

# TRUESTORY

Responsive HTML5 Template

## User Guide

---

# Table of Contents

<b>Word from the Author</b>	4
<b>Package Contents</b>	5
HTML Files	5
PHP Files	6
Styles Folder	6
Scripts Folder	6
Plugins Folder	6
Images Folder	7
Assets Folder	7
<b>Template Structure</b>	8
<b>Page</b>	9
Page Structure	9
Page Layouts	13
Page Templates	14
Home Page Templates	14
Portfolio Templates	16
<b>Page Elements</b>	19
Dividers	19
Icons	20
Buttons	22
Image Overlays	24
Image Slider	26
ColorBox	28
Flickr Feed	29
Twitter Feed	30
Contact Form	31

---

---

<b>The Grid</b>	33
Basic Usage	33
Nesting Columns	34
Offset Columns	35
Centered Columns	36
<b>Navigation Menu</b>	37
Adding a menu item	37
Adding a submenu	38
<b>Credits</b>	39
<b>Support</b>	39

## *Word from the Author*

Thank you for purchasing this file. If you have any questions that are beyond the scope of this documentation, please feel free to email me via my user page contact form.

# Package Contents

## HTML Files:

This template includes 34 HTML files:

- **Homepage :**
    - index.html
    - home-accordion.html
    - home-slider.html
    - home-static-image.html
    - home-showreel.html
  - **Features :**
    - columns.html
    - elements.html
    - **Page Layouts:**
      - page-layout-right.html
      - page-layout-center.html
      - page-layout-left.html
      - page-full-width.html
  - **Blog :**
    - blog.html
    - blog-full-width.html
    - blog-single.html
  - **Portfolio :**
    - portfolio-single.html
    - **Classic Portfolio:**
      - portfolio-classic-1.html
      - portfolio-classic-2.html
      - portfolio-classic-3.html
      - portfolio-classic-4.html
    - **Classic Portfolio with Filters:**
      - portfolio-classic-1-w-filter.html
      - portfolio-classic-2-w-filter.html
      - portfolio-classic-3-w-filter.html
      - portfolio-classic-4-w-filter.html
-

- **Gallery:**

- portfolio-masonry.html
- portfolio-metro.html
- portfolio-grid.html

- **Pages :**

- template-clients.html
- template-contact.html
- template-team.html
- template-services.html
- template-error-404.html

### ***PHP Files:***

#### **contact.php**

This file is used and required by the contact form widget and the contact page template, you might need to edit this file and set the e-mail address to which messages should be sent.

*\* You can edit PHP files using and text editors.*

#### **email address validator.php**

This file is required by 'contact.php', you don't need to edit this file.

#### **ajax portfolio entries.php**

This PHP file simulates a database, that is, it has a number of entries defined in an array. In a real world application, this PHP file should connect to a database and retrieve the entries dynamically.

### ***Styles Folder***

This folder contains all style sheets used by the template. It also contains a CSS reset file.

### ***Scripts Folder***

This folder contains all JavaScript files used by the templates, including jQuery.

### ***Plugins Folder***

This folder contains other folders, each folder represents a plugin that is used by the template and each folder contains all the CSS, JavaScript and image files required by that plugin.

---

### ***Images Folder***

This folder contains all the image files and sprites used by the template.

### ***Assets Folder***

This folder contains all images, audio and video files that are used as assets.

## Template Structure

The template consists mainly of the following:

1. Page
2. Dock
  - a. Header
    - i. Logo
    - ii. Navigation menu
  - b. Footer
    - i. Copyright text
    - ii. Social links
3. Page background
4. Preloader

All template sections are contained in a main wrapper which is a div element with an ID **'main-wrap'**.



# Page

## - Page Structure

The page consists mainly of 3 parts:

- 1 - Page head *(Optional)*
- 2 - Page content *(Mandatory, unless the page layout is 'fullscreen')*
  - a – Content *(Mandatory if #page-content is present)*
  - b – Sidebar *(Optional)*
- 3 - Page foot *(Optional)*

```
<div id="page" class="sidebar-right">
  <header id="page-head">
    <!-- PAGE HEAD CONTENT GOES HERE -->
  </header>

  <div id="page-content">
    <div id="content">
      <!-- PAGE CONTENT GOES HERE -->
    </div>

    <aside id="sidebar">
      <!-- SIDEBAR CONTENT GOES HERE -->
    </div>
  </div>

  <footer id="page-foot">
    <!-- PAGE FOOT CONTENT GOES HERE -->
  </footer>
</div>
```

A mandatory part must be included in the HTML code for the page to display correctly.

The following is the code required to display a page without a sidebar, head or foot. This is the most basic form of a page:

```
<div id="page">
  <div id="page-content">
    <div id="content">
      <!-- PAGE CONTENT GOES HERE -->
    </div><!-- #content -->
  </div><!-- #page-content -->
</div><!-- #page -->
```

### 1. *Page head*

---

The page head will usually contain the page title and optional aside content.

```
<header id="page-head">
  <h1>Page title <small>// Page subtitle</small></h1>

  <aside>
    <!-- OPTIONAL ASIDE CONTENT GOES HERE -->
  </aside>
</header>
```

## 2. Page content

---

### Sidebar layouts:

To specify the location of the sidebar, add one of the following classes to the main **#page** HTML element:

#### 1 - [sidebar-right](#)

*Places the sidebar on the right side of the page  
(page width=1020px , content width=960px)*

#### 2 - [sidebar-left](#)

*Places the sidebar on the left side of the page  
(page width=1020px , content width=960px)*

#### 3 - [sidebar-none](#)

*Hides the sidebar and reduces total width of the page to the content width  
(page width=750px , content width=690px)*

After specifying the sidebar-\* class to the #page HTML element, you need to actually add the sidebar HTML code to the page as follows:

```
<div id="page" class="sidebar-right">
  <div id="page-content">
    <div id="content">
      <!-- PAGE CONTENT GOES HERE -->
    </div><!-- #content -->

    <aside id="sidebar">
      <!-- SIDEBAR CONTENT GOES HERE -->
    </div><!-- #sidebar -->
  </div><!-- #page-content -->
</div><!-- #page -->
```

Please note that if you don't specify a sidebar-\* class, the sidebar will be automatically hidden even if the sidebar HTML code is on the page.

### 3. Page foot

---

The page foot will usually contain a pagination.

```
<footer id="page-footer">  
    <!-- PAGINATION CODE -->  
</footer>
```

## - Page Layouts

To specify the page layout, add one of the following classes to the main #page HTML element:

### 1 - layout-right

*Places the page on the right side of the window*

### 2 - layout-left

*Places the page on the left side of the window*

### 3 - layout-center

*Places the page in the middle of the window*

### 4 - fullwidth (default)

*Allows the content to take full width of the window.*

### 5 – fullscreen

*Removes page padding and margin and allows the content to take full width of the window.*

```
<div id="page" class="layout-right">
  <div id="page-content">
    <div id="content">
      <!-- PAGE CONTENT GOES HERE -->
    </div><!-- #content -->
  </div><!-- #page-content -->
</div><!-- #page -->
```

Please note that if you don't specify a layout-\* class, the page layout will be automatically set to 'fullwidth'.

You can add a class 'no-scroll' in addition to 'fullscreen' class in order to disable vertical scrollbars.

This class is mainly used in the home page layouts.

---

## - Page Templates

### 1. Home page templates

---

All home page templates use the **'fullscreen'** page layout and all of them (except static image) use the **'no-scroll'** class.

You can add captions to fullscreen images by wrapping paragraphs, buttons and h1 through h6 elements in a div element with a class **'fullscreen-caption'**.

Children of the **'fullscreen-caption'** element will be automatically animated in the 'Accordion' and 'Slider' templates.

If, for any reason, you need to animate descendant children elements, add a class **'slide'** to that element.

```
<div class="fullscreen-caption">
  <h1>We are TrueStory</h1>
  <h4>
    <span class="slide">We make beautiful </span>
    <span class="slide">applications for the web</span>
  </h4>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus ac turpis eu
    dolor lacinia faucibus.
  </p>
  <button class="color">
    <i class="small briefcase icon l"></i>
    <span class="inline-block">View our work</span>
  </button>
</div>
```

The accordion template will distribute the images uniformly across the width of the window.

This will cause cropping of the images, to change the part of the image that is displayed, add a data attribute to the image indicating the desired position relative to the total width of the image.

In the following example, the position of the image is moved by **44%** of the total image width (**44%=0.44**).

If the image width is **800px**, and we need to move it by **352px** to the left, then

**data-pos=352/800=0.44**

```
<ul id="home-accordion">
  <li>
    <a href="#">
      
    </a>
    <div class="fullscreen-caption">
      <h1>We are TrueStory</h1>
      <h4>
        <span class="slide">We make beautiful </span>
        .
        .
        .
      </h4>
    </div>
  </li>
</ul>
```

## 2. Portfolio templates

---

### Setting up portfolio filters

This template uses two types of filters for the portfolio, search and tags.

```
<div class="portfolio-filters">
  <nav class="filterable-nav" data-id="portfolio">
    <span>Show </span>
    <ul>
      <li>All</li>
    </ul>
  </nav>

  <div class="filterable-search">
    <input type="text" class="has-icon" data-id="portfolio" />
    <i class="search icon"></i>
  </div>
</div>

<ul class="portfolio filterable entries classic-1" data-id="portfolio">
  <li>
    .
    .
    .
```

To attach a filter (search or tags) to a portfolio display, add a data attribute to the portfolio display as shown in the code above and add the same data attribute to the filter.

The search filter doesn't need any special configuration, just add the data attribute to the input element and it will work right out of the box.



The tags filter however might need a little bit more setting up.

If you use the standard entry format, which displays the tags:

```
<span class="entry-tags">
  <span>Digital Art</span>
  <span class="separator">|</span>
  <span>Illustration</span>
</span>
```

The tags will be automatically extracted from each entry and you will not need to do anything to get the filter working.

If you are not using the standard format, then you need to add the tags manually to each entry:

```
<ul class="portfolio filterable entries classic-4" data-id="portfolio">
  <li data-tags="Art Direction,Illustration">
    <article class="entry">
      <div class="entry-featured">
        .
        .
        .
      </div>
    </article>
  </li>
</ul>
```

**Note:**

Character case must be taken into consideration, 'Art Direction' is not the same as 'art direction'.

## Gallery layouts

To setup each one of the gallery layouts (metro, grid and masonry), edit [scripts/main.js](#) using any text editor and set the configuration options at the head of the file.

```
config: {  
  showSubMenuOn: "click" ,  
  /* click | hover */  
  
  portfolio_initialCount: 20 ,  
  /* initial number of entries that should be requested by the ajax call */  
  
  portfolio_count: 10 ,  
  /* number of entries that should be requested by the ajax call on page scroll */  
  
  portfolio_masonry_unitWidth: 320 ,  
  
  portfolio_grid_unitWidth: 320 ,  
  
  portfolio_metro_unitWidth: 320 ,  
  
  portfolio_metro_dynamic: false ,  
  /* set unit size based on the size of the image */  
  
  portfolio_metro_pattern: false  
  /* pattern to follow when sizing metro layout units */  
},
```

The gallery layouts use a PHP file ([ajax\\_portfolio\\_entries.php](#)) to retrieve portfolio entries.

This PHP file simulates a database, that is, it has a number of entries defined in an array. In a real world application, this PHP file should connect to a database and retrieve the entries dynamically.

---

## - Page Elements

### 1. Dividers

---

To create a divider line, create an empty div element with class 'divider'.

```
<div class='divider'></div>
```

This will create a blank divider, in order to create a solid line, add another class 'line'.

```
<div class='divider line'></div>
```

To add a colored bar, add another class 'bar'

```
<div class='divider line bar'></div>
```

## 2. Icons

To create an icon, create an empty span, div or i element with class 'icon'.

```
<i class='icon'></i>
```

Add one of the following names to specify the icon:

delete	left-3	calendar	bookmark
right-2	left	refresh	heart
left-2	right-3	trash	lab
return-right	right	check-2	rocket
return-left	up-3	check	highspeed
return-bottom	up	mic	normalspeed
return-top	down-2	zoomout	lowspeed
link	down	zoomin	attach
pinterest	phone	right-4	fire
anchor	tablet	left-4	puzzle
brush	lifebuoy	comments	piechart
tie	location	comment	lock
ice-cream	star	quotes-right	cancel-2
truck	drop	eye	cancel
cord	cloud	star-2	search
tumbler	leaf	quotes-left	minus-2
dribbble	leaf-2	behance	minus
linkedin	document	wordpress	plus-2
umbrella	pencil	git-2	plus
list	headphones	git	tag
list-2	pictures	blogger	person
grid	picture	googleplus	mail
grid-2	pause	deviantart	android
clock	play	vimeo	phone-2
moon	stop	facebook	down-4
cart	previous	twitter	up-4
basket	next	amazon	screen
warning	loop	skype	bell
question	feather	windows	videocam
info	volume	apple	minus-3
bolt	camera	at	plus-3
feed	home	text	page-top
share	flickr	sun	show-dock
briefcase	youtube-2	cup	hide-dock
flag	youtube	wrench	fullscreen
1. fullscreen-exit			

```
<i class='star icon'></i>
```

To specify the size of the icon, add one of the following classes:

1. `tiny`
2. `small`
3. `large`
4. `huge`

```
<i class='huge star icon'></i>
```

### 3. Buttons

---

Buttons are available in 3 types and 4 sizes:

**Types:**

1. `normal` (default)
2. `secondary`
3. `color`

**Sizes:**

5. `small`
6. `normal` (default)
7. `large`
8. `huge`

```
<button class='huge secondary'>Label</button>
```

You can also create an anchor and give it the same styles of a button by assigning a 'button' class to its element.

```
<a href='#' class='huge secondary button'>Label</a>
```

```
<input class='huge secondary button' value='Label' />
```

To create an iconed button, add an icon to the button content and specify the location of the icon by adding a class that indicates the position as follows:

**r=right**

**l=left**

**t=top**

**b=bottom**

The following example, shows a button with twitter icon placed on the right

```
<button class="large">
  <span class="inline-block">Iconed</span>
  <i class="r twitter icon"></i>
</button>
```

The following example, shows a button with facebook icon placed on the left

```
<button class="large">
  <i class="l facebook icon"></i>
  <span class="inline-block">Iconed</span>
</button>
```

#### 4. Image overlays

---

To create an image overlay, start by wrapping the image tag in a div with class 'overlay'.

```
<div class="overlay">
    
</div>
```

Nothing should change as of yet, you need to add a span tag with class '**color-overlay**' as a child of the '**.overlay**' div:

```
<div class="overlay">
    <span class="color-overlay"></span>
    
</div>
```

This will create a white overlay on the image (rollover the image to see it).

You can change the color overlay to dark by adding a class '**dark**' to the color-overlay element.

```
<div class="overlay">
    <span class="dark color-overlay"></span>
    
</div>
```



To add content overlay, add another span with class 'content-overlay':

```
<div class="overlay">
  <span class="dark color-overlay"></span>
  <span class="content-overlay">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </span>
  
</div>
```

To use an icon as content overlay, add a class 'icon' to the content-overlay element and the icon type.

```
<div class="overlay">
  <span class="dark color-overlay"></span>
  <span class="content-overlay play icon"></span>
  
</div>
```

The overlay icon size will be fixed, so there is no need to specify the icon size.

## 5. Image Slider

---

To create an image slider, copy and paste this code:

```
<div class="flexslider">
  <ul class="slides">
    <li></li>
    <li></li>
    <li></li>
  </ul>
</div>
```

The slider accepts a set of options that can be set as data attributes:

1. **controls**

*true or false , indicates whether to show the control navigation or not*

2. **nav**

*true or false , indicates whether to show the direction navigation or not*

3. **slideshow**

*true or false , indicates whether the autoplay mode is enabled or not*

4. **animation**

*"slide" or "fade" , specifies the animation type*

5. **pauseonhover**

*true or false*

6. **autohide**

*true or false , indicates whether the interface should be hidden on mouse out or not*

7. **duration**

*number, default value is 5000 milliseconds*

8. **speed**

*number, default value is 600 milliseconds*

```
<div class="flexslider" data-autohide="false" data-controls="false">
  <ul class="slides">
    <li></li>
    <li></li>
    <li></li>
  </ul>
</div>
```

You can add a text caption to any slide by appending a paragraph element to the list item with class **'flex-caption'**

```
<div class="flexslider" data-autohide="false" data-controls="false">
  <ul class="slides">
    <li>
      
      <p class="flex-caption">This is a text caption</p>
    </li>
    <li></li>
    <li></li>
  </ul>
</div>
```

## 6. ColorBox

-----

To open a link in a colorbox window, add a 'colorbox' class to the link or the element.

For example, to open an image in colorbox using a normal link:

```
<a href="path_to_image" title="Launch Colorbox" class="colorbox">
    Launch Colorbox
</a>
```

Here's another example, to open an image in colorbox using a button:

```
<button data-href="path_to_image" class="colorbox">
    Launch Colorbox using a button
</button>
```

For more information on Colorbox, please visit this link:

<http://jacklmoore.com/colorbox/>

## 7. Flickr Feed

---

Create a div with class '**flickr-feed**' and add an attribute to this div as follows:

```
<div class = "flickr-feed" data-user = " Flickr_username "></div>
```

To specify the HTML structure for each image, add another div as a child of that div as follows:

```
<div class="flickr-feed" data-user=" Flickr_username " data-count="6">
  <div class="image">
    <a href="{url}" rel="nofollow" target="_blank">
      
    </a>
  </div>
</div>
```

## 8. Twitter Feed

---

Create a div with class 'twitter-feed' and add an attribute to this div as follows:

```
<div class = "twitter-feed" data-user = "Twitter_username"></div>
```

To specify the HTML structure for each tweet, add another div as a child of that div as follows:

```
<div class = "twitter-feed" data-user = "Twitter_username">
  <div class = "tweet">
    <div class = "text">{text}</div>
    <div class = "time">
      <a href = "{tweet_url}" rel = "nofollow">{time}</a>
    </div>
  </div>
</div>
```

## 9. Contact Form

---

This is the HTML structure for the contact form:

```
<form class="contact-form" action="contact.php" method="post">
  <input type="hidden" name="to" value="email@domain.com">
  <input type="text" data-prompt="Name" name="name">
  <input type="text" data-prompt="E-mail" name="email">
  <textarea data-prompt="Message" name="message"></textarea>

  <p class="info"><span class="color">*</span> All fields are required</p>

  <button name="submit" type="submit">
    <i class="small mail icon l"></i>
    <span class="inline-block">Send Message</span>
  </button>

  <div class="form-success">
    <h5 class="color">Thank you</h5>
    <p>
      We received your message and will be in touch with you as soon as possible.
    </p>
  </div>

  <div class="form-failure">
    <div class="red micro box">
      <h6 class="red">Error</h6>
      Your message couldn't be sent, please try again.
    </div>
  </div>
</form>
```

The form submits data to 'contact.php' via an ajax call.

To set the default email address to which messages should be sent, edit '**contact.php**' using any text editor and modify this variable:

```
$to = isset($_POST['to']) ? $_POST['to'] : 'your@email.com';
```

Or you can add a hidden input named '**to**' as shown in the code above.



## The Grid

### Basic Usage

---

This template uses a 6 columns responsive grid.

Basically, you need to create a **.row** element and then place a number of columns that add up to a whole one.

#### For example:

$$(1/4) + (1/4) + (1/2) = 1$$

```
<div class="row">
  <div class="col-1-4">Content</div>
  <div class="col-1-4">Content</div>
  <div class="col-1-2">Content</div>
</div>
```

## Nesting Columns

---

To nest your content, add a new `.row` and set of `.col-*` children within an existing `.col-*` element.

Nested rows should include a set of columns that also add up to a whole 1.

Nested columns are also responsive and there is no limit on how deep can nesting go.

Note that the width of each column will be set based on the width of its parent `.row` and NOT the total page width. This opens a door for creating more complicated layouts and allows creating more than 6 columns per row.

**For example:**

```
<div class="row">
  <div class="col-1-2">
    <div class="row">
      <div class="col-1-2">Content</div>
      <div class="col-1-2">Content</div>
    </div>
  </div>

  <div class="col-1-2">
    <div class="row">
      <div class="col-1-3">Content</div>
      <div class="col-1-3">Content</div>
      <div class="col-1-3">Content</div>
    </div>
  </div>
</div>
```

## Offset Columns

---

Sometimes, for any reason, you might want to move a column to the right or to the left. Offsets allow you to create additional space between columns in a row.

You can offset a column simply by adding a class of **push-{columns span}** (move to the right) or **pull-{columns span}** (move to the left).

**For example:**

```
<div class="row">
  <div class="col-1-4 push-3-4">Pushed 1/4</div>
  <div class="col-3-4 pull-1-4">Pulled 3/4</div>
</div>
```

## Centered Columns

---

You can center a column just by adding a class centered to the column block.

Centered columns are placed in the middle of the row.

This does not center their content, but centers the grid element itself.

This is a convenient way to make sure a block is centered, even if you change the number of columns it contains.

### Note:

There cannot be any other column blocks in the row for this to work.

```
<div class="row">
  <div class="col-3-4 centered">
    <div class="row">
      <div class="col-1-2"><span>Nested 1/2 Column</span></div>
      <div class="col-1-2"><span>Nested 1/2 Column</span></div>
    </div>
  </div>
</div>
```

## Navigation Menu

### *Adding a menu item*

---

Open any HTML page using a text editor and scroll down to `<nav id = "main-nav ">` and add a new list item to the un-ordered list.

```
<nav id = "main-nav ">
  <ul>
    <li><a href = "index.html">Home</a></li>
    <li><a href = "new-page.html">New Page</a></li>
  </ul>
</nav>
```

You must add any changes you make to each one of the HTML files.

### *Adding a submenu*

---

Open any HTML page using a text editor and scroll down to **<nav id = "main-nav ">** and add an un-ordered list to a certain item as follows:

```
<nav id = "main-nav ">
  <ul>
    <li><a href = "index.html">Home</a></li>

    <li>
      <a href = "new-page.html">New Page</a>
      <ul>
        <li><a href = "page-1.html">Sub Item 1</a></li>
        <li><a href = "page-2.html">Sub Item 2</a></li>
      </ul>
    </li>
  </ul>
</nav>
```

## Credits

- jQuery by [John Resig](#)
- jQuery ColorBox by [Jack Moore](#)
- jQuery Easing plugin by [George McGinley Smith](#)
- Flex Slider plugin by [WooThemes](#)
- MediaElement plugin by [John Dyer](#)
- [Modernizr](#)
- All web fonts used in this theme are hosted by [Google Web Fonts](#)

## Support

Once again, thank you so much for purchasing this file. As I said at the beginning, I'd be glad to help you if you have any questions relating to this item. No guarantees, but I'll do my best to assist. If you have a more general question relating to the files on ThemeForest, you might consider visiting the forums and asking your question in the "Item Discussion" section.