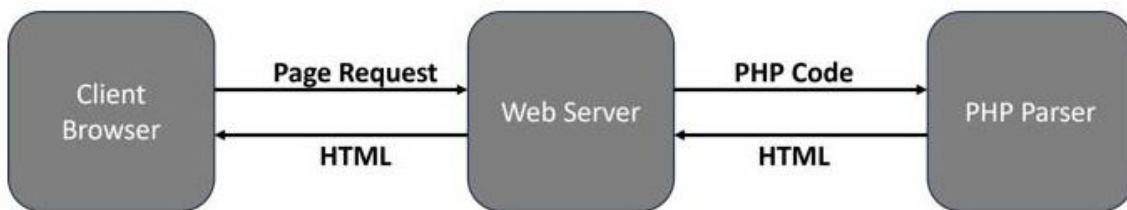
- It can encrypt data.

PHP provides a large number of reusable classes and libraries are available on "PEAR" and "Composer". PEAR (PHP Extension and Application Repository) is a distribution system for reusable PHP libraries or classes. "Composer" is a dependency management tool in PHP.

PHP SYNTAX:

The syntax rules of PHP are very similar to C Language. PHP is a server side scripting language. A PHP code is stored as a text file with ".php" extension. A ".php" file is essentially a web page with one or more blocks of PHP code interspersed in the HTML script. However, it must be opened in a browser with a HTTP protocol URL. In other words, if you double-click on the PHP file icon, it will be opened locally with the file protocol. For example, if you open the "index.php" file in the document root folder of your Apache server, it may just show the text of the PHP code. However, if you launch the Apache server and open the URL <http://localhost/index.php>, it displays the Apache home page.

A ".php" file may contain HTML, CSS and JavaScript code blocks along with the PHP code. Hence, the PHP parser must differentiate between the PHP code from the other elements. When a ".php" file is opened in the web browser, the HTML engine renders the HTML/CSS/JavaScript part and escapes out of the HTML block as soon as the statements included in PHP tags are encountered. The PHP parser interpreter processes this block and returns the response to the browser.



PHP defines two methods of using tags for escaping the PHP code from HTML. Canonical PHP tags and Short-open (SGML-style) tags.

Canonical PHP Tags

The most universally effective PHP tag style is –

<?php

One or more PHP statements

?>

If you use this style, you can be positive that your tags will always be correctly interpreted.

Short-open (SGML-style) Tags

Short or short-open tags look like this –

<?php

One or more PHP statements

?>

Short tags are, as one might expect, the shortest option. You must do one of two things to enable PHP to recognize the tags

–

- Choose the "--enable-short-tags" configuration option when you're building PHP.
- Set the "short_open_tag" setting in your php.ini file to on.

short_open_tag=on

This option must be disabled to parse XML with PHP because the same syntax is used for XML tags.

The use of **ASP-style tags** –

```
<%...%>
```

and **HTML script tags** –

```
<script language = "PHP">...</script>
```

has been discontinued.

[Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](#)

Escaping from HTML

The PHP parser ignores everything outside of a pair of opening and closing tags. Thus, a PHP file can have mixed content. This allows PHP to be embedded in HTML documents –

```
<p>This is a HTML statement</p>
```

```
<?php echo This is a PHP statement.'; ?>
```

```
<p>This is another HTML statement.</p>
```

A little advanced example of escaping using conditions is shown below –

```
<?php if ($expression == true): ?>
```

 This HTML statement will be rendered.

```
<?php else: ?>
```

 Otherwise this HTML statement will be rendered.

```
<?php endif; ?>
```

PHP skips the blocks where the condition is not met, even though they are outside of the PHP open/close tags.

For outputting large blocks of text, dropping out of PHP parsing mode is generally more efficient than sending all of the text through echo or print.

Basic Syntax of PHP

The basic syntax of PHP is very similar to that of C and C++.

Statements are expressions terminated by semicolons

A statement in PHP is any expression that is followed by a semicolon (;). Any sequence of valid PHP statements that is enclosed by the PHP tags is a valid PHP program.

Here is a typical statement in PHP, which in this case assigns a string of characters to a variable called "\$greeting" –

```
$greeting = "Welcome to PHP!";
```

A physical line in the text editor doesn't have any significance in a PHP code. There may be multiple semicolon-terminated statements in a single line. On the other hand, a PHP statement may spill over more than one line if required.

Expressions are combinations of tokens

The smallest building blocks of PHP are the indivisible tokens such as numbers (3.14159), strings ("two"), variables (\$two), constants (TRUE), and the special words that make up the syntax of PHP itself like "if", "else", "while", "for", and so forth.

Braces make blocks

Although statements cannot be combined like expressions, you can always put a sequence of statements anywhere a statement can go, by enclosing them in a set of curly braces.

Here, both the following statements are equivalent –

```
if (3 == 2 + 1)
    print("Good - I haven't totally lost my mind.");

if (3 == 2 + 1) {
    print("Good - I haven't totally");
    print("lost my mind.");
}
```

PHP is case sensitive

PHP is a case sensitive language. The names of various PHP identifiers such as variable, function, class, etc., are case sensitive. As a result, the variable "\$age" is not the same as "\$Age". Similarly, a function called "myfunction()" is different from another function called "MyFunction()".

PHP VARIABLES: A variable in PHP is a named memory location that holds data belonging to one of the data types.

- PHP uses the convention of prefixing a dollar sign (\$) to the name of a variable.
- Variable names in PHP are case-sensitive.
- Variable names follow the same rules as other labels in PHP. A valid variable name starts with a letter or underscore, followed by any number of letters, numbers, or underscores.
- As per the naming convention, "\$name", "\$rate_of_int", "\$Age", "\$mark1" are examples of **valid variable names** in PHP.
- **Invalid variable names:** "name" (not having \$ prefix), "\$rate of int" (whitespace not allowed), "\$Age#1" (invalid character #), "\$11" (name not starting with alphabet).

Variables are assigned with the "=" operator, with the variable on the left hand side and the expression to be evaluated on the right.

No Need to Specify the Type of a Variable

PHP is a dynamically typed language. There is no need to specify the type of a variable. On the contrary, the type of a variable is decided by the value assigned to it. The value of a variable is the value of its most recent assignment.

Take a look at this following **example** –

Open Compiler

```
<?php
```

```
$x = 10;

echo "Data type of x: " . gettype($x) . "\n";

$x = 10.55;

echo "Data type of x now: " . gettype($x) . "";
```

```
?>
```

It will produce the following **output** –

Data type of x: integer

Data type of x now: double

Automatic Type Conversion of Variables

PHP does a good job of automatically converting types from one to another when necessary. In the following code, PHP converts a string variable "y" to "int" to perform addition with another integer variable and print 30 as the result.

Take a look at this following **example** –

Open Compiler

```
<?php
```

```
$x = 10;
```

```
$y = "20";
```

```
echo "x + y is: ", $x+$y;
```

```
?>
```

It will produce the following **output** –

```
x + y is: 30
```

[Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](#)

Variables are Assigned by Value

In PHP, variables are always assigned by value. If an expression is assigned to a variable, the value of the original expression is copied into it. If the value of any of the variables in the expression changes after the assignment, it doesn't have any effect on the assigned value.

Open Compiler

```
<?php
```

```
$x = 10;
```

```
$y = 20;
```

```
$z = $x+$y;
```

```
echo "(before) z = ". $z . "\n";
```

```
$y=5;
```

```
echo "(after) z = ". $z . "";
```

```
?>
```

It will produce the following **output** –

```
(before) z = 30
```

```
(after) z = 30
```

Assigning Values to PHP Variables by Reference

You can also use the way to assign values to PHP variables by reference. In this case, the new variable simply references or becomes an alias for or points to the original variable. Changes to the new variable affect the original and vice versa.

To assign by reference, simply prepend an ampersand (&) to the beginning of the variable which is being assigned (the source variable).

Take a look at this following **example** –

Open Compiler

```
<?php
```

```
$x = 10;

$y = &$x;

$z = $x+$y;

echo "x=". $x . " y=" . $y . " z = " . $z . "\n";
```

```
$y=20;

$z = $x+$y;

echo "x=". $x . " y=" . $y . " z = " . $z . "";
```

```
?>
```

It will produce the following **output** –

```
x=10 y=10 z = 20
```

```
x=20 y=20 z = 40
```

Variable Scope

Scope can be defined as the range of availability a variable has to the program in which it is declared. PHP variables can be one of four scope types –

- [Local Variables](#)
- [Global Variables](#)
- [Static Variables](#)
- [Function Parameters](#)

Variable Naming

Rules for naming a variable is –

- Variable names must begin with a letter or underscore character.
- A variable name can consist of numbers, letters, underscores but you cannot use characters like + , - , % , (,) . & , etc

There is no size limit for variables.

PHP ARRAY: PHP defines the following set of symbols to be used as operators on array data types –

Symbol	Example	Name	Result
+	\$a + \$b	Union	Union of \$a and \$b.

==	\$a == \$b	Equality	TRUE if \$a and \$b have the same key/value pairs.
===	\$a === \$b	Identity	TRUE if \$a and \$b have the same key/value pairs in the same order and of the same types.
!=	\$a != \$b	Inequality	TRUE if \$a is not equal to \$b.
<>	\$a <> \$b	Inequality	TRUE if \$a is not equal to \$b.
!==	\$a !== \$b	Non identity	TRUE if \$a is not identical to \$b.

The Union operator appends the right-hand array appended to left-hand array. If a key exists in both arrays, the elements from the left-hand array will be used, and the matching elements from the right-hand array will be ignored.

Example: Union Operator in PHP

The following example shows how you can use the union operator in PHP –

Open Compiler

```
<?php
$arr1=array("phy"=>70, "che"=>80, "math"=>90);
$arr2=array("Eng"=>70, "Bio"=>80,"CompSci"=>90);
$arr3=$arr1+$arr2;
var_dump($arr3);
?>
```

It will produce the following **output** –

```
array(6) {
  ["phy"]=>
  int(70)
  ["che"]=>
  int(80)
  ["math"]=>
  int(90)
  ["Eng"]=>
  int(70)
  ["Bio"]=>
  int(80)
  ["CompSci"]=>
  int(90)
}
```

Example: When Two Array are Equal

Two arrays are said to be equal if they have the same key-value pairs.

In the following example, we have an indexed array and other associative array with keys corresponding to index of elements in first. Hence, both are equal.

Open Compiler

```
<?php
```

```
$arr1=array(0=>70, 2=>80, 1=>90);
```

```
$arr2=array(70,90,80);
```

```
var_dump ($arr1==$arr2);
```

```
var_dump ($arr2!=$arr1);
```

```
?>
```

It will produce the following **output** –

```
bool(true)
```

```
bool(false)
```

[Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](#)

PHP COOKIES:

The PHP superglobal `$_COOKIE` stores the variables passed to the current PHP script along with the HTTP request in the form of cookies. `$HTTP_COOKIE_VARS` also contains the same information, but it is not a superglobal, and it has now been deprecated.

What is a Cookie?

Cookies are text files stored by a server on the client computer and they are kept for tracking purpose. PHP transparently supports HTTP cookies. Cookies are usually set in an HTTP header. JavaScript can also sets a cookie directly on a browser.

The server script sends a set of cookies to the browser. It stores this information on the local machine for future use. Next time, when the browser sends any request to the web server, it sends those cookies information to the server and the server uses that information to identify the user.

The `setcookie()` Function

PHP provides the **setcookie** function to create a cookie object to be sent to the client along with the HTTP response.

```
setcookie(name, value, expire, path, domain, security);
```

Parameters

- **Name** – Name of the cookie stored.
- **Value** – This sets the value of the named variable.
- **Expiry** – This specifies a future time in seconds since 00:00:00 GMT on 1st Jan 1970.
- **Path** – Directories for which the cookie is valid.
- **Domain** – Specifies the domain name in very large domains.
- **Security** – 1 for HTTPS. Default 0 for regular HTTP.

[Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](#)

How to Set Cookies

Take a look at the following **example**. This script sets a cookie named **username** if it is not already set.

Example

<?php

```
if (isset($_COOKIE['username'])) {  
    echo "<h2>Cookie username already set: " . $_COOKIE['username'] . "</h2>";  
} else {  
    setcookie("username", "Mohan Kumar");  
    echo "<h2>Cookie username is now set.</h2>";  
}
```

?>

Run this script from the document root of the Apache server. You should see this message as the **output** –

Cookie username is now set

If this script is re-executed, the cookie is now already set.

Cookie username already set: Mohan Kumar

Example

To retrieve cookies on subsequent visit of client –

<?php

```
$arr=$_COOKIE;  
foreach ($arr as $key=>$val);  
echo "<h2>$key => $val </h2>";
```

?>

The browser will display the following **output** –

Username => Mohan Kumar

How to Remove Cookies

To delete a cookie, set the cookie with a date that has already expired, so that the browser triggers the cookie removal mechanism.

<?php

```
setcookie("username", "", time() - 3600);  
echo "<h2>Cookie username is now removed</h2>";
```

?>

The browser will now show the following **output** –

Cookie username is now removed

Setting Cookies Using the Array Notation

You may also set the array cookies by using the array notation in the cookie name.

```
setcookie("user[three]", "Guest");
```

```
setcookie("user[two]", "user");
```

```
setcookie("user[one]", "admin");
```

If the cookie name contains dots (.), then PHP replaces them with underscores (_).

PHP MANAGEMENT

The event management system is a web-based application. The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all event-related information. The goal is to support various functions and processes necessary to manage the data efficiently.

Project Requirements

Project Name	Event Management System in PHP
Language Used	PHP5.6, PHP7.x
Database	MySQL 5.x
User Interface Design	HTML, AJAX,JQUERY,JAVASCRIPT
Web Browser	Mozilla, Google Chrome, IE8, OPERA
Software	XAMPP / Wamp / Mamp/ Lamp (anyone)
Last Updated	22 May 2023

Project Modules

In EMS project we use PHP and MySQL database. It has two modules.

1.Admin Module

2.User Module

Admin Module

1. Dashboard: In this section, admin can see all detail in brief like listed categories, Sponsors, Total Events, Total Registered Users, Total Booking, Total New Booking, Total Confirmed Booking and Total Cancelled Booking.
2. Category: In this section, admin manage event category (add and update).
3. Manage Sponsors: In this section, admin can add sponsors and manage sponsors details (Add/Update/Delete).
4. Events: In this section, admin manage events (add and update).
5. Manage Users: In this section, admin can update details of registered users and also block them.
6. Manage Booking: In this section, admin can manage booking by cancel and confirm it.

7. News: In this section admin manage news (add and delete).
8. Website Setting: In this section, admin can update about us and another general website setting.

Admin can also update his profile, change password and recover password.

User Module

In this module there is two types of user guest user and registered user.

Guest User: In this guest user can see only general information like about us, event details, contact details and news about events.

Registered users can do the following activity

1. Books the events.
2. Update his/her own profile.
3. Change Password.
4. Users can also cancel booking which is not confirmed.
5. Registered user can also recover his/her own password.

What is MySQL?

- MySQL is a relational database management system
- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

Who Uses MySQL?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

Show Data On Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (like MySQL)
- A server-side scripting language, like PHP

- To use SQL to get the data you want
- To use HTML / CSS to style the page

What is RDBMS?

RDBMS stands for Relational Database Management System.

RDBMS is a program used to maintain a relational database.

RDBMS is the basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access.

RDBMS uses [SQL queries](#) to access the data in the database.

What is a Database Table?

A table is a collection of related data entries, and it consists of columns and rows.

A column holds specific information about every record in the table.

A record (or row) is each individual entry that exists in a table.

Look at a selection from the Northwind "Customers" table:

CustomerID	CustomerName	ContactName	Address
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
4	Around the Horn	Thomas Hardy	120 Hanover Sq.
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8

The columns in the "Customers" table above are: CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. The table has 5 records (rows).

ADVERTISEMENT

What is a Relational Database?

A relational database defines database relationships in the form of tables. The tables are related to each other - based on data common to each.

Look at the following three tables "Customers", "Orders", and "Shippers" from the Northwind database:

Customers Table

CustomerID	CustomerName	ContactName	Address
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
4	Around the Horn	Thomas Hardy	120 Hanover Sq.
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8

The relationship between the "Customers" table and the "Orders" table is the CustomerID column:

Orders Table

OrderID	CustomerID	EmployeeID	OrderDate
10278	5	8	1996-08-12
10280	5	2	1996-08-14
10308	2	7	1996-09-18
10355	4	6	1996-11-15
10365	3	3	1996-11-27
10383	4	8	1996-12-16
10384	5	3	1996-12-16

The relationship between the "Orders" table and the "Shippers" table is the ShipperID column:

Shippers Table

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-
2	United Package	(503) 555-

3

Federal Shipping

(503) 555-

The MySQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Note: Be careful when updating records in a table! Notice the WHERE clause in the UPDATE statement. The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated!

Demo Database

Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
4	Around the Horn	Thomas Hardy	120 Hanover Sq.

UPDATE Table

The following SQL statement updates the first customer (CustomerID = 1) with a new contact person *and* a new city.

Example

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'
WHERE CustomerID = 1;
```

The selection from the "Customers" table will now look like this:

CustomerID	CustomerName	ContactName	Address
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312

4	Around the Horn	Thomas Hardy	120 Hanover Sq.
---	-----------------	--------------	-----------------

The MySQL DELETE Statement

The DELETE statement is used to delete existing records in a table.

DELETE Syntax

DELETE FROM *table_name* WHERE *condition*;

Note: Be careful when deleting records in a table! Notice the WHERE clause in the DELETE statement. The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

Demo Database

Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
4	Around the Horn	Thomas Hardy	120 Hanover Sq.
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8

ADVERTISEMENT

SQL DELETE Example

The following SQL statement deletes the customer "Alfreds Futterkiste" from the "Customers" table:

Example

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

The "Customers" table will now look like this:

CustomerID	CustomerName	ContactName	Address
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312
4	Around the Horn	Thomas Hardy	120 Hanover Sq.

Delete All Records

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

```
DELETE FROM table_name;
```

The following SQL statement deletes all rows in the "Customers" table, without deleting the table:

Example

```
DELETE FROM Customers;
```

Test Yourself With Exercises

Exercise:

Delete all the records from the Customers table where the Country value is 'Norway'.

Customers

```
Country = 'Norway';
```

Joomla is an open source Content Management System (CMS), which is used to build websites and online applications. It is free and extendable which is separated into front-end templates and back-end templates (administrator). Joomla is developed using PHP, Object Oriented Programming, software design patterns and MySQL (used for storing the data). This tutorial will teach you the basics of Joomla using which you can create websites with ease. The tutorial is divided into sections such as Joomla Basics, Joomla Menus, Joomla Modules, Joomla Global Settings, and Joomla Advanced. Each of these sections contain related topics with screenshots explaining the Joomla admin screens.

What is Content Management System (CMS)?

The **Content Management System (CMS)** is a software which keeps track of the entire data (such as text, photos, music, document, etc.) which will be available on your website. It helps in editing, publishing and modifying the content of the website.

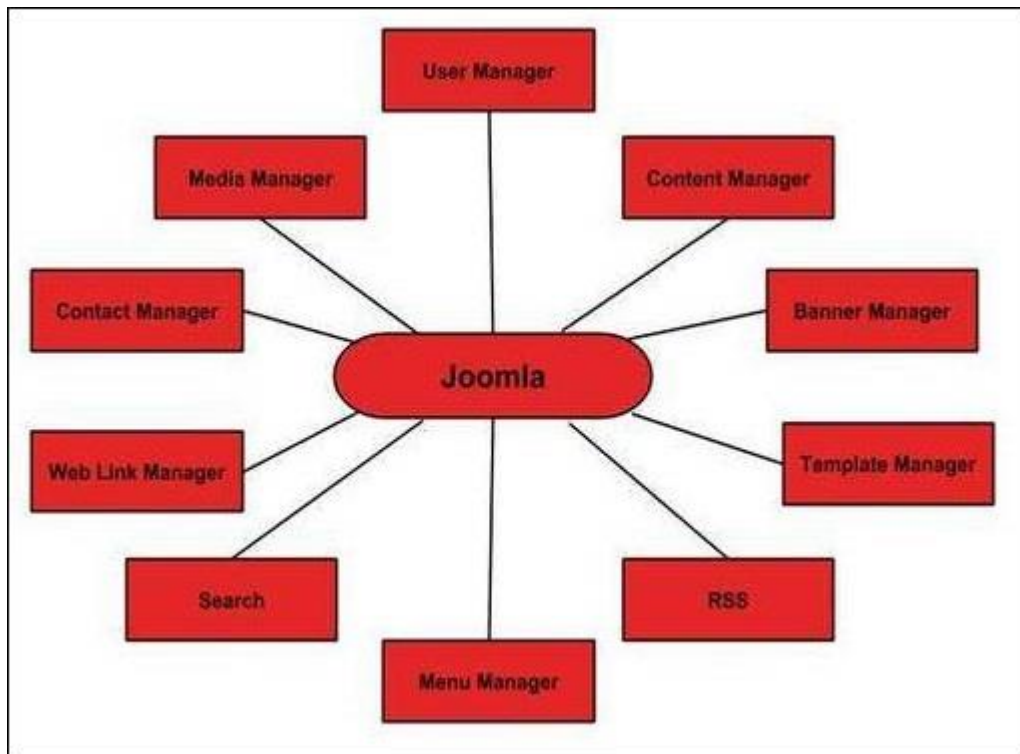
History

Joomla is based on **Mambo** CMS which was developed by an Australian company in 2001 and initially released on *August 17, 2005*. The official version of Joomla 1.0 was released on *September 22, 2005*.

[Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.](#)

Features

Joomla has its own powerful built-in features (core features).



- **User Manager** – It allows managing the user information such as permission to edit, access, publish, create or delete the user, change the password and languages. The main part of the user manager is *Authentication*.
- **Content Manager** – It allows managing the content using WYSIWYG editor to create or edit the content in a very simple way.
- **Banner Manager** – It is used to add or edit the banners on the website.
- **Template Manager** – It manages the designs that are used on the website. The templates can be implemented without changing the content structure within a few seconds.
- **Media Manager** – It is the tool for managing the media files and folder in which you can easily upload, organize and manage your media files into your article editor tool.
- **Contact Manager** – It allows to add contacts, managing the contact information of the particular users.
- **Web Link Manager** – The link resource is provided for user of the site and can be sorted into categories.
- **Search** – It allows users to search the appropriate information on the site. You can use smart indexing, advanced search options, auto suggest searches to make Joomla search best.
- **Menu Manager** – It allows to create menus and menu items and can be managed subsequently. You can put menu in any style and in multiple places.
- **RSS** – It stands for Really Simple syndication which helps your site contents and RSS files to be automatically updated.

Advantages

- It is an open source platform and available for free.
- Joomla is designed to be easy to install and set up even if you're not an advanced user.
- Since Joomla is so easy to use, as a web designer or developer, you can quickly build sites for your clients. With minimal instructions to the clients, clients can easily manage their sites on their own.

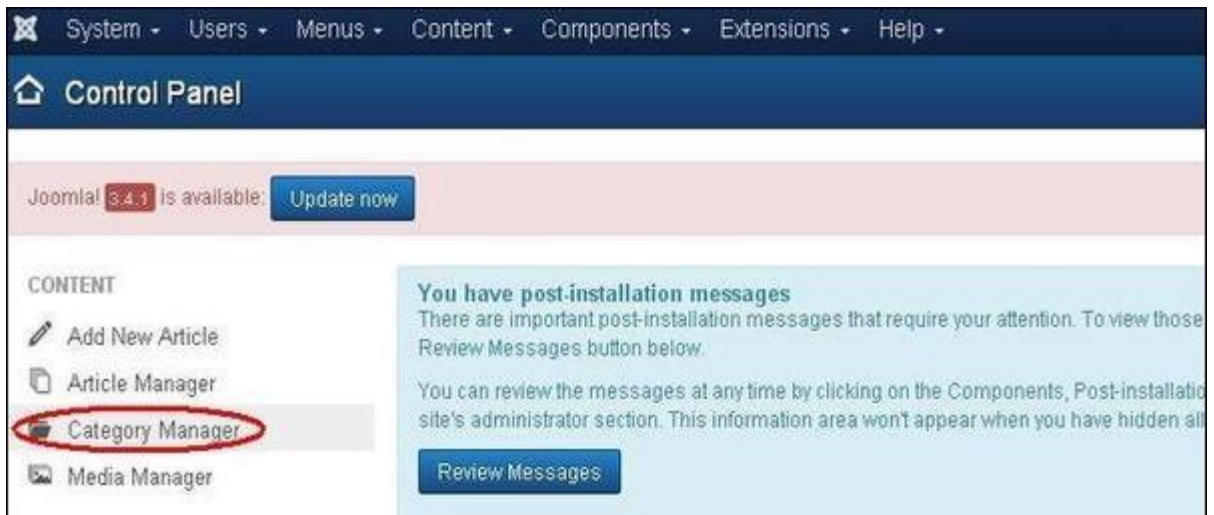
- It is very easy to edit the content as it uses WYSIWYG editor (**What You See Is What You Get** is a user interface that allows the user to directly manipulate the layout of the document without having a layout command).
- It ensures the safety of data content and doesn't allow anyone to edit the data.
- By default, Joomla is compatible with all browsers.
- The templates are very flexible to use.
- Media files can be uploaded easily in the article editor tool.
- Provides easy menu creation tool.

Disadvantages

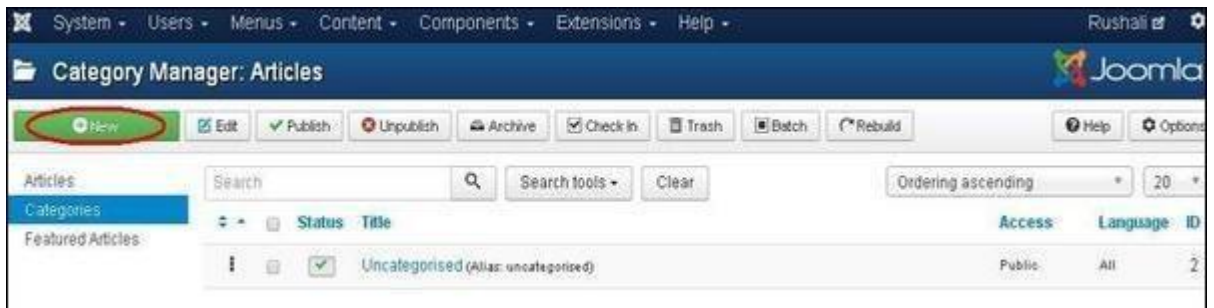
- It gives compatibility problem while installing several modules, extensions and plugins simultaneously.
- Plugins and modules are not free in Joomla.
- Development is too difficult to handle when you want to change the layout.
- Joomla is not much SEO (Search Engine Optimization) friendly.
- It makes website heavy to load and run.

Real World Examples of What Joomla Can Create?

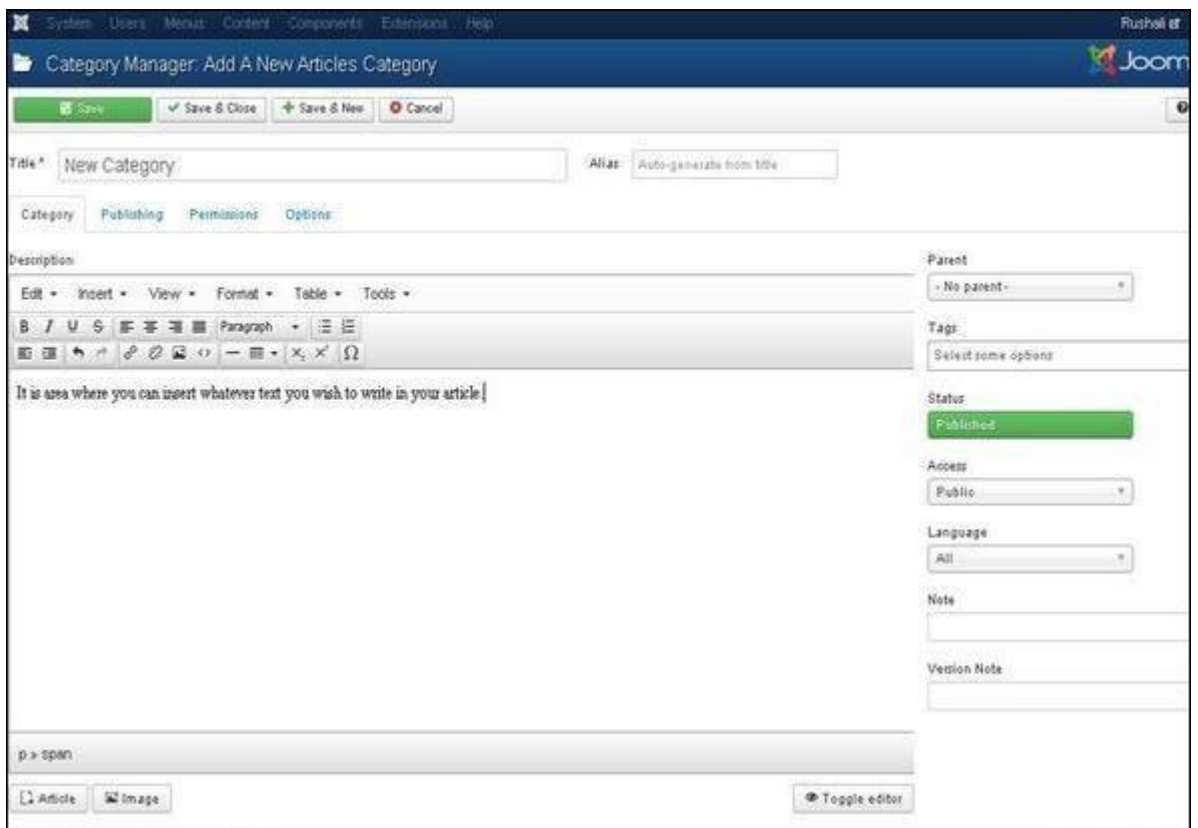
- Corporate web sites or portals
- Corporate intranets and extranets
- Online magazines, newspapers, and publications
- E-commerce and online reservations
- Government applications
- Small business web sites
- Non-profit and organizational web sites
- Community-based portals
- School and religious web sites
- Personal or family homepages
- Category Management
- Following are the simple steps to add new category manager in Joomla.
- **Step 1** – Click on **Category Manager** in Control Panel as shown below.



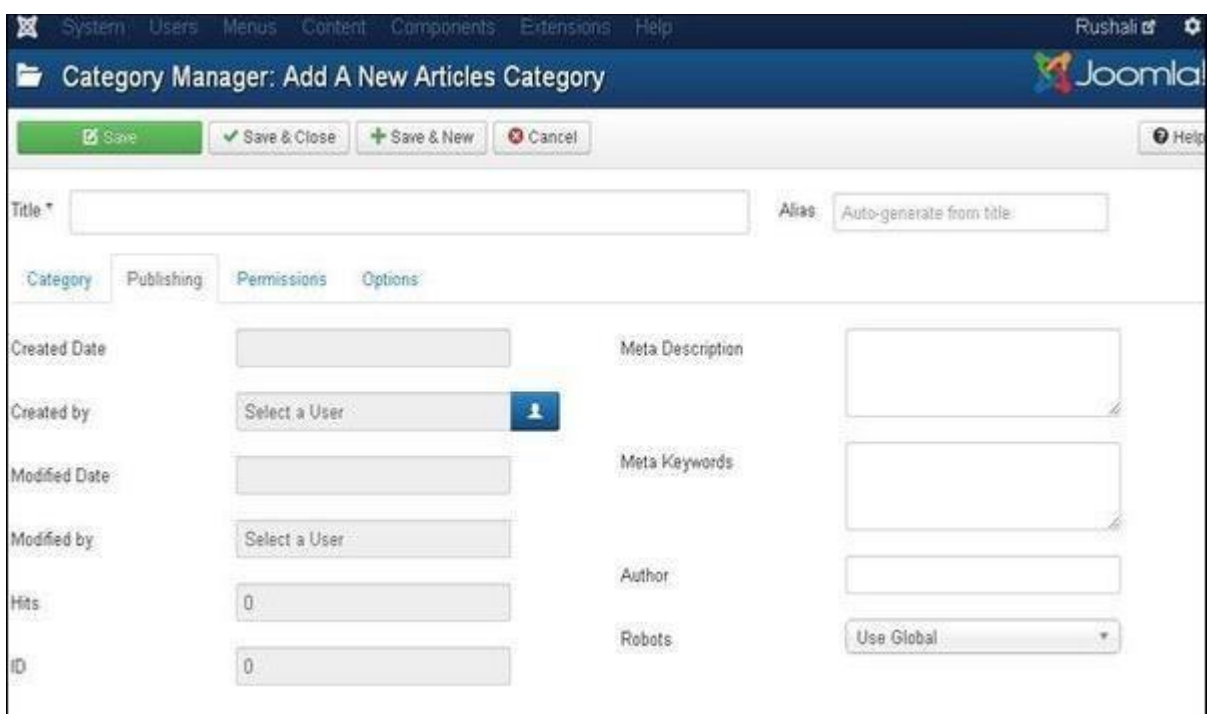
-
- **Step 2** – After clicking on a **Category Manager**, you will get the following screen.



-
- **Step 3** – Click on **New** button in the above screen. Following is the basic editor page we get for category manager. This editor is explained in detail in the article [Joomla - Adding Content](#). Different tabs are present on this page. By default, the **Category** tab is displayed.



- **Step 4** – In **Publishing** tab, it gives the information of Joomla category.



- Following are the details of the fields present in the **Publishing** tab.
- **Created Date** – It is the date the category was created.
- **Created by** – It is the name of the user who has created the category.
- **Modified Date** – It is the date the category was modified.

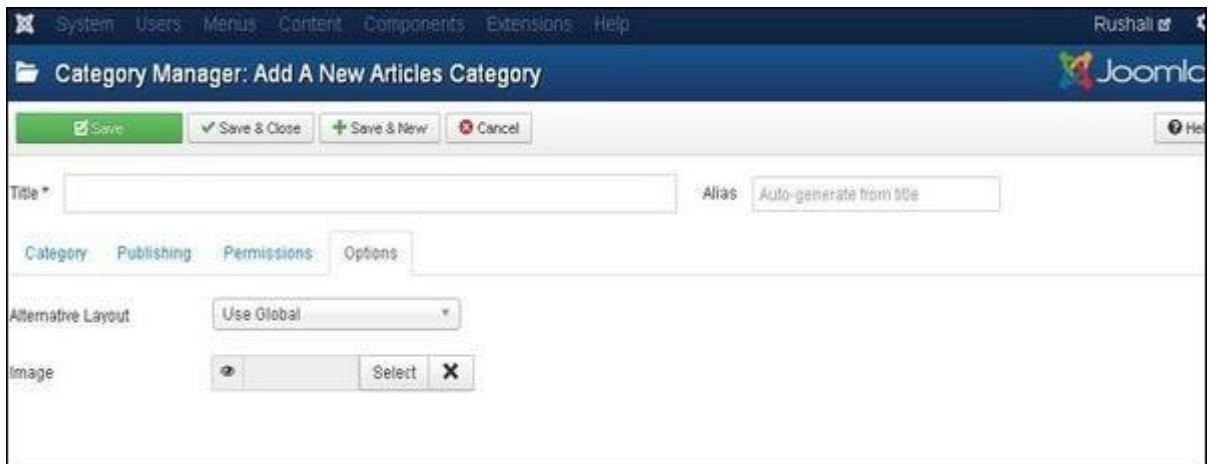
- **Modified by** – It is the name of the user who has modified the category.
- **Hits** – Specifies the number of times the item was viewed.
- **ID** – It is the unique identification number assigned to the items automatically.
- **Meta Description** – It is used to write an explanation of the contents of web pages.
- **Meta Keywords** – Define keywords for search engines.
- **Author** – Specifies the name of the author within the metadata.
- **Robots** – It is the instruction given to the robots that browse the page using commands such as *Use Global, Index, Follow, No index, No follow*.
- **Step 5** – In **Permission** tab, you can set the permission for different user groups for that category.

Public	Action	Select New Setting ¹	Calculated Setting ²
- Guest	Create	Inherited	Not Allowed
- Manager	Delete	Inherited	Not Allowed
-- Administrator	Edit	Inherited	Not Allowed
- Registered	Edit State	Inherited	Not Allowed
-- Author	Edit Own	Inherited	Not Allowed
--- Editor			
---- Publisher			
- Super Users			

1. If you change the setting, it will apply to this article. Note that:
Inherited means that the permissions from global configuration, parent group and category will be used.
Denied means that no matter what the global configuration, parent group or category settings are, the group being edited cannot take this action on this article.
Allowed means that the group being edited will be able to take this action for this article (but if this is in conflict with the global configuration, parent group or category it will have no impact; a conflict will be indicated by *Not Allowed (Locked)* under Calculated Settings).

2. If you select a new setting, click **Save** to refresh the calculated settings.

- **Create** – It allows the users in a group to create the categories.
- **Delete** – It allows the users in a group to delete the categories present in the extension.
- **Edit** – It allows the users in a group to edit the categories present in the extension.
- **Edit State** – It allows the users in a group to change the state of categories present in the extension.
- **Edit Own** – It allows editing the categories which is created by own.
- **Step 6** – In the **Options** tab, you can choose different layouts for your website.



- **Alternative Layout** – It uses an alternative layout from the supplied component view such as *Use Global, Blog* and *List*.
- **Image**– It selects the image to be displayed.
- **Toolbar**
- Following are the toolbar options in the Category Management.
- **Save** – Saves your category.
- **Save & Close** – Saves the category and closes the current screen.
- **Save & New** – Saves the categories and opens a new create category screen.
- **Cancel** – Cancels the created category in Joomla.

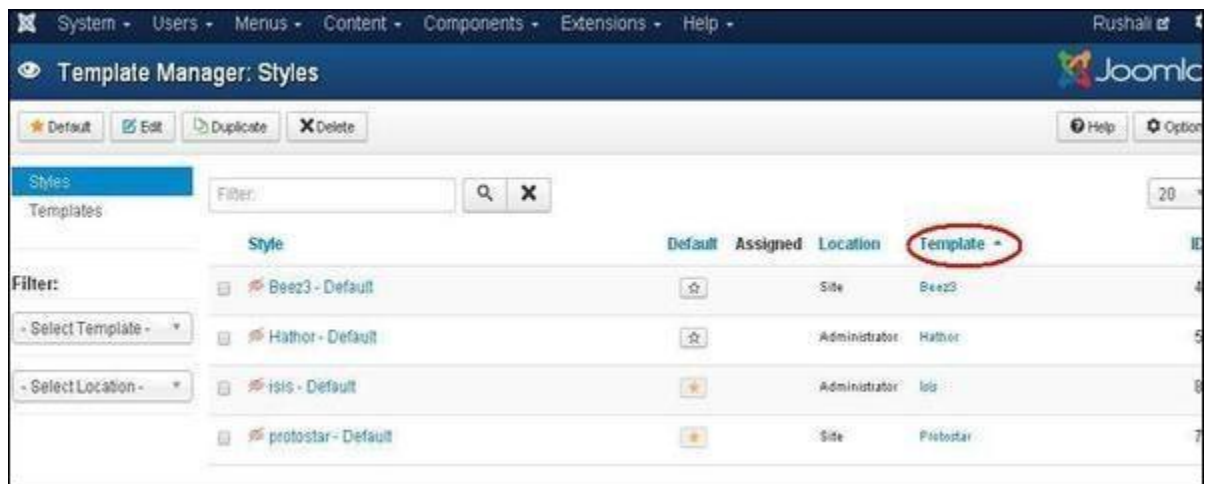
Template Manager

Following are the simple steps to edit the Template Manager in Joomla.

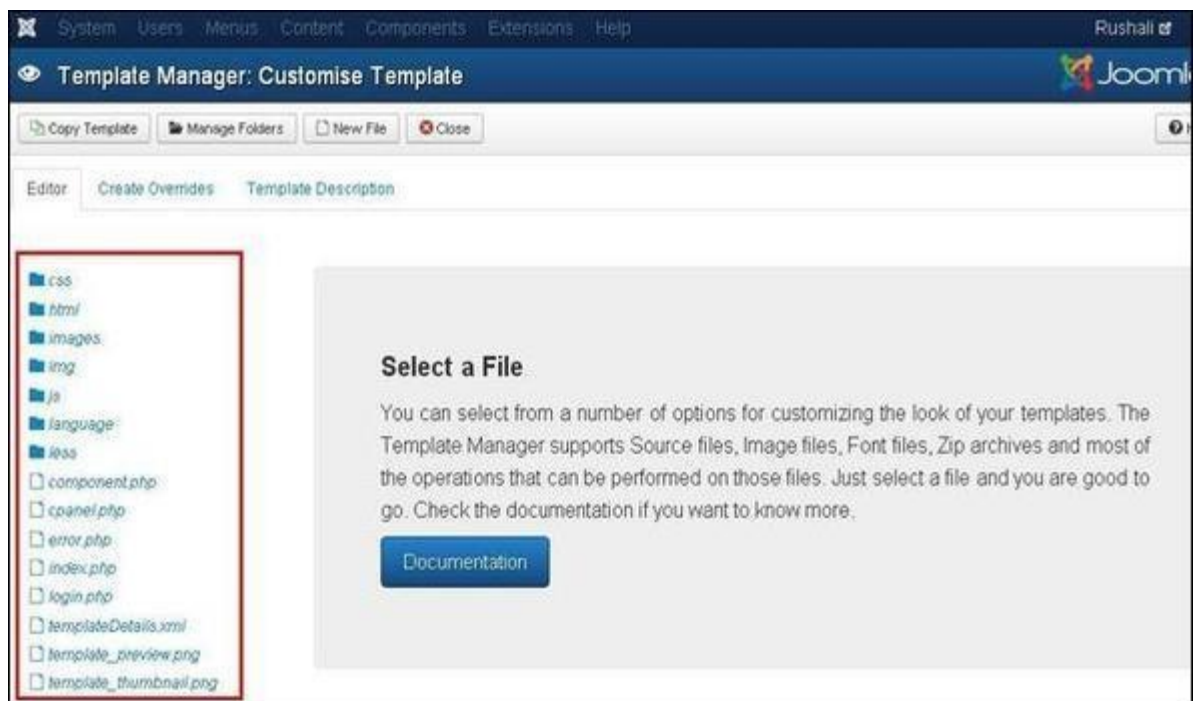
Step 1 – Click **Extensions** → **Template Manager** as shown below.



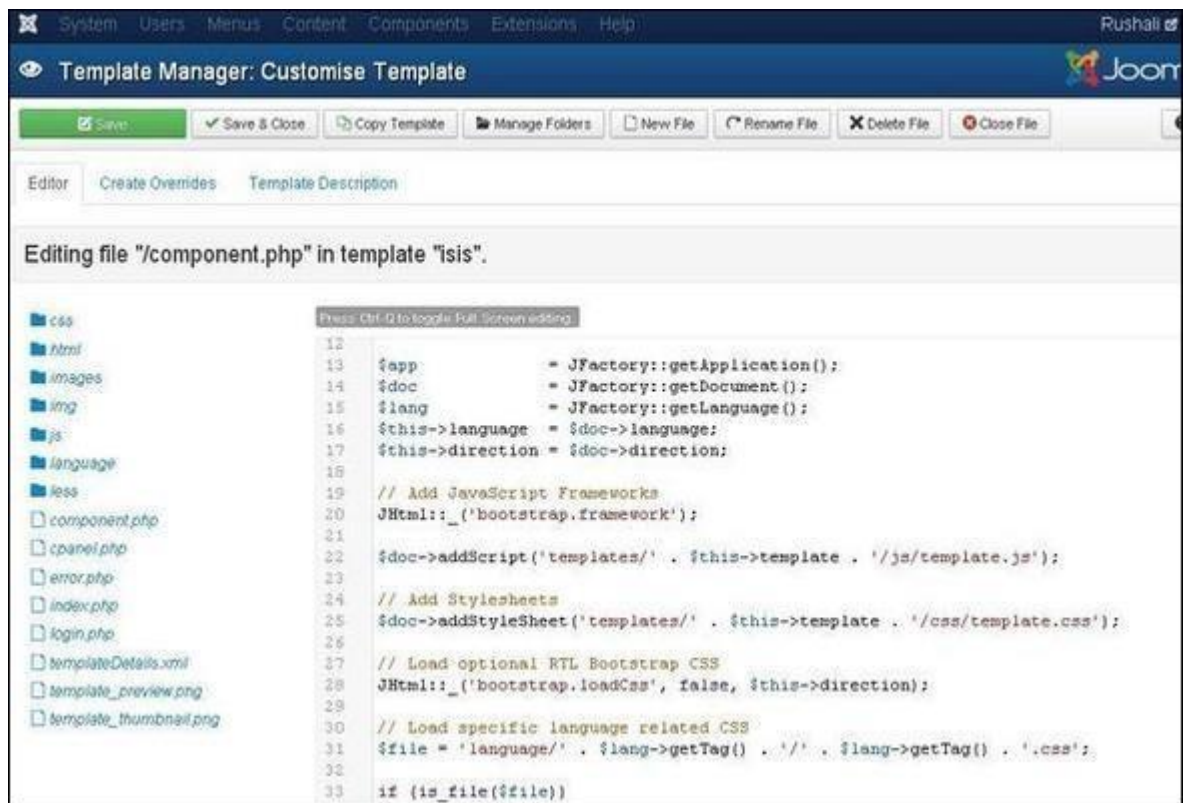
Step 2 – **Template Manager:Styles** page get displayed as shown below. Here, you can click directly on any of the templates listed in **Template** column.



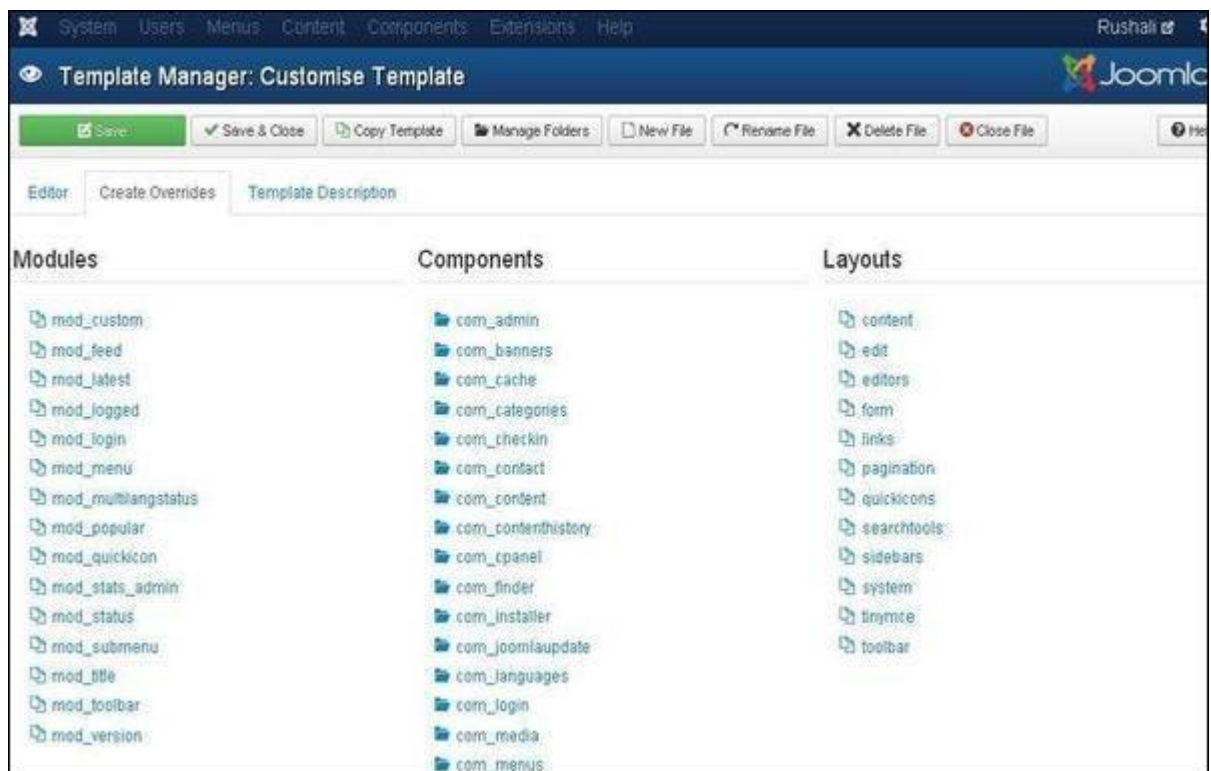
Step 3 – After clicking on any one of the above templates you will get the screen as shown below. In the **Editor** tab, template files are listed on the left side of the page as seen in the screenshot below –



Step 4 – Select any of the files by clicking on it for editing.



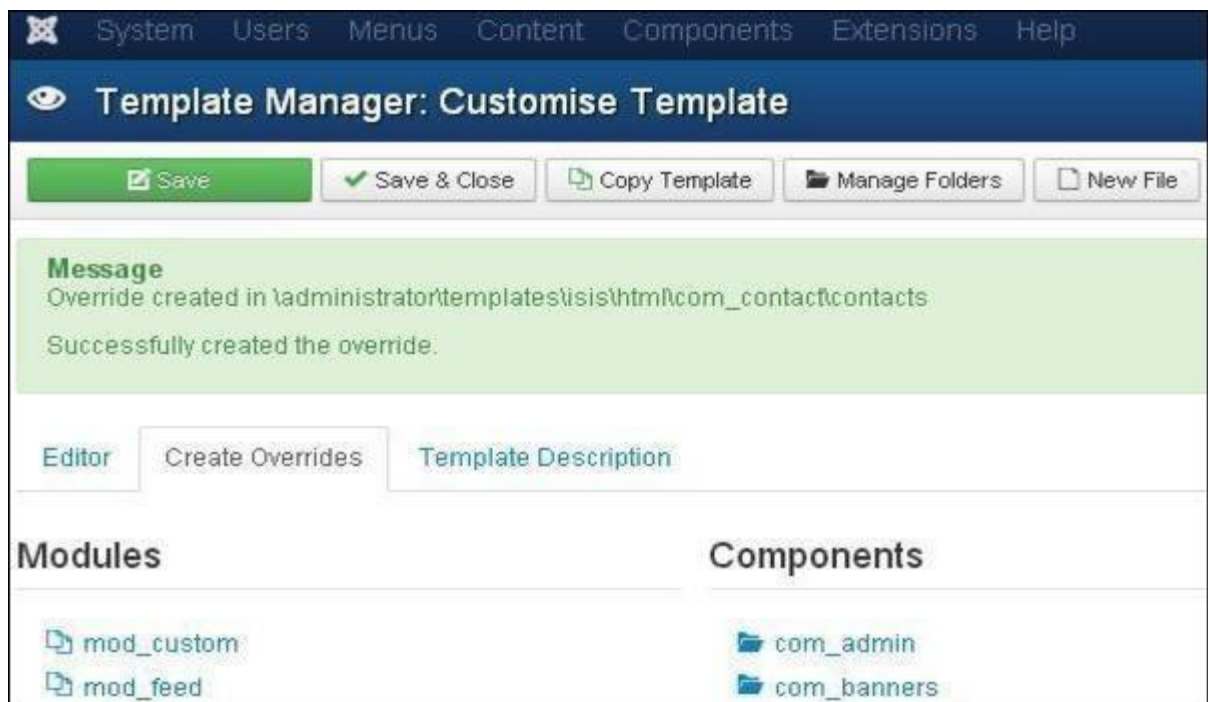
Step 5 – Click on **Create Overrides** tab, you will see the list of **Modules**, **Components**, and **Layouts** present in the template manager as shown below. Here you can easily override the new templates.



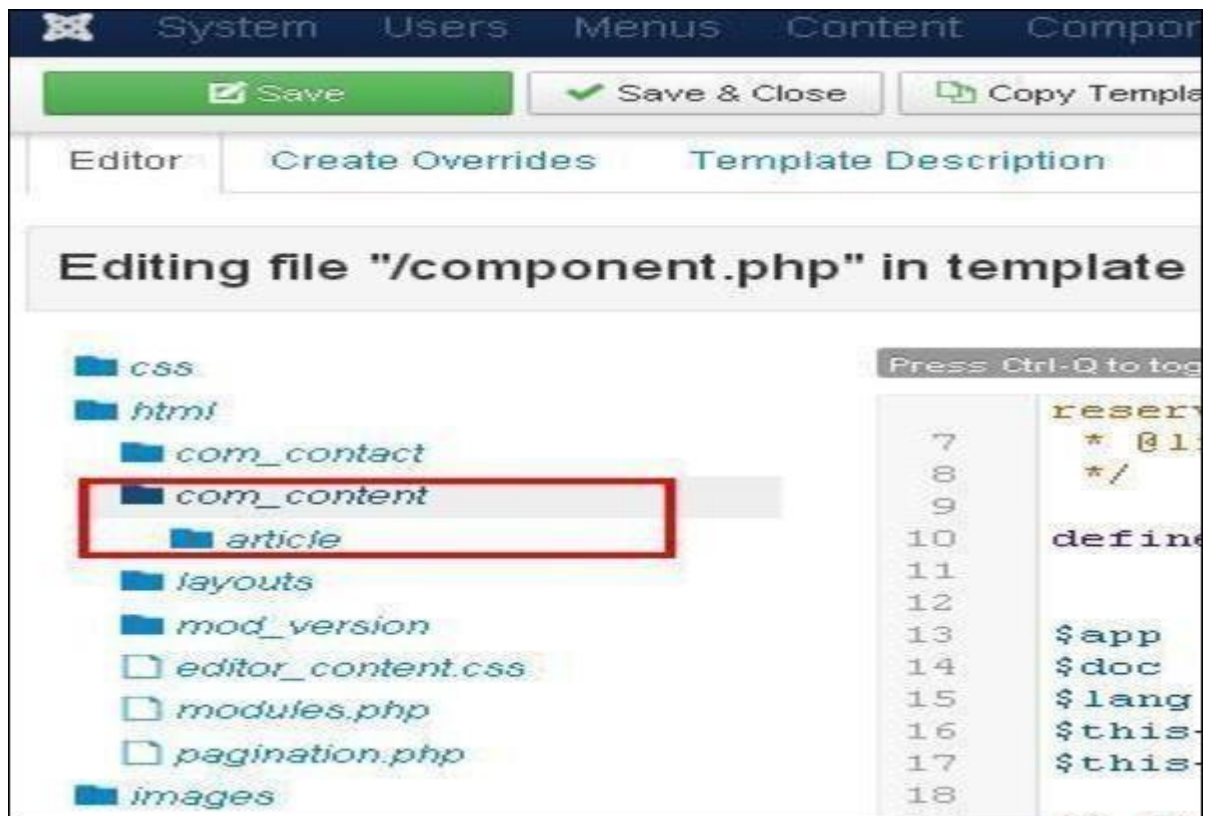
Step 6 – Click on any one component and you will see many sub components present inside it as shown below.



Step 7 – When you click on any one sub component then you will get the following message.



Step 8 – You can go back to **Editor** tab to check if override file is available for editing or not as shown below.



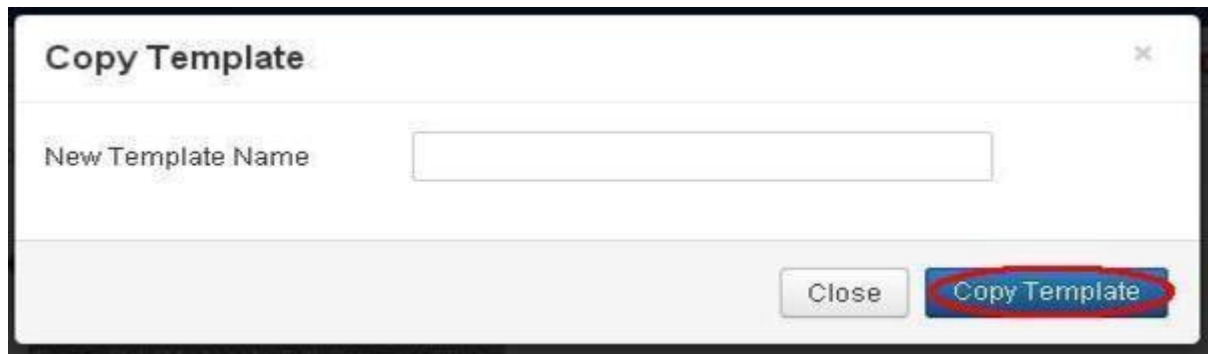
Step 9 – Click on **Template Description** tab, you will see the detailed description of the selected template as shown below.



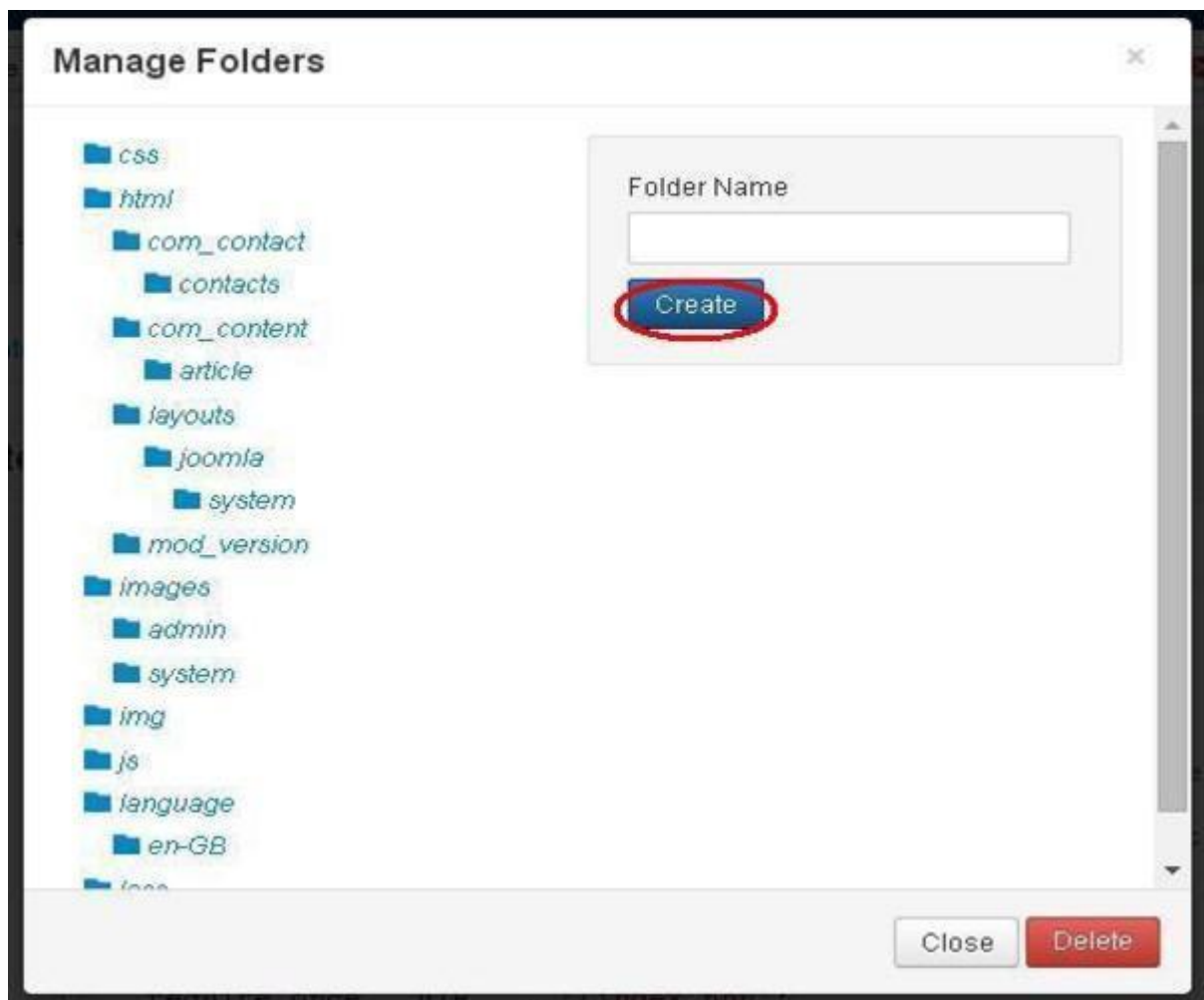
Toolbar

Following are the toolbar options in the Template Manager.

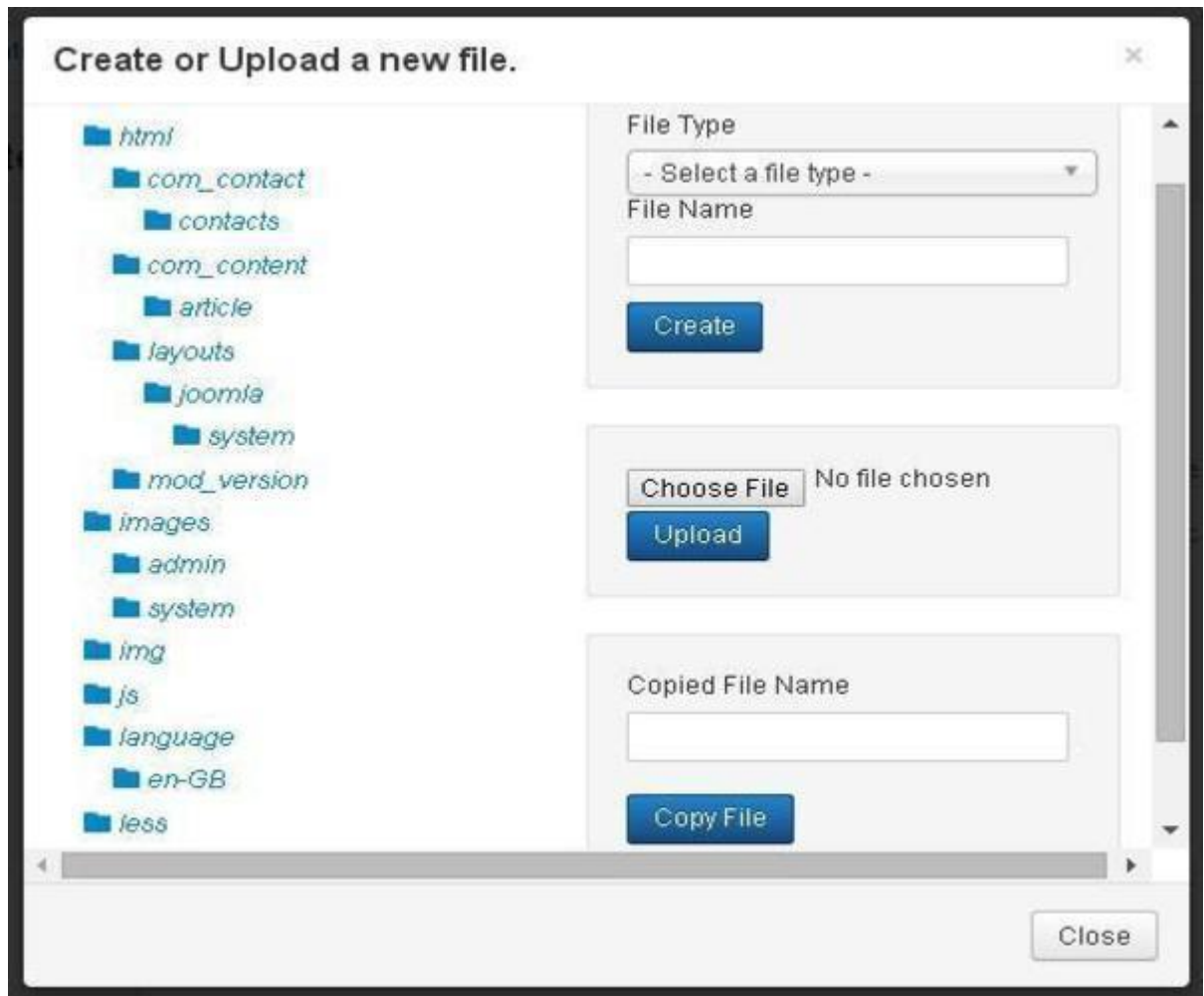
- **Save** – Saves your template manager.
- **Save & Close** – Saves the templates and closes the current screen.
- **Copy Template** – Enter the template name and click on **Copy Template** button. This will copy the template.



- **Manage Folders** – It creates a new folder inside the template. Enter the folder name in the field and click on **Create** button.



- **New File** – You can select the file type by clicking on the dropdown of **File Type** and enter the name of the file and click on **Create** button. This will create a new file. The second block is used to upload a file directly from your system and the last block is used to copy the file.



- **Rename File** – Enter the new name for the file and click on **Rename** button.



- **Delete File** – Deletes the particular file.
- **Close File** – Closes the current file.

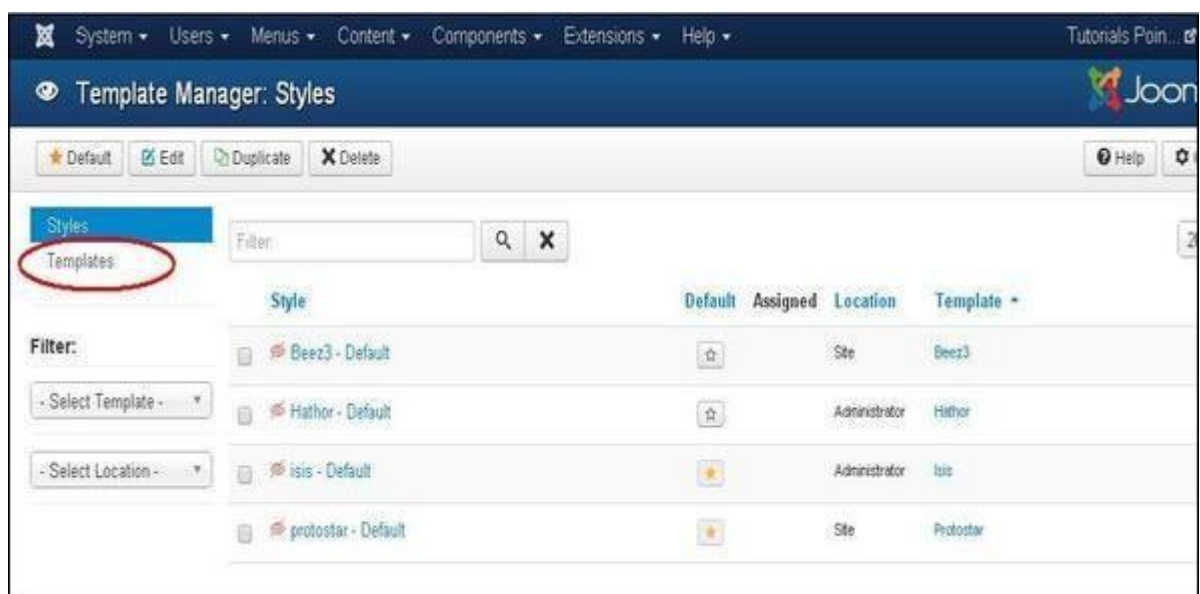
Customize Template

Following are the simple steps to customize the template in Joomla.

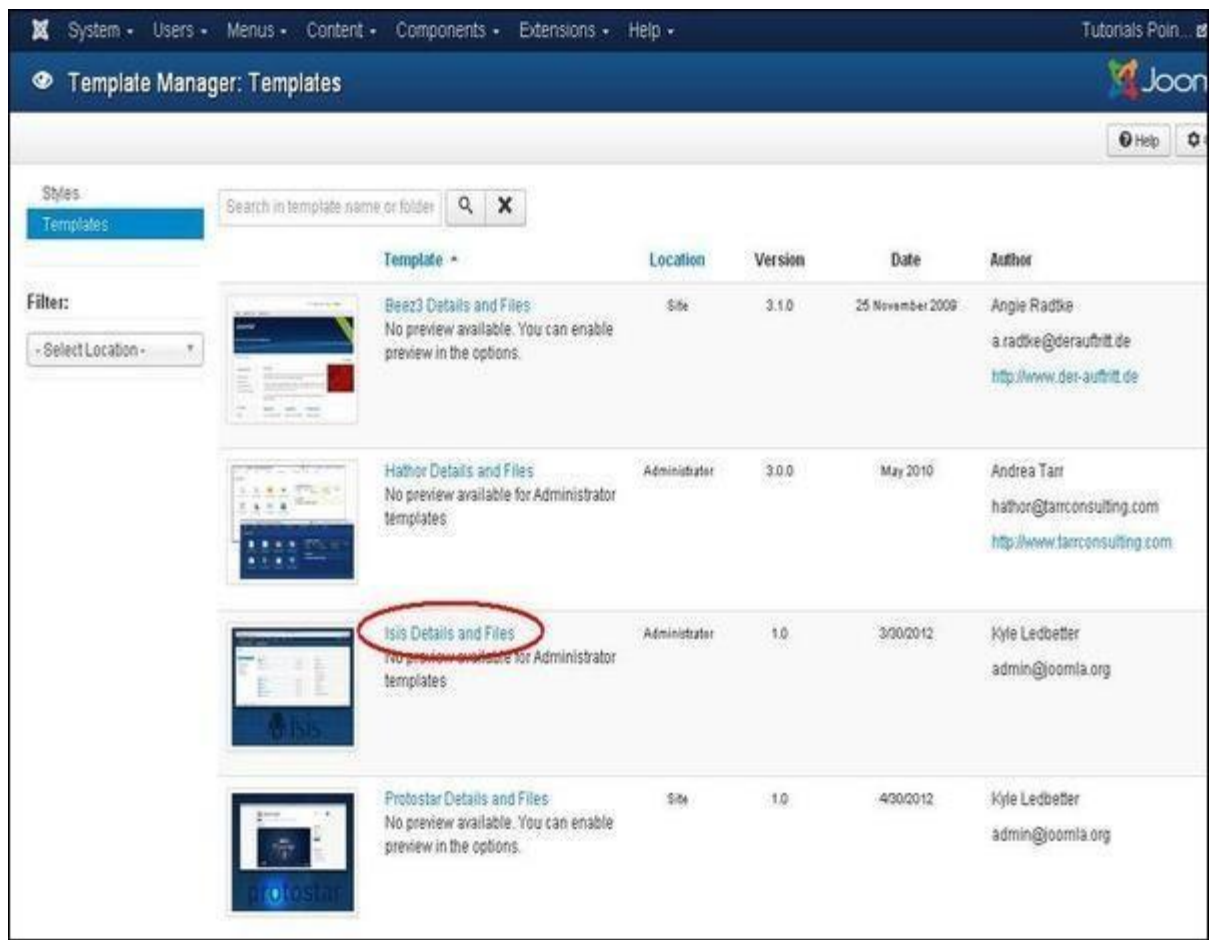
Step 1 – Click **Extensions** → **Template Manager** as shown below.



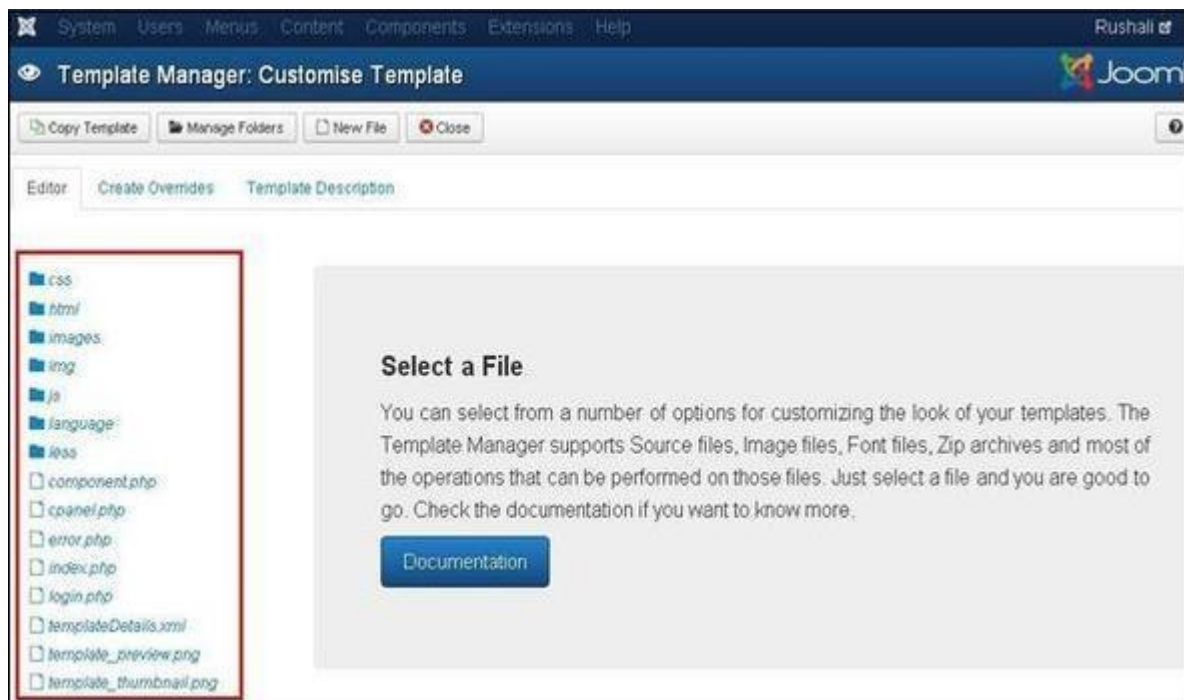
Step 2 – After clicking on **Template Manager**, the **Template Manager: Styles** page will get displayed.



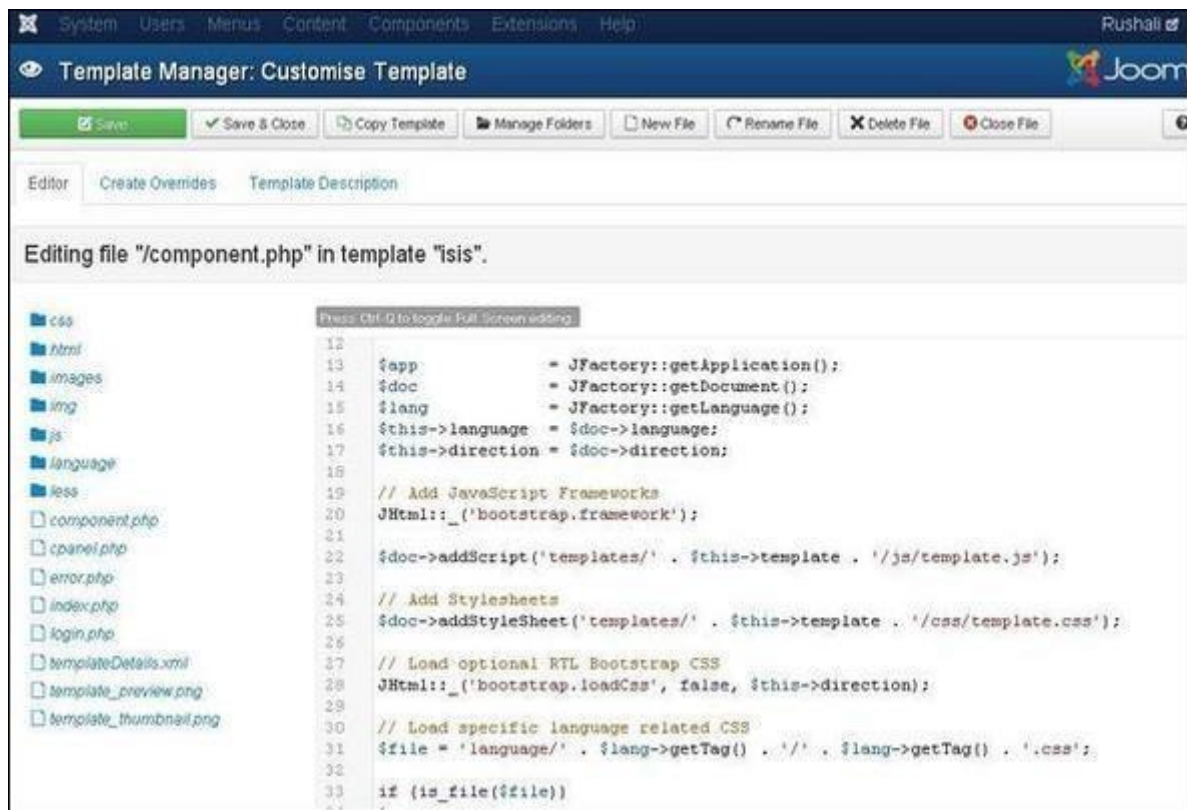
Step 3 – Next, click on **Templates** as shown above, you will get the following screen. Here select any one of the Templates.



Step 4 – After selecting a template you will get the screen as shown below. In the **Editor** tab, template files are listed on the left side of the page as seen below.



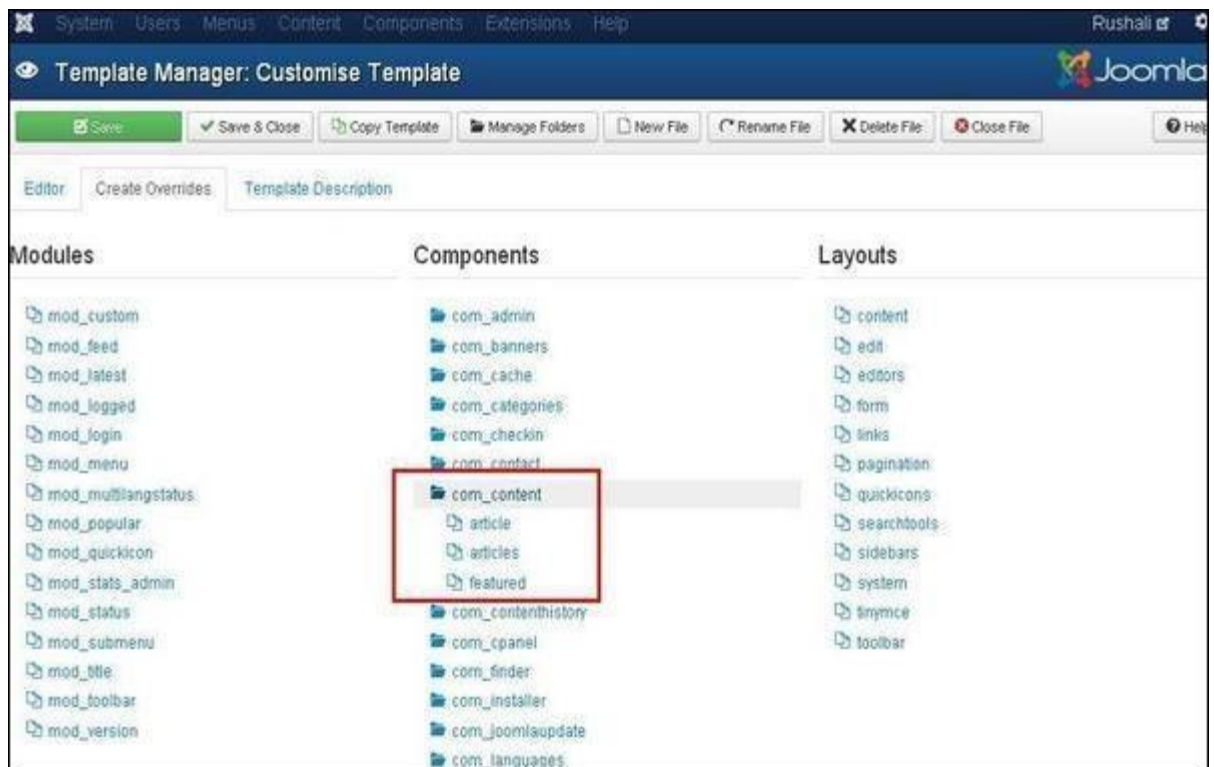
Step 5 – Select any of the files by clicking on it for editing.



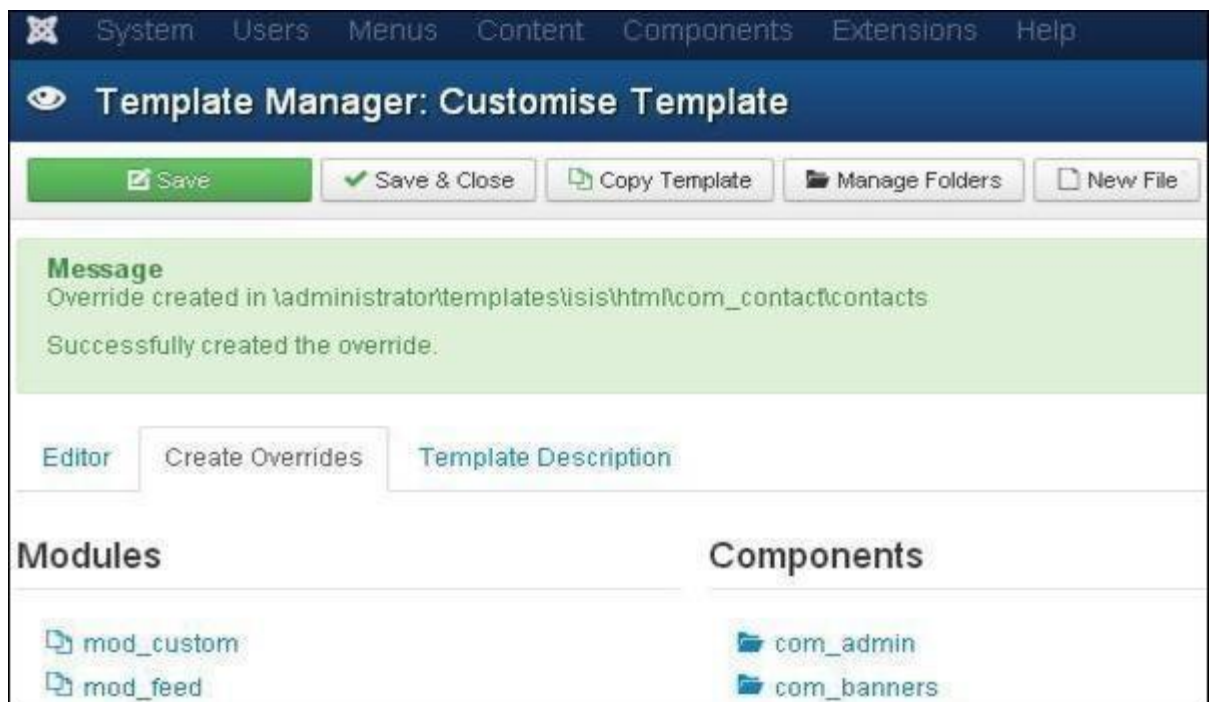
Step 6 – Click on **Create Overrides** tab, you will see the list of **Modules**, **Components**, and **Layouts** present in the template manager as shown below. Here you can easily override the new templates.



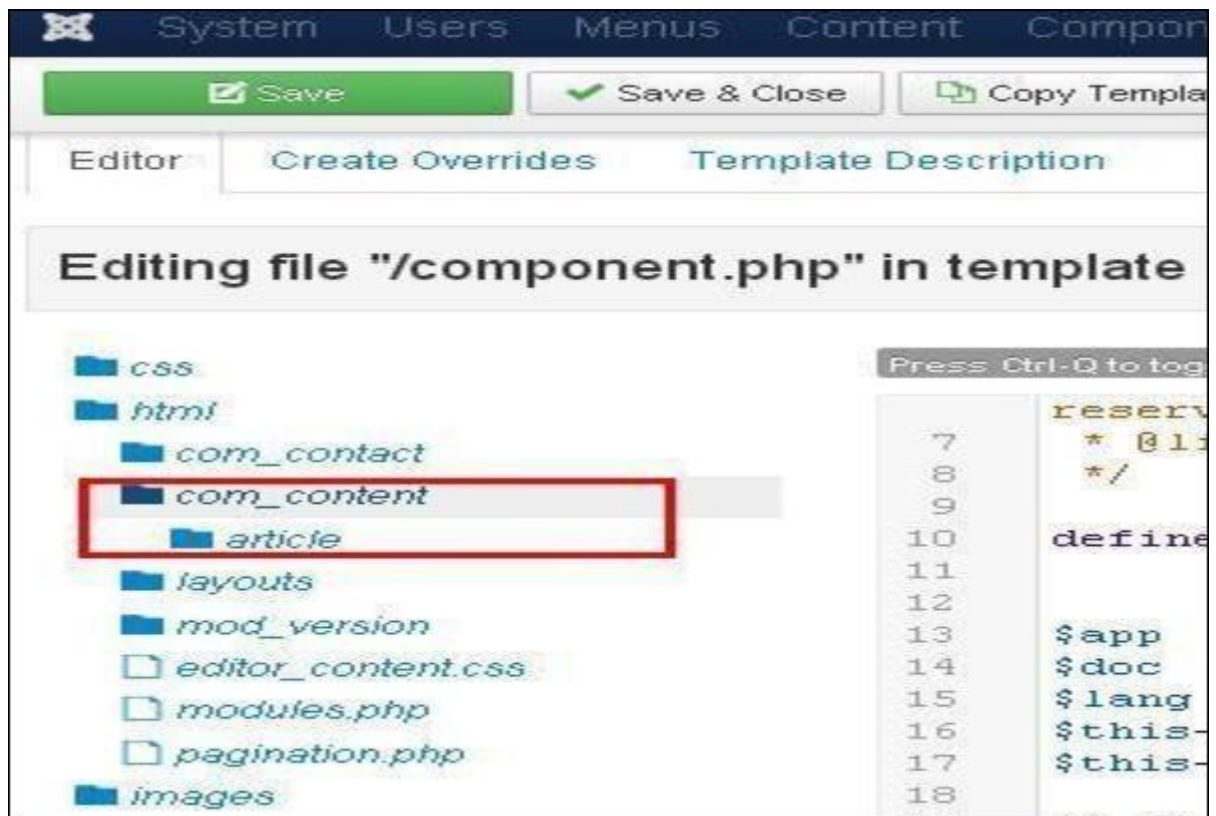
Step 7 – Click on any one component and you will see many sub-components present inside it as shown below.



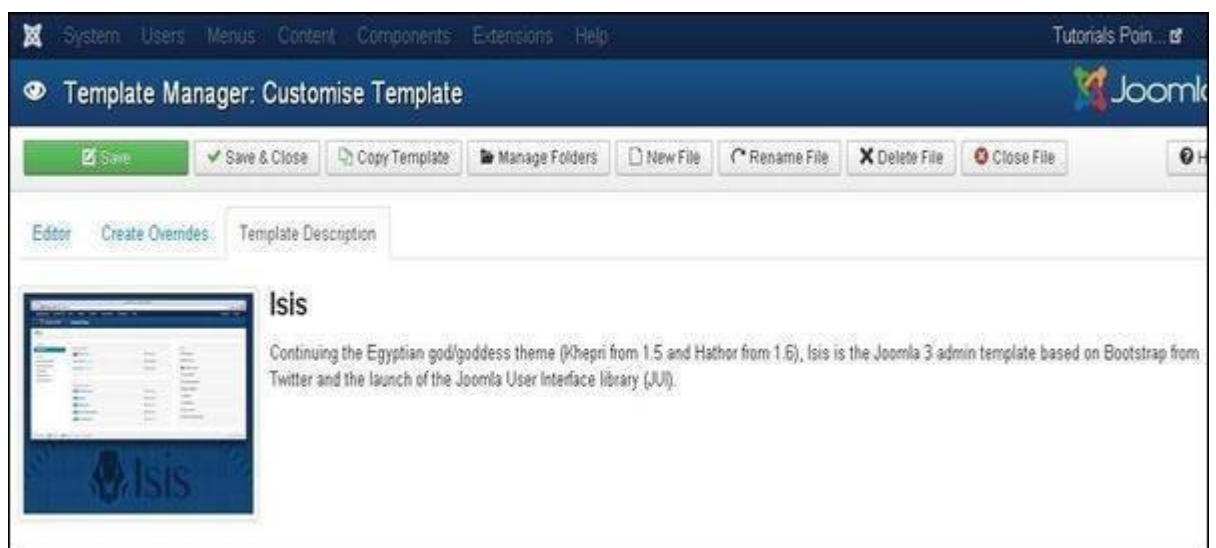
Step 8 – When you click on any one sub-component then you will get the following message.



Step 9 – You can see the overrides file is available for editing or not by going back to the **Editor** tab as shown below.



Step 10 – After clicking on **Template Description** tab, you will see the detail description of the selected template as shown below.



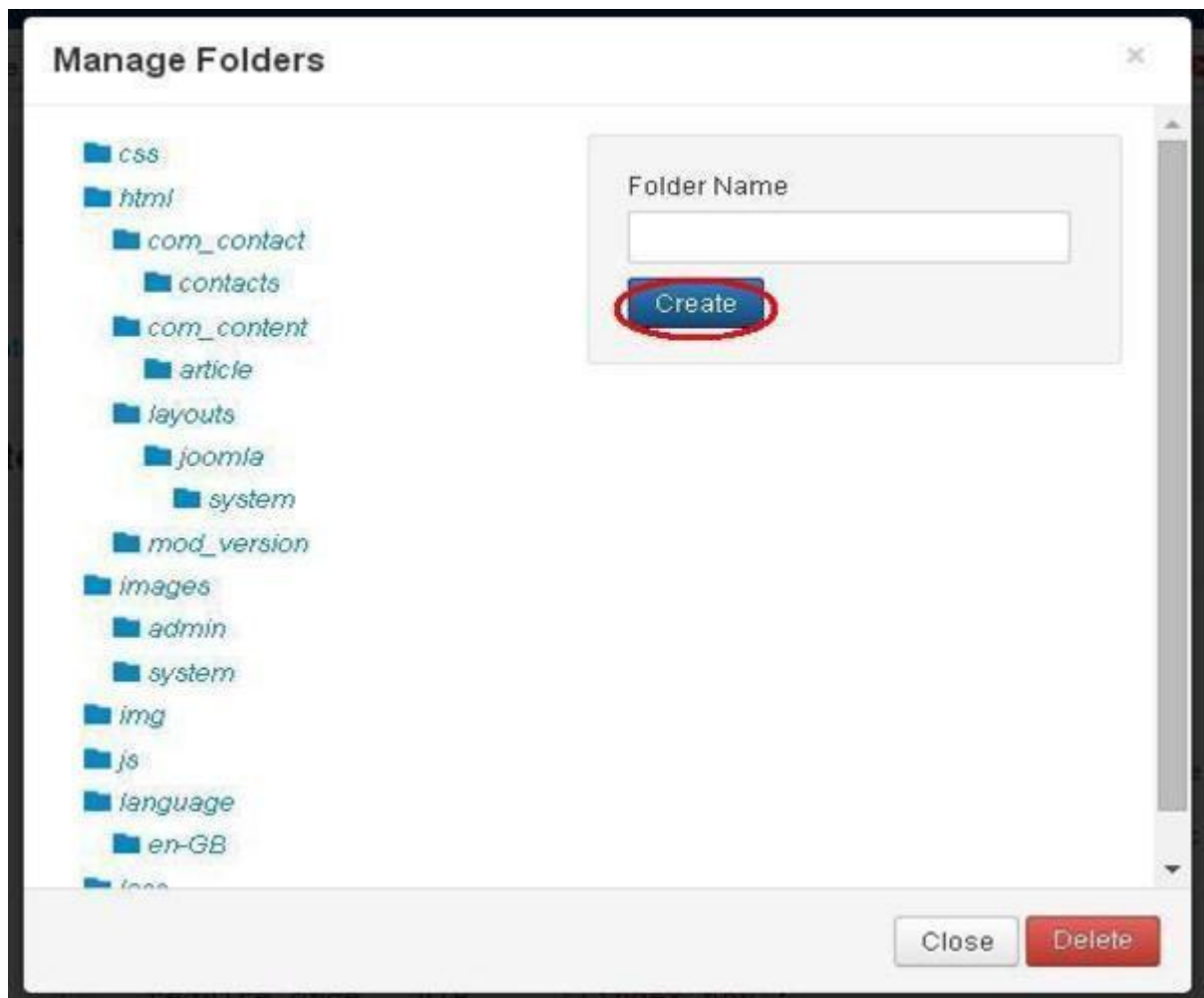
Toolbar

Below we have mentioned details about the fields on toolbar in the Template Manager.

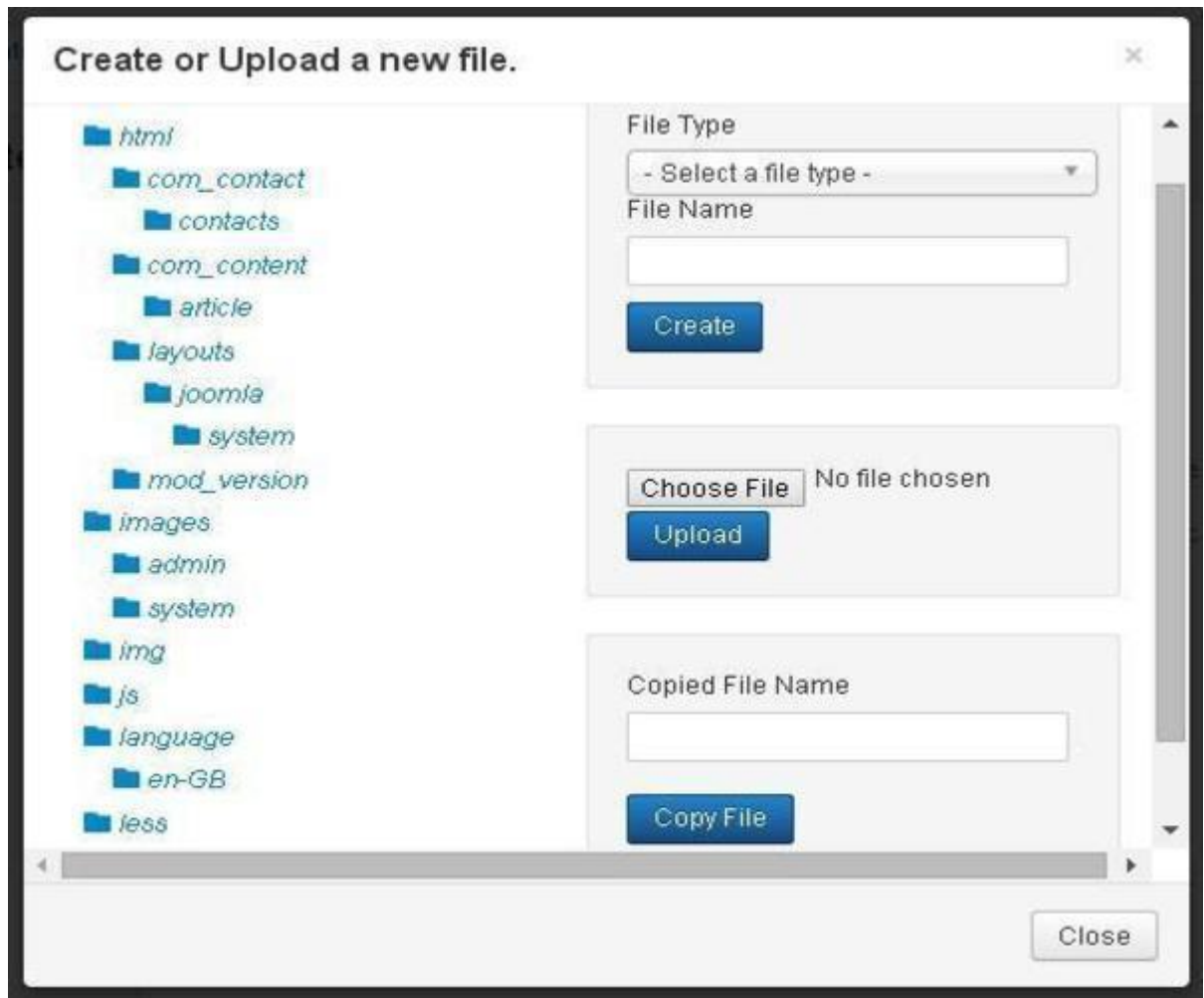
- **Save** – Saves your template manager.
- **Save & Close** – Saves the templates and closes the current screen.
- **Copy Template** – Enter the template name and click on **Copy Template** button which will help you to copy the template.



- **Manage Folders** – It creates a new folder inside the template. Write the folder name in the field and click on **Create** button.



- **New File** – You can select the file type by clicking on the dropdown of **File Type** and write the name of the file and click on **Create** button which will create a new file. The second block is used to upload a file directly from your system and the last block is used to copy the file.



- **Rename File** – Enter the new name for the file and click on **Rename** button.



- **Delete File** – Deletes the particular file.
- **Close File** – Closes the current file.