



www.cocobasic.com
<https://cocobasic.ticksy.com/>
cocobasicthemes@gmail.com

Template Manual

Seppo HTML - Demo 2

Thank You :)

Thanks for purchasing the template and for supporting our work. Seppo is our new fancy template. Responsive, clean and professional look with sleek design will satisfy all your needs.

You will need basic HTML, CSS, jQuery skills to edit Seppo HTML. We will try to help you out with this documentation. If we fail here, you can always contact us on <https://cocobasic.ticksy.com/> and our developers will try to solve your issue.

How To Install?

Considering that Seppo is HTML template, with fully functional email/subscribe notification mechanism for sending messages via contact form, all you need to do is to upload all template files via your FTP account to your web server (for uploading files you can use FTP client like **FileZilla**) and just to modify code in **config.php** file. To edit the **config.php** you can use any text editor (for example **NotePad++**)

These are parameters that you need to modify:

- Your email addresses where you want to receive emails filled with information from contact form.

In **config.php** file, you will see this pattern:

```
$email = "yourmail@here.com";    //<-- Your email
```

You need to modify only the value on the right side.

Do not remove any quote or double quote signs while doing this.

Folder Structure

TEMPLATE

```
|  
|  
|--- css  
|    |--- all css files (except main style.css file)  
|  
|  
|--- fonts  
|    |--- Font Awesome files  
|  
|  
|--- images  
|    |--- all demo images  
|  
|  
|--- js  
|    |--- all js files  
|  
|  
|--- php  
|    |--- sendMail.php  
|  
|  
|--- all .html files  
|  
|  
|--- config.php  
|  
|  
|--- style.css (main css files)
```

Which file to edit?

Of course, you can edit any of the files from above (if you know what you are doing) also you can add (include) some additional JS/CSS files for some additional features, but if your skills are limited and you want just to replace the demo/dummy content with your own, you need:

- To edit **.html** files (using some text editor, for example **NotePad++**) and to change the demo content (text) with your own
- If you don't need some part from our demo content, I am suggesting to use comments tags in html files to disable that part.
More about HTML comments tags - <https://html.com/tags/comment-tag/>
- To override the images from **images** folder with your own
- Change only the main **style.css** file to set your own colors, font size, etc...
- Edit **config.php** file and set your own mail address

HTML Structure

Seppo (Demo 2) consist several HTML files

- index.html - front page
- single.html
- blog.html
- portfolio-1.html
- portfolio-2.html

To edit the **html** file you can use any text editor (for example **NotePad++**).

General HTML structure

Each .html file has **head** tag which looks like this:

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

<head>

    <title>Seppo - Demo 2</title>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="description" content="Template by CocoBasic">

    <meta name="keywords" content="HTML, CSS, JavaScript, PHP">

    <meta name="author" content="CocoBasic">

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

    <link rel="shortcut icon" href="images/favicon.ico">

    <link href='https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700,800'
rel='stylesheet' type='text/css'>

    <link rel="stylesheet" type="text/css" href='style.css'>

</head>
```

Here you need to set your own **title**, **description**, **keywords** and **author**. Also, don't forget to create your own **favicon** and to replace the default one (it is placed in the images folder). If you need to change/add Google fonts, you need to edit this line:

```
<link href='https://fonts.googleapis.com/css?family=Montserrat:400,500,600,700,800' rel='stylesheet'
type='text/css'>
```

Also, each .html file consist the **menu** part and here you will find code like this:


```
<nav id="header-main-menu">
  <ul class="main-menu sm sm-clean">
    <li>
      <a href="#services">Services</a>
    </li>
    <li>
      <a href="#about">About</a>
    </li>
    <li>
      <a href="#portfolio">Portfolio</a>
    </li>
    <li>
      <a href="#pricing">Pricing</a>
    </li>
    <li>
      <a href="#testimonial">Testimonial</a>
    </li>
    <li>
      <a href="#news">Blog</a>
    </li>
    <li>
      <a href="#skills">Skills</a>
    </li>
    <li>
      <a href="#time">Timeline</a>
    </li>
    <li>
      <a href="#contact">Contact</a>
    </li>
  </ul>
</nav>
```

And the last part which is the same for all html files are JavaScripts.

At the end of each .html file, there is a part of code where we are including all **JavaScripts** files which we are using in our template. This part looks like this:

```
<!--Load JavaScript-->
```

```
<script src="js/jquery.js"></script>
```

```
<script src='js/imagesloaded.pkgd.js'></script>
```

```
<script src='js/jquery.prettyPhoto.js'></script>
```

```
<script src='js/isotope.pkgd.js'></script>
```

```
<script src='js/jquery.smartmenus.min.js'></script>
```

```
<script src='js/owl.carousel.min.js'></script>
```

```
<script src='js/circle-progress.min.js'></script>
```

```
<script src='js/main.js'></script>
```

Front Page index.html

On Index page are placed 10 sections:

Home, Services, About, Portfolio, Pricing, Testimonial, Blog, Skills, Timeline and Contact. Each section is separated and start with code like this:

```
<!-- Services Section -->

<div id="services" class="section services">
```

Also, for each of these sections you will find in main **style.css** file appropriate css code, like this (example for Home section):

```
/* =====

3. Home Section CSS

===== */

#home {

    background-image: url(images/main_img_01.jpg);

    background-repeat: no-repeat;

    background-size: cover;

    background-position: center center;

    min-height: 80vh;

    display: flex;

    align-items: center;

    background-attachment: fixed;

}
```

Now, for each section we will explain just some part of the code which is specific for it.

Milestones Section

Code for milestones numbers looks like this:

```
<ul class="milestone-holder">
    <li>
        <p><span class="milestone-num milestone-num-k" data-count="57">1</span></p>
        <p class="milestone-text">Downloads</p>
    </li>
    <li>
        <p><span class="milestone-num milestone-num-k" data-count="14">1</span></p>
        <p class="milestone-text">Campaigns</p>
    </li>
</ul>
```

Here you need to change just the number - value in **data-count**

Portfolio Section

Portfolio filter

On Portfolio section you will find code for category filtering:

```
<div class="category-filter-list button-group filters-button-group">

    <div class="button is-checked" data-filter="*">All</div>

    <div class="button" data-filter=".text">Text</div>

    <div class="button" data-filter=".video">Video</div>

    <div class="button" data-filter=".image">Image</div>

</div>
```

First filter value is “All” and it is important to set **data-filter="*"** to show all items. For all other categories, the **data-filter** has a class name as value. This class should be also included in html code for portfolio item, like this:

```
<div id="p-item-1" class="grid-item element-item p-one text">
```

This means the item will be visible when you select **text** in the category filter. You can place one item in two or more categories, just add class like this:

```
<div id="p-item-1" class="grid-item element-item p-one-third text interior abstract super-mario">
```

Portfolio Items Links - there are 3 different ways how those items are linked:

- Item links to **portfolio-x.html** (ajax load)
- Item links to Pop-Up image (Lightbox)
- Items links to Pop-Up video (Lightbox)

- Ajax load - Items links to portfolio-x.html

Portfolio items 1 and 2 are using ajax to load the content inside the index.html file. When you click on an image, the code will load a portfolio html file and will get just the content part from that file and show it directly on the front/one page.

Here is an example for portfolio item which is loading content via ajax:

```
<div id="p-item-1" class="grid-item element-item p-one text">
  <a class="item-link ajax-portfolio" href="portfolio-1.html" data-id="1">
    
    <div class="portfolio-text-holder">
      <div class="portfolio-text-wrapper">
        <p class="portfolio-text">Home</p>
        <p class="portfolio-cat">Text</p>
      </div>
    </div>
  </a>
</div>
```

Portfolio item content is loaded in the page using Ajax. The important part here is this 2 lines of code:

```
<div id="p-item-1" class="grid-item element-item p-one text">  
  <a class="item-link ajax-portfolio" href="portfolio-1.html" data-id="1">
```

Class "**p-one**" means that the portfolio item (image) will be 100% width. There is an option to set class "**p-one-half**" and width of the item will be 50%;

Class "**ajax-portfolio**" must be included if you want to have in page/ajax load. If you remove this class, the link will work on "regular way" and will open the page which you have set in "**href**".

Another important part is to set a unique **data-id** number of each item like: **data-id="1"** , **data-id="2"** , **data-id="3"**

Also, the **id="p-item-1"** must be unique for each item.

So, when you click on a portfolio item image, the ajax will load the content from the html file which is set in "**href**" attribute (in the example above it is "**portfolio-1.html**" file).

The ajax will load a whole file, but we are displaying only content which is placed inside this div:

```
<div class="portfolio-item-wrapper center-relative">
...
</div>
```

All content which is placed inside this div will be loaded and displayed on the front/home page (index.html) using Ajax.

- Image slider

On portfolio-1.html you will find code like this (image slider code):

```
<div class="image-slider-wrapper relative">

<div class="owl-carousel owl-theme image-slider slider" data-speed="2000" data-auto="false" data-hover="true">

    <div class="owl-item">

        

    </div>

    <div class="owl-item">

        

    </div>

    <div class="owl-item">

        

    </div>

</div>

</div>
```


For image sliders you can set slider speed, to set auto start and to control slider pause on mouse over. You need to change this:

data-speed="2000" data-auto="false" data-hover="true"

- Item links to Pop-Up image (Lightbox)

Here is an example for portfolio item which links to pop-up image:

```
<div id="p-item-3" class="grid-item element-item p-one-half image">
    <a class="item-link" href="images/blog_img_03.jpg" data-rel="prettyPhoto[portfolio]">
        
        <div class="portfolio-text-holder">
            <div class="portfolio-text-wrapper">
                <p class="portfolio-text">Sneakers</p>
                <p class="portfolio-cat">Image</p>
            </div>
        </div>
    </a>
</div>
```

As you see the link ("**href**" value) links to the image which will be opened in the pop-up (Lightbox). In our demo that is image **blog_img_03.jpg**

- Item links to Pop-Up video (Lightbox)

Here is an example for portfolio item which links to pop-up video:

```
<div class="grid-item element-item p-one-half video">

    <a class="item-link" href="https://vimeo.com/199192931" data-rel="prettyPhoto[portfolio]">

        

        <div class="portfolio-text-holder">

            <div class="portfolio-text-wrapper">

                <p class="portfolio-text">Design</p>

                <p class="portfolio-cat">Video</p>

            </div>

        </div>

    </a>

</div>
```

As you see the link (“**href**” value) links to the Vimeo video which will be opened in the pop-up (Lightbox). Also, you can use YouTube video in format **https://youtube.com/watch?v=XXXXXXXXXX**

CSS Structure

In the css folder are placed all css files except the main **style.css** file which is in the root folder.

Basically, you don't need to change any of the css files which are placed in the css folder - the only css file which you may need to change (if you need it at all) is **style.css** file.

In **style.css** file you will find code for each section and also for other parts like image slider, contact forms, etc... and each of this part is to start with code (comment code) like this:

```
/* =====  
4. Services Section CSS  
===== */
```

so you will easily find css code for each part.

JS Structure

In the js folder are placed all js files.

There are all 3rd party js files (files for sliders, isotope, jquery, etc...) except the **main.js** - this is our custom file.

In this file you will find the code for setting up the image slider, menu, sending mails, etc....

If you need to change some part here, in JS code, you will need to have some JS skills to do that. If you have trouble, please use our support forum to contact us and we will try to help you.

Note:

We are giving support only for the features which are included with the template, not for additional ones.

Source & Credits

Special thanks to creators and contributors of these awesome libraries, we couldn't have done this without them.

Images used in theme

<https://unsplash.com>

Video used in template

<https://vimeo.com/199192931>

jQuery Isotope Plugin

<http://isotope.metafizzy.co>

jQuery

<http://jquery.com>

Google Web Fonts

<http://www.google.com/webfonts>

Smart Menus

<http://www.smartmenus.org/>

Owl Carousel 2

<https://owlcarousel2.github.io/OwlCarousel2/>

AOS

<https://github.com/michalsnik/aos>

Swiper

<https://swiperjs.com>

prettyPhoto

<http://www.no-margin-for-errors.com/>

We are sorry if we forgot to mention someone.

Questions?

We tried to help you out with this documentation. So please read carefully. If we fail here, you can always go to our [support forum](#) and post a new discussion in the appropriate category. Our developers will try to solve your issue.

Thanks for understanding.



www.cocobasic.com

<https://cocobasic.ticksy.com/>

cocobasicthemes@gmail.com