



# THE PIE FRAMEWORK

PieCSS 1.0+ Beta

Documentation by Eugy Enoch



## INTRO

PieCSS(Pie-CSS) or pie.css is a modern and SEO friendly framework and part of the Pie Framework which also includes The PieJs or pie.js technology. It is intended to be among the most used HTML, CSS and JS frameworks for developing websites and web applications that are responsive and suitable for any and every occasions.

PieCSS is also intended to add speed and make learning and using the framework and technology easy for all.

**Above all, PieCSS is the first SEO, SMO and Digital Marketing friendly CSS framework.**

## AVAILABILITY AND COST

Availability of the PieCSS framework is intended for free and at no costs at all. The PieCSS framework can be downloaded from the internet at authorized points and used freely for building websites and web applications.

Begin here => <https://eugyenoch.github.io/pie/>

## LICENSING

The PieCSS framework is licensed under the MIT license.

Read more about the MIT license here:

<https://github.com/eugyenoch/pie/blob/master/LICENSE>



## THE DEVELOPER

The initial developer for pie is an experienced, talented and exceptional leader committed to maintaining cutting edge technical and relationship skills and up-to-date industry knowledge; with outstanding relationship building, training and presentation skills and excellent at juggling multiple tasks and working under pressure. Broad industry experience includes Education, ICT, Books Publishing/Promotion, Media, Sales, Administration, Charity and Religion.

David Ugochukwu Enoch(Eugy Enoch) currently works at the APTECH Education Center as a Faculty and welcomes collaborators and issues rising out of the usage of the PieCSS software. Reach him on Github + Twitter + LinkedIn + Facebook + Telegram + Instagram = @eugyenoch

## COLLABORATORS AND SPONSORS

Sponsors are welcome to contribute via any number of ways and Collaborators are welcome to contribute via Github to PieCSS in any ways possible including but not limited to new issues, commits and other contributions. Collaborators and sponsors are treated as co-owners in the project.

## ISSUE REPORTS

Reporting issues for now can be done through the Pie Github platform and on all Ojaay platforms.

## THE CSS INITIALIZATION

This section is added, partly, to reset a few major HTML elements to a consistent baseline. The browser initialization rules handles the rest.

## NORMALISE.CSS AS USED IN THE PIECSS FRAMEWORK

The essence of the normalize.css extract in this project is to “normalize” browser and cross-browser behavior when web pages are rendered across the different browsers. The normalize project has been featured in so many tools, frameworks and projects up till date.

*Normalize.css is by Nicolas Gallagher and Jonathan Neal and released under the MIT license*

## WHY USE PIECSS

Advantages of PieCSS:

- ☐ PieCSS is built to be and is {absolutely} SEO, SMO and digital marketing friendly
- ☐ Easy to use by anybody with even the faintest knowledge of HTML and CSS
- ☐ PieCSS is compatible with all modern browsers be it Firefox, Edge, Chrome, Opera, UCBrowser, Safari or the myriads of desktop and mobile browsers that implement modern standards and API
- ☐ The pie frameworks, generally, allow users build prototypes of a solution quickly.



## PREPROCESSOR

A preprocessor is a computer program that processes input data and produces an output that is used as input to another computer program. In CSS, a preprocessor lets you generate CSS from the preprocessor's own unique code.

The preprocessor implemented for the pie.css framework asides “#CSS” is the “Sassy CSS => SCSS” because it allows for easy implementation of CSS3 syntax. Basically, every valid SCSS stylesheet is valid in CSS3 and Sass as well and which makes Scss less cumbersome and easier to understand .

## IMPLEMENTATION OF SCSS

For this version of pie.css, the Scss preprocessor have been implemented in the *chunks*(individual stylesheets for the different presentation systems) and not in the *combine*(combine and combine-minified) pie.css file.

*\*\*To use Scss, Sass will have to be downloaded and installed on local machine.*

## PREPROCESSOR INSTALLATION

### General

Sass has to be downloaded (from Github or elsewhere) and installed first on the command prompt to be able to run the .sass executable and to compile .sass and .scss files to .css files. After installing, run this code to check installation and compatibility => `sass --version` or `sass -v`

### On Node.js

Sass can be installed on Node.js using npm by running this code => `npm install -g sass`

### On Windows

Sass can be installed on Windows by downloading the Chocolatey package manager for windows and running this code to install => `choco install sass`

### On MacOS

Sass can be installed on Windows by downloading the Homebrew package manager for MacOS and running this code to install => `brew install sass/sass/sass`

*\*\*The popular Sass on Ruby has reached end-of-life in 2019 and Dart Sass is recommended instead  
- Install Dart Sass on Windows and MacOS using any of the procedures already outlined above*

## THE PEOPLE BEHIND PIECSS

The team at the Pie Framework

Github => <https://eugyenoch.github.io/pie/>

Facebook => <https://www.facebook.com/ojaay>

Twitter => <https://twitter.com/ojaay>

Linkedin => <https://www.linkedin.com/company/ojaay>

Founder

Github: <https://github.com/eugyenoch>

Twitter: <https://twitter.com/eugyenoch>

Facebook: <https://www.facebook.com/eugyenoch>

Linkedin: <https://www.linkedin.com/in/eugyenoch>

**An aside =>** Are you a CSS or other website styling guru or enthusiast? We want you to join us, to contribute to the continued development of the PieCSS framework. We appreciate your coming a lot and look forward to working with you.





# STRUCTURE

Folder structure and other preambles

- Contains part files of the css declarations and may require linking to more files in web document

- Part files for implementation of the sass/scss protocols

File

SASS

Raw &  
Minified  
PieCSS

Include

- pie.css
- pie.min.css
- documentation

- Contains declared fonts and their documentation

The PieCSS Folder Structure for v1.0+ Beta Release

## CONTENT OF THE PIECSS FRAMEWORK

PieCSS adopts a simple folder structure and adopts an easy to understand file naming pattern includes HTML and CSS based design classes for forms, buttons, tables, image galleries, flex boxes, grid boxes, navigation, typography(text and lists), cursor functions, box model functions and entire web page manipulation.

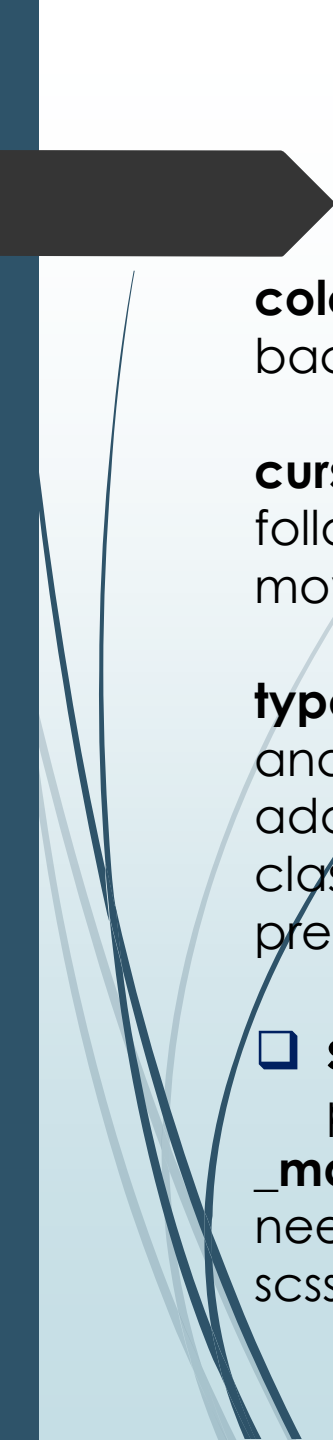
**Note: If *dates.js* was moved to a different folder, it is advised to indicate the correct file path.**

## FOLDER STRUCTURE AND CONTENT

The following folders are available:

- ❑ **Piecsc** => root folder for the piecss framework and also contains the files: *pie.css*; *pie.min.css*; some licensing documents and some folders: *files*; *include*; *scss*.
- ❑ **Files** => contains fragments or the broken down parts of the PieCSS framework such as the: **main.pie.css** – Contains the “center” of pie css and the normalize css extracts.

**arc.pie.css** – Intended for “panel products” and contains classes that can be used for navigation lists(vertical and horizontal), tree view lists(checked and arrow), dropdowns and basic panel products.



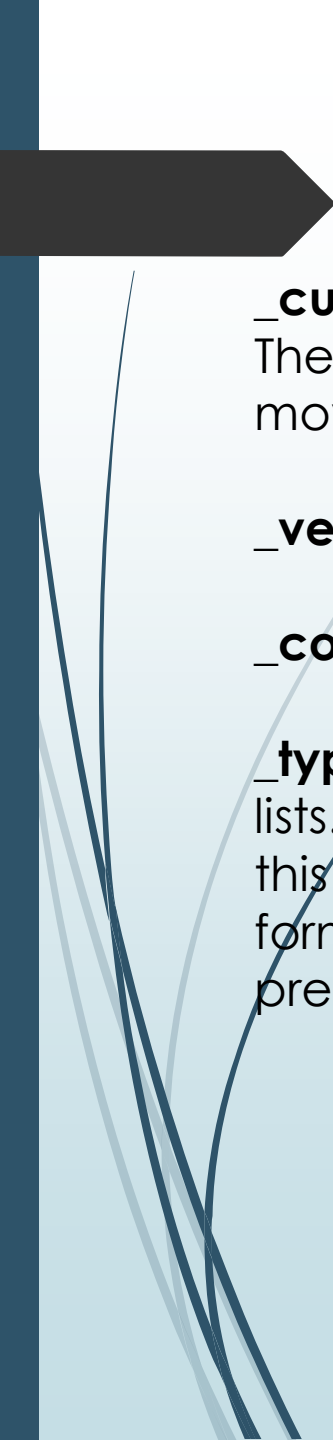
**color.pie.css**- This will handle all premade and custom color schemes for the website including for backgrounds, texts, border, column fills, carets, borders and their hover and focus colors.

**cursor.pie.css**- Dedicated entirely to cursor functions and manipulation using CSS classes. The following are possible: cursor styling, cursor pointer styling, hiding/disable cursor, bi-directional movements, printing and cursor animation.

**typo.pie.css/scss** – This file provides the user with CSS classes for typography which includes texts and lists. The file helps with classes for texts – all kind of resets and formatting with texts and in addition, this file also replaced the *list.pie.css/scss* file (in v1.0+ Beta releases) as it also contains classes that formats lists with CSS declarations that handles list style types, list style positions, premade and custom made theme modes for both ordered and unordered list elements.

❑ **scss**=> This is a separate scss folder included which contains all sass(.scss) compliant files for the pie framework.

**\_main.pie.css** – Contains the “center” of pie css and the normalize css extracts and the only needed file as it contains link to other declarations and variables needed and included in other scss part files.




**\_cursor.pie.scss** - Dedicated entirely to cursor functions and manipulation using CSS classes.

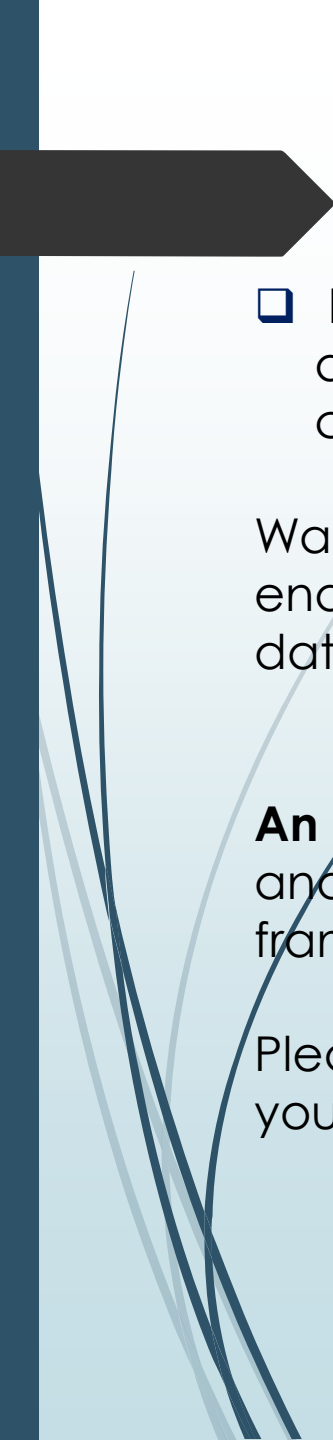
The following are possible: cursor styling, cursor pointer styling, hiding/disable cursor, bi-directional movements, printing and cursor animation.

**\_vendor.pie.scss** – Declares vendor prefixes for other sass compliant declarations.

**\_colorVariable.pie.scss** – Declares vendor prefixes for the `_color.pie.scss` file.

**\_typo.pie.css** – This file provides the user with CSS classes for typography which includes texts and lists. The file helps with classes for texts – all kind of resets and formatting with texts and in addition, this file also replaced the *list.pie.scss* file (in v1.0+ Beta releases) as it also contains classes that formats lists with sass-compliant declarations that handles list style types, list style positions, premade and custom made theme modes for both ordered and unordered list elements.

- 
- ❑ **Fonts** => contains used fonts and documentations on used fonts. Folders and files included:
    - > **abeeze** – contains the “abeezee italic” and “abeezee regular” fonts and the abeeze fonts documentations.
    - > **homoarak** – contains the “homoarak” fonts and the expressway rg font documentations.
    - > **loopy** – contains the “loopy italic” and “loopy regular” fonts and the loopy fonts documentation.
    - > **trashco** – contains the “trashco” fonts and the trashco fonts documentations.

- 
- ❑ **Doc** => Contains documentation for the PieCSS(pie.css) framework. Please note that the documentation, though , a very good guide is not intended to replace learning and certification in the pie framework.

Watch out for official trainings and certification in the Pie frameworks, meanwhile send any enquiries to the following contact information and more as may be revealed officially in future dates by the Pie team. You should also follow these social media handles.

**An aside** => All non-minified files that come with the pie framework are commented enough and are intended to guide you even further in appreciating the work done in and using the Pie framework.

Please always use `<meta name="viewport" content="width=device-width,initial-scale=1.0" />` in your `<head>` section for responsiveness and standard.



# GETTING STARTED

Guide to getting started



## GUIDE TO GETTING STARTED

The PieCSS folders contain the files needed for your website and there are two ways of linking your CSS files from the PieCSS folders - one or the other:

1. Method 1 is by linking one main file in the folder named `pie.css`
2. Method 2 is by linking one main sass file named `_main.pie.scss` and found in the `scss` folder

### METHOD 1 – The CSS method

The method 1 is achieved by opening the `piecss` and linking to the combined file there named “`pie.css`” The appropriate way to link to this file is via the `<link>` tag placed within the `<head>` section of your `<html>` document as shown below:

```
<link rel="stylesheet" href="https://mywebsite.com/pie/piecss/pie.css" type="text/css" />
```

The “`type=text/css`” is, nowadays, optional and hence not required at all as CSS is now the default styling language of the web and for most browsers as well so the above linking can be refined and presented preferably thus:

```
<link rel="stylesheet" href="https://mywebsite.com/pie/piecss/pie.css" />
```

*NOTE: Replace “mywebste” with the actual name of your web document.*

## METHOD 2 – The SASS method

The method 2 is achieved by opening the folder named `scss` and linking to the `_main.pie.css` file only. This part file includes links to the other declarations and variables you will need from the other part files.

The appropriate way to link to these files is via the `<link>` tag placed within the `<head>` and, optionally, `<body>` section of your `<html>` document as shown below:

The sass stylesheet below must be linked first

```
<head> <title>My Website</title>
/*You must link to the _main.pie.css first in the head section*/
<link rel="stylesheet" href="https://mywebsite.com/pie/piecss/scss/_main.pie.css" />
</head>
```

*NOTE: Replace "mywebste" with the actual name of your web document.*

## DIFFERENCE BETWEEN METHOD 1 AND METHOD 2 STYLES AND STYLESHEETS

Method 1 – the combined `pie.css` is most preferred for production sites as all CSS declarations are contained in a combined and minified/non-minified files coupled with our CSS initialization functions and Browser normalization extracts. The CSS codes are commented appropriately for the non-minified `css` to help explain the parts and introductions that are made to the PieCSS framework.

Method 2 – Works very well too in production, this will use the sass protocols to add stylesheets to your web document. The various files are more ideal to learn the PieCSS in an easy to understand broken down manner.

- The `pie.css` file contains all the declarations in the sass stylesheets and part `css` files (`_main.pie.css`, `_arc.pie.css`, `_color.pie.css`, `_cursor.pie.css` and `_typo.pie.css`) files combined and minified.

**The “myPie” word attached to a class or id used in any files, allows you to add custom values to CSS declarations.**



# WORKING WITH LAYOUTS

Understanding the PieCSS layout system

## GUIDE TO LAYOUTS

In PieCSS, your `<body>` element is proscribed a circle containing various sectors for holding elements which include the `pie`, `radius` and `arc` sectors.

### ❑ The “`pie`” sector

The `pie` sector is the major sector, that is, the **outer major sector** that holds the container for holding your rows, elements and contents. The `.pie` related classes are useful in describing, creating and including this major container on your website. The default *display* mode for all `pie` sectors is *block* with a *margin* of `6px` as can be seen in the code below:

```
#pie{overflow:hidden; display:block; margin:6px;}
```

For now, the `.pie` class can be used in place of the `#pie` id to achieve the same results pending future releases of the PieCSS framework where the `#pie` id is proposed for modification or removal

## ❑ The “radius” sector

The radius sector is the **inner major sector**, that is, the major inner row(s) that holds your elements and their contents within the main container, that is the pie. The `.radius` related classes are useful in describing, creating and including the *inner major rows* on your web pages.

The default *display* mode for all radius sectors is *inline-block* with a *padding-top* and *padding-bottom* of *10px* as can be seen in the code below:

```
#radius{position:absolute;overflow:hidden;padding-top:10px;padding-bottom:10px;
display:inline-block;}
```

For now, the `.radius` class can be used in place of the `#radius` id to achieve the same results pending future releases of the PieCSS framework where the `#radius` id is proposed for modification or removal.

The `.radius` class also defines several row width as can be seen in the codes in the figure that follow:

## The “radius” sector – inner major row classes:

Class/Id - only one selector is required	Row/Col width
.radius-row, #pie-360	100% => Full row width => 1 column
.radius-row-half, #pie-180	50% width each => 2 equal columns in a row
.radius-row-third, #pie-120	33.33% width each => 3 equal columns in a row
.radius-row-quarter, #pie-90	25% width each => 4 equal columns in a row
.radius-row-two, #pie-66	66.66%, 33.33% width => 2 unequal columns in a row
.radius-row-three, #pie-75	75% , 25% width => 2 unequal columns in a row

*\*\*All rows have been adjusted for mobile responsiveness*

## Row(radius) manipulations in PieCSS

Class	Meaning
<code>.radius-recover</code>	Work with this class to recover any overflown row(s)
<code>.radius-stretch</code>	Work with this class to stretch any row(radius)
<code>.pie-pov</code>	Position, Overflow and Visibility of a radius(row)
<code>.radius-block .col-2</code> up to <code>.col-5</code>	Creates newspaper style columns; up to 5 columns
<code>.col-gap-xs</code> up to <code>.col-gap-xl</code>	Creates column gaps between columns
<code>.radius-block .col-reset</code>	Resets newspaper style columns



## ❑ The “arc” sector

The radius sector is an **inner minor sector**, that is, a minor inner row(s) that holds your elements and their contents within the main container(the pie) or within a radius. The `.arc` related classes are useful in describing, creating and including *inner minor rows* on your web pages which can be useful for asides, dropdowns, sidebars, or any of their likes.

*{{Your dropdowns can also use the .radius class functionality}}*

Arcs comes in two forms – Fluid and Fixed.

A typical fluid `.arc` declaration comes with this rule:

```
.arc{display:inline-block;width:auto;height:auto;position:absolute;overflow:hidden;}
```

A fixed `.arc` declaration comes with this rule:

```
.arc-fixed{display:block;position:fixed;z-index:1001;}
```

For now, the `#arc` id can be used in place of the `.arc` class to achieve the same results pending future releases of the PieCSS framework where the `#arc` id is proposed for modification or removal.

*\*\*arcs are adjusted for responsiveness*

## ❑ Other Layout Manipulations

Your layout(containers(pies) or rows(radiuses/arcs) or blocks(arcs) – major or minor) can be manipulated in more ways than one, for examples:

Function	Class	Description
Overwrite CSS	.pie-overwrite	Overwrite CSS for a radius/arc/row/container
Hide/Show	.pie-tab .pie-hide .pie-show, .pie-show-block .pie-show-inline-block .pie-show-inline	Different modes of the Hide and show (site/pagewide) for containers and their contents.  <i>Refer to my official tutorial or the CSS files – pie.css or _main.pie.css for a description of the classes.</i>
Adding responsiveness	.pie-responsive, .radius-responsive, .arc-responsive	Add responsiveness to a container and row
Center a page	.site-center	Center an entire site/page
Vertical center	.vc-align	Vertical center alignment
Block Radius	.radius-block	Displays a radius(inner row) in block
Disable right click <i>within</i>	.radius-disable-click	Prevent user right-clicking any element within a radius(i.e row/container)

## ❑ Other Layout Manipulations II

In continuation, your layout(containers(pies) or rows(radiuses/arcs) or blocks(arcs) – major or minor) can be manipulated in more ways than one, for examples:

Function	Class	Description
Disable right click on entire site	*, .pie-disable-click	Prevent user right-clicking within entire site or within a class - will be deprecated in future release
Recover a radius	.radius-recover	Use this to recover a radius (row or section of a row)
Stretch a radius	.radius-stretch	Stretch a radius(row)
Row measurements	. pie-pov	row measurements using Position -> Overflow -> Visibility
Row contents	.radius-content, .row-auto	Formatting the various row contents
Sidebar	.pie-sidebar	Used together with the .move-left or .move-right class, This class is useful for sidebars of not more than 200px in width. <i>Adjusted for responsiveness</i>
Radius widgets	.radius-linewidget	Markout a functionality or element and make it clearly visible amongst other elements. Use together with the .sector-width class to specifically add the width property.

## ❑ The Modal Container

A modal container or modal box pops up or pops over content to display relevant information to a site visitor.

`.radius-modal` and `.radius-modal-content` for the modal box and modal box contents.

A typical modal box class with pie has the following declarations:

```
.radius-modal{z-index:3;display:none;padding-top:90px;position:fixed;left:0;top:0;width:100%;height:100%;overflow:auto;background-color:rgb(0,0,0);background-color:rgba(0,0,0,0.4)}
```

A modal box content class with pie has the following declarations:

```
.radius-modal-content{margin:auto;background-color:#fffafa;position:relative;padding:0;outline:0;width:600px}
```

Use with any color class of your choice and including pie.js will help add the close button for the modal box.

*\*\*The modal box is built to be responsive*

## ❑ Aspect Ratios

Maintain aspect ratios for a radius(container/row)

Use together with ".text-align-aspect" or ".content-align-aspect" classes.

Aspect Ratio	Class
1:1 aspect ratio	.aspect-ratio-11
3:2 aspect ratio	.aspect-ratio-32
4:3 aspect ratio	.aspect-ratio-43
8:5 aspect ratio	.aspect-ratio-85
16:9 aspect ratio	.aspect-ratio-169

*\*\*Aspect ratios work with any content forms – texts, video, audio and more of such kind.*

## ❑ Tables

A table is basically an element that produces container rows and columns when used.

Using `.table-section` class will make table layouts responsive by cutting off overflow along the x-axis only and this is achieved by adding `overflow-x:auto;` to your layout, however, for a PieCSS table, we recommend you use the `.responsive` class to your `<table>` element.

**Working with borders** - By default PieCSS tables have no borders(except at the bottom when you use the `table.bordered` class , however, to add border, use any of the `.border-color-?` color classes to add fancy borders of your choice.

**Working with other table designs** – Your table can be striped, using the `.striped` class and hoverable using the `.hoverable` class to your `<table>` element.

**Aligning elements within and without your table** – Elements inside your table can be centered using the `.centered` class; Left-aligned using the `.left-aligned` class and right aligned using the `.right-aligned` class to your `<table>` element.

Table captions can be placed at the bottom of table using the `.bottom-caption` class and at the top of table using the `.top-caption` class.

## ❑ Flip Boxes

Flip boxes are boxes with content that rotate horizontally or vertically, displaying contents on either side of the box. The rotation can be initiated by a variety of events. In PieCSS we make use of the Hover event only.

You can use `.pie-flip-back` and `.pie-flip-front` classes together with any color class to produce a different background color and text color.

### How to –

- `.pie-flip` class defines you a flip box to use for your content.
- `.pie-flip-front` class styles the immediate visible part of your flipbox(the front).
- `.pie-flip-back` class styles the reverse of the box(the back).
- `.pie-flip-front`, `.pie-flip-back` class positions the front and back sides.
- `.pie-flip-back-horizontal`, `.pie-flip-inner-horizontal` classes flips the box horizontally.
- `.pie-flip-back-vertical`, `.pie-flip-inner-vertical` classes flips the box vertically.

## GRID

The grid system is a layout module that presents a content area of a web page in rows and columns, making it easier to design web pages without the need for positioning and floats.

By default, in this version, the PieCSS grid display adopts a 4x4 layout to display data in a grid.

Use the `.pie-grid-block` to define a grid(block layout) in your webpage => all grid rows and column are equal in this version of PieCSS. Use `.pie-grid-block .grid-content` classes to style contents in a grid.

Grids can be represented in block or inline layouts: to use the inline grid layout, simply replace `.pie-grid-block` class with `.pie-grid-inline` class

*\*\*You should know that grids can also be styled together with any of the PieCSS color and text presentation styles.*





## ❑ Footers

While you can work with any container element available to create your footer, the pie.css framework gives you an opportunity to style your footer in more unique ways than one. For now, all footer classes is intended to work with the `<footer>` HTML element.

### How to –

- `.page-footer` class maps out a footer area for your web page.
- `.page-footer .footer-copyright` class adds styling to a content laden footer.

There is a default color scheme that comes with the but you can try to blend with our several other color classes.



# WORKING WITH CONTENT

Understanding the PieCSS Content and Content Presentation System



## GUIDE TO CONTENT AND CONTENT PRESENTATION

Contents are the pieces on the puzzle box. They are the visible items on your website that users will interact with such as: texts, images, menus, and other related items.

In the PieCSS framework, Content include: typography(texts and lists), Multimedia(images, audio and video), hyperlinks and the Content Presentation System include: the way content is presented/displayed(navigation and slides, for example) and positioned(animated and fixed, for example.)

### How to present content in PieCSS

Contents are presented following appropriate CSS rules. In PieCSS, contents are advisably presented by first creating a major outer container – optional, in which can contain a major inner container or row(radius in PieCSS) or a minor inner container or row(arc in PieCSS). It is within an inner container or rows that your contents should go in, ideally, to be able to achieve the PieCSS styling in perfection.

*\*\*You shouldn't have any issues if you choose to go instead with the normal CSS rules as PieCSS is, at best, crafted to obey such rules as well. However should you encounter any issues in PieCSS, please use any of the contact information in the early pages of this documentation to report.*



# CONTENT PRESENTATION



## NAVIGATION

Navigation list major links that are intended to guide a user through his journey on a web page. Navigation is an integral part of any web development and SEO efforts for any business or website. The PieCSS framework deals with navigation and we will try to explain as much as we can.

**Dropdowns and hover** – Use the `.arc-dropdown-clicks`, `.arc-dropdown-hover` classes together in on naming instance to define a navigation. Also include the `.arc-dropdown-content` while naming the contents of your menu.

Optionally, there is a `.arc-button` class if you want your navigation via a button element.

**Vertical navigation** – Use the `.arc-navv` class to define a vertical navigation. In addition, use the `.arc-navv` `.arc-navv-item` classes to style navigation items.

If you prefer button for your navigation, then use the `.arc-navv` `.arc-navv-button` classes in one class naming instance.

**Vertical navigation using list** – To use the ordered list and unordered list for vertical navigation and style it effectively, add the `.arc-navv` class to your `<ol>` and `<ul>` element. Thereafter, for each list item, use the `.arc-navv-list` class.



**Horizontal Navigation** – Use the `.arc-navh` class to define a vertical navigation. In addition, use the `.arc-navh` `.arc-navh-item` classes to style navigation items.

If you prefer button for your navigation, then use the `.arc-navh` `.arc-navh-button` classes in one class naming instance.

**Horizontal navigation using list** – To use the ordered list and unordered list for vertical navigation and style it effectively, add the `.arc-navh` class to your `<ol>` and `<ul>` element. Thereafter, for each list item, use the `.arc-navh-list` class to your `<li>` element.


**Navigation tree view for both checked and arrow modes** – To achieve treeview style navigation using either the arrow or checkbox as “click to dropdown” pointers, use the following classes.

`.tree-view` to style your `<ol>` and `<ul>` element. `.tree-view-arrow` to style arrow property.

Include `.tree-view-nested` and `.tree-view-active` classes to toggle list view.

Replace the `.tree-view-arrow` class with `.tree-view-check` and `.tree-view-checkbox` classes to use the checkbox mode for your navigation (using list view).

*\*\* The tree and checkbox navigation list view modes work with PieJS to achieve perfection so use together with pieJS. You may refer to pieJS documentation on usage.*



**Logo in navigation** – Use the `.nav-logo` to position your logo with `position: absolute;` or in the alternative, use the `.nav-logo-center` and `.nav-logo-right` classes to position your logo center and right respectively.

Logo in navigation is built to be responsive and automatically adjusts for screens not wider than 900px.

**Fixed navigation** – `.navbar-fixed` property to fix your navigation bar only or the `.nav-fixed` property to Achieve this same with by appending the following `{position: relative; height: 58px; z-index: 998;}` The fixed navigation has been adjusted for responsiveness on screens with minimum width of 600px to achieve a height of 62px.

*\*\*You shouldn't have any issues if you choose to go instead with the normal CSS rules as PieCSS is, at best, crafted to obey such rules as well. However should you encounter any issues in PieCSS, please use any of the contact information in the early pages of this documentation to report.*

*\*\*Use navigation classes together with any of our color classes or text classes to add color schemes and text styling.*

*\*\*Icons are supported in any navigation pattern you choose.*




## THE PIEBOX ANIMATION MODEL

The piebox is simply a way of sliding images with icon and text contents. The piebox animation model can be used together with any PieCSS text and color class to achieve stunning text color, background color, border color, hover and focus effects and there is more...

We believe the piebox model to be a better replacement for sliders and the slider section as it presents all your information in one without the need for a user to wait (which rarely happens) for your images to slide to reveal information, thereby it giving users the feeling you value their time a lot and in return improving on your bounce rates and retention rates. Think SEO => Think PieCSS

The piebox uses the flex model to achieve the piebox model.





**How To =>** Use `.piebox` class to identify the main section or div for your piebox and then `.piebox-screen` for each of the inner sections in the main section or div that will hold your slides(multimedia: preferably images, for this version).

Each `.piebox-screen` class should be followed by another section or div with `.piebox-content` class for your slide contents(texts, icons, and other smaller multimedia).

Text tags recommended in PieCSS for use as piebox contents in this version of PieCSS include: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, `<p>` and `<span>`.

*\*\*The piebox is made to be responsive and you should know that piebox contents can also be styled together with any of the PieCSS color and text presentation styles.*



## ANIMATION

There are several classes to use to animate elements in PieCSS and we will be diving into this shortly.

### ADD ANIMATION EFFECTS TO ANY ELEMENT

This section describes “general” animation effects, transitions and filters that can be added to any element in your web pages.

Use `.arc-animate-element` to animate any element. Default animation style is ease-in-out and transition is for 0.4s using `.arc-animate-element`.

PieCSS allows you some linear effects for use in animating elements in your document. To use the linear animation timing function, call the `.arc-animate-element-linear` class instead. Transition also lasts 0.4s.

*\*\*Feel free to use animated elements with any color and typography classes with the PieCSS framework.*



## **FADE INS AND TRANSPARENT BACKGROUND FOR ANY ELEMENT**

- To fade in on any element, use the `.fadein` class to achieve this.
- To make any element of transparent background, use the `.element-transparent` class to achieve this. To achieve this effect on hover only, use the `.element-transparent-hover` class.
- To disable effects on hover, use the `.element-hover-none` class.

### **Spin**

- To add spin to an element, use the `.element-animate-spin` class to spin element.

### **Fade**

- To fade any elements, use the `.element-animate-fading` class to do this.

## BOXES WITH SHADOWS

### The element box

The element box class is used to add box shadow to elements simply by calling the `.element-box` and `.element-box-hover` to add box shadow on hover to elements.

Here is default style declaration for the Element box – standalone and on hover

```
.element-box, .element-box-hover: hover {  
box-shadow: 0 4px 10px 0 rgba(0,0,0,0.2), 0 4px 20px 0 rgba(0,0,0,0.19);  
}
```

### Shadow box variations

#### Black

Use the `.radius .disp-shadow-black` to achieve a shadow box black color effect on your element

#### Grey

Use the `.radius .disp-shadow-grey` or `.radius .disp-shadow-gray` to achieve a shadow box grey color effect on your element. Compensation was considered for varied spelling.

## OPACITY

Opacity describes how opaque elements appear. In PieCSS the opacity function can be added to elements in two ways; full mix(fill) and partial mix(part). To create the effects using the following classes:

- Full mix(Fill) => Use `.element-opacity-fill` class to fill
- Partial mix(part) => Use `.element-opacity-part` class to fill

### Opacity on hover

Use the `.element-hover-opacity` class to achieve opacity on hover on your elements.

### Turn off opacity

Use the `.element-opacity-off` and `.element-hover-opacity-off` classes to turn off opacity on elements and their hover effects.

*\*\*Use the `.element-animate-opacity` class to achieve “a general” opacity for any element – this is aside from fill and part opacity mix.*

*\*\* Support was added for IE8 and earlier*

## GREYSCALE AND SEPIA FILTER

### Greyscale

Greyscale and sepia are animation filters that can be added to elements. “Greyscale” and “Grayscale” have been added to mean the same and to compensate for spelling. In PieCSS the greyscale fill can be added to elements in two ways; full mix(fill) and partial mix(part). To create the effect using the following classes:

- Full mix(Fill) => Use `.element-greyscale-fill` or `.element-grayscale-fill` class to fill
- Partial mix(part) => Use `.element-greyscale-part` or `.element-grayscale-part` class to fill

### Greyscale on hover

Use the `.element-hover-greyscale` or `.element-hover-grayscale` class to add rich greyscale mix on hover on your elements.

*\*\* Support was added for IE8 and earlier*

## GREYSCALE AND SEPIA FILTER

### Sepia

In PieCSS, the sepia fill can be added to elements in two ways; full mix(fill) and partial mix(part). To create the effect using the following classes:

- Full mix(Fill) => Use `.element-sepia-fill` class to fill
- Partial mix(part) => Use `.element-sepia-part` class to fill

### Sepia on hover

Use the `.element-hover-sepia` class to achieve sepia on hover on your elements.

*\*\* You can also use `.element-sepia` to achieve 70% fill*

*\*\* Support was added for IE8 and earlier*

## DIRECTION AND ZOOM

### Direction

Let us try to add awesome effect and direction to our animated elements and objects, in this case, by specifying the direction to animate.

Name	Class/Id	Duration	Effect applied
Top	.element-animate-top	0.4s	Opacity
Bottom	.element-animate-bottom	0.4s	Opacity
Left	.element-animate-left	0.4s	Opacity
Right	.element-animate-right	0.4s	Opacity

### Zoom

To zoom elements, use the `.element-animate-zoom` class to achieve the zoom function on a timing of 0.6s.



## EXTEND AN ELEMENT.

Let us try to extend an element and it's content vertically and horizontally on hover. In PieCSS, elements can be extended horizontally(by width) and vertically(by height) and also both ways.

### Extend element width on hover

This means extending elements by width and there are three ways to extend elements by width on the screen in PieCSS(that is horizontally). Now let me show you these in a tabular format.

Width extension forms	Class/Id	Starting width	Extended width
Extend over a small area	#extend-width-sm	100px	300px
Extend over a large area	#extend-width-lg	200px	600px
Extend over the whole screen	#extend-width-fill	200px	100% of screen

## Extend element height on hover

This means extending elements by height and there are three ways to extend elements by height on the screen in PieCSS(that is vertically). Now let me show you these in a tabular format.

Height extension forms	Class/Id	Starting width	Extended width
Extend over a small area	#extend-height-sm	100px	300px
Extend over a large area	#extend-height-lg	200px	600px
Extend over the whole screen	#extend-height-fill	200px	100% of screen

## Extend element width and height on hover

This means extending elements by width and height and there are three ways to extend elements by both width and height on the screen in PieCSS(that is horizontally and vertically). Now let me show you these as well.

Height extension forms	Class/Id	Starting width & height	Extended size
Extend over a small area	#extend-width-height-sm	200px	100% of screen
Extend over a large area	#extend-width-height-lg	200px	100% of screen
Extend over the whole screen	#extend-width-height-fill	200px	100% of screen

*\*\*Responsiveness and SEO friendly have been ensured for your elements and remember to use contents together with any typography and color classes for stunning background and text formats.*

## EXTEND AND TRANSFORM AN ELEMENT.

Let us try to add some interesting transformation and also, simultaneously, extend an element and its content vertically and horizontally on hover. In PieCSS, elements can be transformed + extended horizontally (by width) and vertically (by height). The transformation style adopted by the PieCSS in this early version is the *rotation* effect.

### Transform and extend element width on hover

This means extending elements by width with a transformation of the content area. There are three ways to transform + extend elements by width on the screen in PieCSS. Let's dive in

Width transform + extend forms	Class/Id	Starting width	Extended width
Transform + extend over a small area	#transform-width-sm	100px	300px
Transform + extend over a large area	#transform-width-lg	200px	600px
Transform + extend over the whole screen	#transform-width-fill	200px	100% of screen

## Transform and extend element height on hover

This means extending elements by height with a transformation of the content area. There are three ways to transform + extend elements by height on the screen in PieCSS. Let's dive in here too

Height transform + extend forms	Class/Id	Starting height	Extended height
Transform + extend over a small area	#transform-height-sm	100px	300px
Transform + extend over a large area	#transform-height-lg	200px	600px
Transform + extend over the whole screen	#transform-height-fill	200px	100% of screen

## Extend element width and height on hover

This means extending elements by width and height and there are three ways to extend elements by both width and height on the screen in PieCSS(that is horizontally and vertically). Now let me show you these as well.

Height extension forms	Class/Id	Starting width & height	Transformed size
Extend over a small area	#transform-width-height-sm	200px	100% of screen
Extend over a large area	#transform-width-height-lg	200px	100% of screen
Extend over the whole screen	#transform-width-height-fill	200px	100% of screen

*\*\*Responsiveness and SEO friendly have been ensured for your elements and remember to use contents together with any typography and color classes for stunning background and text formats.*



CONTENT



MULTIMEDIA (IMAGES, AUDIO AND VIDEO)



## IMAGES

Images are pictorial representation of data on an output device such as a screen.

PieCSS supports several image presentation forms. In PieCSS, all images have a default vertical alignment of "middle." In addition, there is a `.pie-image` class that add a maximum width of 100% and automatically increases and decreases height when needed and across devices. Always use the `.clearfix` class to clear both sides by displaying your gallery in a table layout.

Several prototypes for working with images in PieCSS include:

- Specifying a auto-increment or an auto-decrement height styling to images using the `.pie-image` class.
- Using images as background and containing them therein by applying the `.bg-image` class to an image.
- Cutting off the sides of an image and filling any spaces to preserve your intended aspect ratio. This can easily be achieved by adding the `.arc` class to your `<img>` element.
- Defining image galleries and their contents by applying the following style declarations => create a `<section>` to be used for image gallery and to hold all images and include the class `.gallery` in the main section. Doing the above instructed will create a styling for included images within the section and also a hover for all gallery items. To use description, include a `.desc` class within your `<section>` to style classes with a default `text-align:center` and `padding:14px`.

## Sliders

Sliders are used to display content in passing, one after another in a looped or non-looped movement. To display your images in a slide, simply go ahead and list your images as you would normally do and then use the `.img-slider` on your `<img>` tags. Then use in conjunction with PieJS to slide.

## Image Behind Text

To place an image behind text in a starting of a container or row, simply add the image with the `.image-behind-text` class and the text on the same container and your image will assume a `position:absolute;left:0px;top:0px;` and `z-index:-1` to place itself behind your text.

- \*\* Add typography and color classes to your gallery to achieve a mixture of stunning effects*
- \*\* Your image galleries are made SEO friendly and are responsive across devices.*

## VIDEO AND AUDIO

Use the `.arc-multimedia` class to effect video/audio multimedia styling for your multimedia container elements.

The `.arc-multimedia` class affects as well all `<object>`, `<iframe>` and `<embed>` elements within it, giving them a position of absolute with top and left margins of 0px together with a width a height specification of 100%.

*\*\* The video and audio container can also be used with the aspect ratio classes. The aspect ratio 16:9 class has been used for the `.arc-multimedia` class.*

## PROGRESS

Use the `.progress` class to style progress bars on your web page.

View the declarations that come with the `.progress` class when used:

```
.progress{  
    position:relative;  
    height:auto;  
    display:block;  
    width:100%;  
    background-color:#000080;  
    border-radius:3px;  
    margin:1px 0 2px 0;  
    overflow:hidden;  
}
```

*\*\* Try using any color class to add other stunning color effects to your `.progress` classes: these color classes should be used together with the `.progress` class.*



# WORKING WITH TYPOGRAPHY

## TEXTS, FONT STYLING AND LISTS

## WORKING WITH TEXTS, FONT STYLING AND LISTS

Texts are an integral part of any website and font/text styling makes text appealing to any website user.

*In future updates, typography(which includes texts and lists) may likely be dealt with in an entire different section as opposed being treated under “Content” section.*

PieCSS has a lot of ways to use texts and format fonts based on user requirements. An entire CSS file is dedicated to typography and can be navigated via `../piecss/file/_typo.pie.css` as a combined CSS file for texts and lists. The `_list.pie.css` will be deprecated in future PieCSS releases and hence it's use in production is not advised(use only in development and especially, for education/learning purposes)

Text classes can be used together with any of animation, color and other Pie-CSS classes to achieve a blend of awesome effects. **Use text classes within text tags such as <p>, <span>, <h1> to <h6> tags and other text related and text compliant tags and elements.**

As stated earlier, the “myPie” word attached to a class or id is used to allow user add custom values

**Now let us dive in into working with typography(texts fonts and lists) in Pie-CSS >>>**



## TEXT AND FONT STYLING

## TEXT AND FONT MANIPULATION

This section describes the various ways text can be manipulated and also shows the various font manipulations available in PieCSS.

### Text style reset

To reset text and restore text to default, use the `.text-normal` class to set text back to the PieCSS default or `.text-initial` class to reset to load initially declared font styles.

### Text align (within an aspect ratio)

`.text-align` to achieve text align with an aspect ratio or `.text-align-aspect` to achieve same with `text-align:center`;

### Text shadow

This adds shadow effects to text. Use `.text-shadow-black` to add black shadow to text and `.text-shadow-myPie` to specify your own color.

**NOTE: Add your own color first to the `.text-shadow-myPie` class before using in your web page.**

The `.text-shadow-myPie` class comes with the following declaration `.text-shadow-myPie{text-shadow:5px 5px 10px /*specify a color*/;} to specify your own color, simply replace /*specify a color*/ with a color name or color hex code or color RGB value of your choice.`



## Text styles

Other text style reset rules are displayed below in a tabular form.

Function	Class/Id	Description
Letter spacing	.text-wide	Spaces letters by 4px
Wrap white space	.text-nowrap	Disables white space wrap between texts
Text direction - right	.text-dir-right	Makes text flow on the screen from right to left
Text direction - left	.text-dir-left	Makes text flow on the screen from left to right
Indent text	.text-indent	Adds responsive text indentation rules
Align center	.text-center	Aligns text center
Align right	.text-right	Aligns text right
Align left	.text-left	Aligns text left
Justify text	.text-justify	Justifies text on screen
Font variation	.text-small-caps	All in caps but in smaller font-sizes
Text overflow	.text-overflow	Use the '=>' symbol to cut-off text overflow

## Text styles II

Function	Class/Id	Description
Makes text bold	.text-bold	Makes text appear bold on screen and print
Makes text bolder	.text-bolder	Makes text appear bolder on screen and print
Italicize text	.text-italics	Makes text appear italics on screen and print
Make text oblique	.text-oblique	Makes text appear oblique on screen and print
Extra small text	.text-x-small	Makes text of extra small size
Small text	.text-small	Makes text of small size
Medium text	.text-medium	Makes text of medium size
Large text	.text-large	Makes text of large size
Extra large text	.text-x-large	Makes text of extra large size
Vertical mode	.text-vertical-mode	Write text in vertical mode
Break word	.text-break	Break words where it appears

## Text styles III

Function	Class/Id	Description
Overline	<code>.text-overline</code>	Makes an overline across text
Underline	<code>.text-underline</code>	Makes an underline across text
Strike	<code>.text-strike</code>	Strike through text
Underline and overline	<code>.text-underline.text-overline</code>	Makes an overline and underline across text
Make line wavy	<code>.text-decorate-wavy</code>	Makes the line across text wavy in appearance
Make line dotted	<code>.text-decorate-dotty</code>	Makes the line across text dotted in appearance
Make thicker line	<code>.text-decorate-solid</code>	Makes the line across text solid in appearance
Make double line	<code>.text-decorate-double</code>	Makes the double line across text
Uppercase	<code>.text-change-upper</code>	Changes text to uppercase
Lowercase	<code>.text-change-lower</code>	Changes text to lowercase
Capitalize each word	<code>.text-change-cap</code>	Capitalizes each word

## Text colorful line decorations

Use the classes here and their declarations to style both overline and underline when used together.

Function	Class/Id	Description
Dotty	<code>.text-line-decorate-dotty</code>	Makes both overline and underline dotty
Wavy	<code>.text-line-decorate-wavy</code>	Makes both overline and underline wavy
Wavy blue	<code>.text-line-decorate-wavy-blue</code>	Makes the wavy overline and underline blue
Dotty blue	<code>.text-line-decorate-dotty-blue</code>	Makes the dotty overline and underline blue
Wavy brown	<code>.text-line-decorate-wavy-brown</code>	Make the wavy overline and underline brown
Dotty brown	<code>.text-line-decorate-dotty-brown</code>	Makes the dotty overline and underline brown
Wavy red	<code>.text-line-decorate-wavy-red</code>	Makes the wavy overline and underline red
Dotty red	<code>.text-line-decorate-dotty-red</code>	Makes the dotty overline and underline red
Wavy custom color	<code>.text-line-decorate-wavy-myPie</code>	Makes the wavy overline and underline of custom color
Dotty custom color	<code>.text-line-decorate-dotty-myPie</code>	Makes the dotty overline and underline of custom color

## Text colorful line decorations

Use the classes here and their declarations to style your underline.  
Formatted for iOS devices as well.

Function	Class/Id	Description
Purple underline	<code>.text-underline-purple</code>	Makes underline of purple color
Blue underline	<code>.text-underline-blue</code>	Makes underline of blue color
Custom color underline	<code>.text-underline-myPie</code>	Makes underline of user own specified color

**To specify your own color, simply replace `/*specify a color*/` in the `.text-underline-myPie` declarations with a color name or color hex code or color RGB value of your choice. Requires you to dig into CSS file to replace so be careful when changing/replacing values in CSS file.**



LISTS

## LISTS, LIST STYLE TYPES, LIST STYLE POSITIONS AND PREMADE THEMES

### LISTS

Lists are used to present data in an outlined and easy to understand manner.

In CSS, lists are used to present data in an outlined and easy to understand in an ordered or unordered fashion.

Pie-CSS supports unique presentation methods for lists – ordered `<ol>` and unordered `<ul>` and their respective list items `<li>`.

### Let us then dive in to all the list styling possibilities in Pie-CSS

- Firstly, to remove the list style types in your styles – ordered and unordered – include `.list-style-none` class to your `<ol>` and `<ul>` element to achieve this.
- Secondly, to display your lists inline, include the `.list-display-line` class in your `<ol>` and `<ul>` element and thereafter `.list-item-line` for each `<li>` element
- Thirdly, to make your list item marker or list style type appear outside of the `<ol>` and `<ul>` block, use the `.list-style-position-o` class to achieve this or `.list-style-position-i` to make your list style types appear inside of the `<ol>` and `<ul>` block.

## Adding list style types

There are three list style categories that can be added to your CSS and they include: Symbols, Decimals and Alphabets/Characters.

Lets dive in – shall we?

List style type - Symbols	Class/Id	Description
Square	.list-style-square	Uses the square symbol as lists style type
Circle	.list-style-circle	Uses the circle symbol as lists style type
Disc	.list-style-disc	Uses the disc symbol as lists style type

List style type - Decimal	Class/Id	Description
Decimal	.list-style-decimal	Uses whole integers as list style type
Decimal leading zero	.list-style-decimal-0	Adds a leading “0” to integers used as list style type.



## Adding list style types II

List style type – Alphabets/characters	Class/Id	Description
Georgian	.list-style-georgian	Uses Georgian letters as lists style type
Hebrew	.list-style-hebrew	Uses Hebrew letters as lists style type
Hiragana	.list-style-hiragana	Uses Hiragana symbols as lists style type
Hiragana iroha	.list-style-hiragana-iroha	Uses Hiragana iroha symbols as lists style type
Katakana	.list-style-katakana	Uses Katakana symbols as lists style type
Katakana iroha	.list-style-katakana-iroha	Uses Katakana iroha symbols as lists style type
Khz	.list-style-khz	*Uses Khz symbols as lists style type – may not function on some devices

### Adding list style types III

List style type – Alphabets/characters	Class/Id	Description
Upper alpha	.list-style-upper-a	Uses English alphabets in uppercase
Lower alpha	.list-style-lower-a	Uses English alphabets in lowercase
Upper Latin	.list-style-upper-l	Uses Latin alphabets in uppercase
Lower Latin	.list-style-lower-l	Uses Latin alphabets in lowercase
Upper Roman	.list-style-upper-r	Uses Roman alphabets in uppercase
Lower Roman	.list-style-lower-r	Uses Roman alphabets in lowercase
Upper Greek	.list-style-upper-g	Uses Greek alphabets in uppercase
Lower Greek	.list-style-lower-g	Uses Greek alphabets in lowercase

## PREMADE THEMES

There are premade themes for your lists that include background colors and text colors. Make your own theme by using any of the pie.css color classes together with your lists to achieve more stunning decoration on your lists.

*The table that follows shows premade themes. All premade themes apply the inline-block display to list*

Theme name	Class/Id
Slate	<code>.radius-theme-list-slate</code>
Royal blue	<code>.radius-theme-list-blue</code>
Black	<code>.radius-theme-list-black</code>
Crimson	<code>.radius-theme-list-crimson</code>
White	<code>.radius-theme-list-white</code>
Custom theme	<code>.radius-theme-list-myPie</code>
Neutral	<code>.radius-theme-list</code>



## DECLARE NEW FONTS

Fonts are pattern of text on your website.

Fonts allow for readability and flowability across devices. The Pie-CSS framework comes with declared fonts and where these fonts are from third party sources, license information and documentation from the owners is included.

The declared fonts in PieCSS v1.0 are shown thus:

- Abeeze (abeezee-Italic and abeezee-Regular)
- Homoarak
- Loopy (loopy\_IT and loopy\_\_\_\_)
- Trashco

The documentation and license information from the authors are contained in the respective folders of the declared fonts. See **/piecss/font/**



# WORKING WITH FORMS

Understanding the PieCSS forms and buttons

## FORM AND BUTTONS

Forms is an element that is capable of accepting inputs from users, and, buttons are used to effect an action, on a form or otherwise outside of a form.

See these elements apiece

### FORM

Define a container or row to build your form in and note the following defaults and other points:

- `<input>` elements occupy 100% of the containing element with intended spacing(inner space of 12px 20px and outer space of 8px)
- Use any color and `:focus` color classes all your form elements and sub-elements
- Use any typography class and color class on your `<label>` elements
- `<textarea>` elements have a width of 100% with a height of 150px and intended spacing(inner space of 12px 20px and outer space of 8px with the ability to resize by your form users. To disable the ability to resize your `<textarea>` element, use the `.no-resize` class)
- `<select>` elements occupy a width of 100% of containing row and inner spacing of 16px 20px and automatically adjusted heights
- All form elements and sub-elements are adjusted for responsiveness.

## BUTTONS

Following is the default declaration for all button element.

```
button, input[type=button], input[type=submit], input[type=reset], .pie-btn, .radius-btn{  
    border:none;  
    display:inline-block;  
    margin:4px 2px;  
    padding:8px 16px;  
    border-radius:4px;  
    vertical-align:middle;  
    overflow:hidden;  
    text-decoration:none;  
    text-align:center;  
    cursor:pointer;  
    white-space:nowrap;  
}
```

For hover effects on buttons, use the `.pie-btn` or the `.radius-btn` to achieve significant stand-alone hover effects.

To disable button, use the `.pie-disabled` class

## BUTTONS II

### Other button rules

Define a container or row to hold your button in and note the following defaults and other points:

- Buttons should be used together with text and color classes to achieve maximum decoration effect
- If you do not wish to import PieCSS color and text classes for your buttons, you may use this default button - `#button-pie` => One class rules all - expected to be removed in future releases.
- To disable user select effect on button, use the `.pie-btn-select` or `.radius-btn-select` classes
- To maintain normal whitespace on your button, use the `.arc` `.button` class
- Use together with any typography class
- All buttons are adjusted for responsiveness.





# WORKING WITH DISPLAY

Understanding the PieCSS display modes

## GUIDE TO DISPLAY MODE AND POSITION

In this version of the PieCSS, we learn the various display modes and display position that elements can be represented with and in.

*\*\*More displays is expected to be added in future revisions of the PieCSS framework*

Display mode/position	Class/Id	Description
Top	.element-disp-top	Displays content at top position
Bottom	.element-disp-bottom	Displays element at bottom position
Overlay	.element-disp-overlay	Overlay an element
Top left	.element-disp-topleft	Displays content at top left position
Top right	.element-disp-topright	Displays content at top right position
Bottom left	.element-disp-bottomleft	Displays content at bottom left position
Bottom right	.element-disp-bottomright	Displays content at bottom right position

## Guide to display mode and position II

Display mode/position	Class/Id	Description
Middle	.element-disp-middle	Displays content at middle center position at page
Left	.element-disp-left	Displays element at left center position
Right	.element-disp-right	Displays element at right center position
Top middle	.element-disp-topmiddle	Displays content at top middle position
Bottom middle	.element-disp-bottom-middle	Displays content at bottom middle position
Block	.display-block	Displays content in block mode
Block on hover	.display-hover-block	Displays content in block mode on hover
Element display position	.element-disp-position	Assigns a position:absolute to elements and contents
No display	.element-disp-none	Used to hide contents

## Guide to display mode and position III

Circle and rounded corner displays: can also be used to create badges and similar items

Display mode	Class/Id	Description
Circle	.circle	Displays container/element in a circle
Border radius - small	.round-s	Displays container/element with small rounded edges
Border radius - medium	.round-m	Displays container/element with fairly noticeable rounded edges
Border radius - large	.round-l	Displays container/element with well noticeable rounded edges
Border radius – xtra large	.round-xl	Displays container/element with extra large rounded edges
Border radius – very large	.round-xxl	Displays container/element with one of the largest rounded edges

## Guide to display mode and position IV

Prevent elements from floating on any sides using the “clear” classes.

Clear position	Class/Id	Description
None	.clear-none	Maintains the natural/induced floats of the document
Both sides	.clear-both	Prevent floats on both sides of your web page
Left	.clear-left	Prevent floats on the left side of your web page
Right	.clear-right	Prevent floats on the right side of your web page
Inherit from a radius	.arc .clear-both	Inherit float property from a radius
Reset	.reset-clear	Reset the float property of elements

## TOOLTIP

Use the `.pie-tooltip` class to markout a tooltip with the following declarations inbuilt:

```
.pie-tooltip{  
    padding:10px 8px;  
    font-size:1rem;  
    z-index:2000;  
    background-color:transparent;  
    border-radius:2px;  
    color:#000;  
    min-height:36px;  
    line-height:1rem;  
    opacity:0;  
    display:none;  
    position:absolute;  
    left:0;  
    top:0;  
    overflow:hidden;  
}
```



# THE BOX MODEL

Using the CSS box model with Pie



## BOX MODEL

The CSS box model defines how the dimensions of HTML elements are modelled in browser engines. The box model is a concept in the composition of webpages.

Basically the CSS box model is composed of these four concepts:

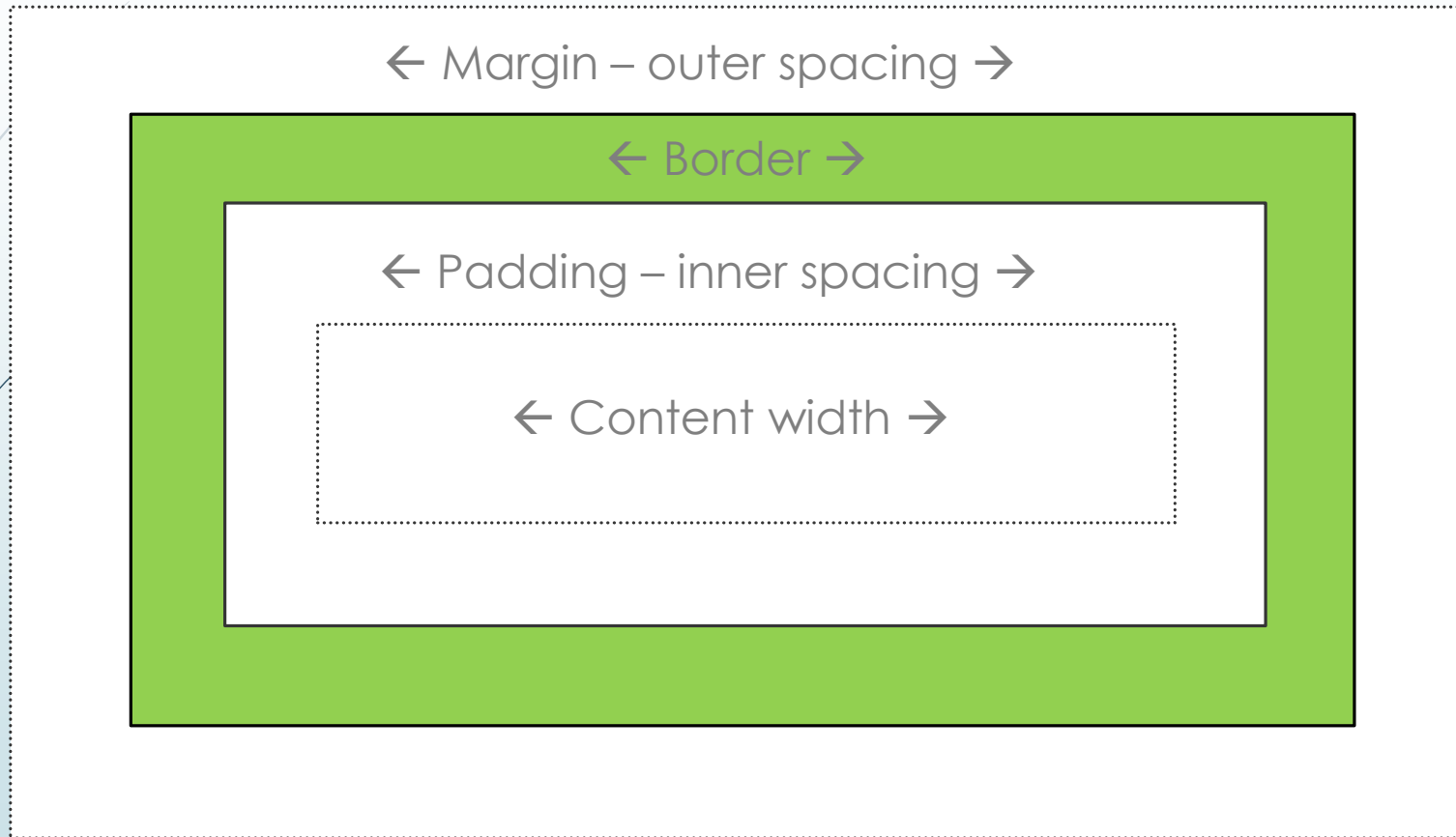
- Width
- Padding (incl. top, left, bottom, right paddings)
- Border (incl. top, left, bottom, right border)
- Margin (incl. top, left, bottom, right margins)

### Important notes

- To disable margins on any element, use the `.nomargin` class
- To disable paddings on any element, use the `.nopadding` class
- To disable border on any element, use the `.noborder` class
- Borders can be used together with any `pie.css` color classes to add stunning decorations
- Also any element can be used together with any box element to achieve the box model effect.

Let us now dive into these at the Pie-CSS framework beginning with margin, padding and border





CSS box model graphical explanation

## WIDTH

Width is the size of the containing vessel for your elements  
Please refer to box model for graphical explanation.

### Width manipulations

While there are row/container classes to define rows, there have been included “width only” classes to influence width in your CSS classes and HTML elements.

All width-only definitions have been fixed for responsiveness

Width size	Class/Id	Description
100%	.sector-width	Makes an element full width
75%	.sector-width-75	Gives an element width of 75%
50%	.sector-width-50	Gives an element width of 50%
33.33%	.sector-width-33	Gives an element width of 33.33%
25%	.sector-width-25	Gives an element width of 25%
20%	.sector-width-20	Gives an element width of 20%

## MARGIN

Margins are the outermost space surrounding your HTML element.  
Please refer to box model for graphical explanation.

### Small margin

Creates small margin spaces at specified sides and around all sides of HTML elements using CSS.

Margin small	Class/Id	Description
All round margin	.margin-s	Clears a small margin space around all sides of element
Margin at top	.margin-top-s	Clears a small margin space at top of element
Margin at bottom	.margin-bottom-s	Clears a small margin space at bottom of element
Margin at left	.margin-left-s	Clears a small margin space at left side of element
Margin at right	.margin-right-s	Clears a small margin space at right side of element

## Medium margin

Creates medium margin spaces at specified sides and around all sides of HTML elements using CSS.

Margin medium	Class/Id	Description
All round margin	.margin-m	Clears medium margin space around all sides of element
Margin at top	.margin-top-m	Clears medium margin space at top of element
Margin at bottom	.margin-bottom-m	Clears medium margin space at bottom of element
Margin at left	.margin-left-m	Clears medium margin space at left side of element
Margin at right	.margin-right-m	Clears medium margin space at right side of element

## Large margin

Creates large margin spaces at specified sides and around all sides of HTML elements using CSS.

Margin large	Class/Id	Description
All round margin	.margin-l	Clears a large margin space around all sides of element
Margin at top	.margin-top-l	Clears a large margin space at top of element
Margin at bottom	.margin-bottom-l	Clears a large margin space at bottom of element
Margin at left	.margin-left-l	Clears a large margin space at left side of element
Margin at right	.margin-right-l	Clears a large margin space at right side of element

## Extra Large margin

Creates very large margin spaces at specified sides and around all sides of HTML elements using CSS.

Margin xtra large	Class/Id	Description
All round margin	.margin-xl	Clears very large margin space around all sides of element
Margin at top	.margin-top-xl	Clears very large margin space at top of element
Margin at bottom	.margin-bottom-xl	Clears very large margin space at bottom of element
Margin at left	.margin-left-xl	Clears very large margin space at left side of element
Margin at right	.margin-right-xl	Clears very large margin space at right side of element

## PADDING

Paddings are inner spaces surrounding the width of your HTML element. Please refer to box model for graphical explanation.

### Small padding

Creates small padding spaces at specified sides and around all sides of HTML elements using CSS.

Padding small	Class/Id	Description
All round padding	.padding-s	Clears a small padding space around all sides of element
Padding at top	.padding-top-s	Clears a small padding space at top of element
Padding at bottom	.padding-bottom-s	Clears a small padding space at bottom of element
Padding at left	.padding-left-s	Clears a small padding space at left side of element
Padding at right	.padding-right-s	Clears a small padding space at right side of element

## Medium sized padding

Creates medium sized padding at specified sides and around all sides of HTML elements using CSS.

Padding medium	Class/Id	Description
All round padding	.padding-m	Clears medium sized padding around all sides of element
Padding at top	.padding-top-m	Clears medium sized padding at top of element
Padding at bottom	.padding-bottom-m	Clears medium sized padding at bottom of element
Padding at left	.padding-left-m	Clears medium sized padding at left side of element
Padding at right	.padding-right-m	Clears medium sized padding at right side of element



## Large sized padding

Creates large sized padding at specified sides and around all sides of HTML elements using CSS.

Padding large	Class/Id	Description
All round padding	.padding-l	Clears large sized padding around all sides of element
Padding at top	.padding-top-l	Clears large sized padding at top of element
Padding at bottom	.padding-bottom-l	Clears large sized padding at bottom of element
Padding at left	.padding-left-l	Clears large sized padding at left side of element
Padding at right	.padding-right-l	Clears large sized padding at right side of element

## Extra large sized padding

Creates very large sized padding at specified sides and around all sides of HTML elements using CSS.

Padding Xtra large	Class/Id	Description
All round padding	.padding-xl	Clears very large sized padding around all sides of element
Padding at top	.padding-top-xl	Clears very large sized padding at top of element
Padding at bottom	.padding-bottom-xl	Clears very large sized padding at bottom of element
Padding at left	.padding-left-xl	Clears very large sized padding at left side of element
Padding at right	.padding-right-xl	Clears very large sized padding at right side of element

## BORDER

Borders are lines(thick and thin) surrounding the width of your HTML element immediately after your padding.

Please refer to box model for graphical explanation.

### Small sized border

Creates thin sized border at specified sides and around all sides of HTML elements using CSS.

Border small	Class/Id	Description
All round border	.border-s	Draws a thin sized border around all sides of element
Border at top	.border-top-s	Draws a thin sized border at top of element
Border at bottom	.border-bottom-s	Draws a thin sized border at bottom of element
Border at left	.border-left-s	Draws a thin sized border at left side of element
Border at right	.border-right-s	Draws a thin sized border at right side of element

## Medium sized border

Creates medium sized border at specified sides and around all sides of HTML elements using CSS.

Border medium	Class/Id	Description
All round border	.border-m	Draws a medium sized border around all sides of element
Border at top	.border-top-m	Draws a medium sized border at top of element
Border at bottom	.border-bottom-m	Draws a medium sized border at bottom of element
Border at left	.border-left-m	Draws a medium sized border at left side of element
Border at right	.border-right-m	Draws a medium sized border at right side of element

## Large sized border

Creates large sized border at specified sides and around all sides of HTML elements using CSS.

Border large	Class/Id	Description
All round border	.border-l	Draws a large sized border around all sides of element
Border at top	.border-top-l	Draws a large sized border at top of element
Border at bottom	.border-bottom-l	Draws a large sized border at bottom of element
Border at left	.border-left-l	Draws a large sized border at left side of element
Border at right	.border-right-l	Draws a large sized border at right side of element

## Extra Large sized border

Creates very large sized border at specified sides and around all sides of HTML elements using CSS.

Border Xtra large	Class/Id	Description
All round border	.border-xl	Draws very large sized border around all sides of element
Border at top	.border-top-xl	Draws very large sized border at top of element
Border at bottom	.border-bottom-xl	Draws very large sized border at bottom of element
Border at left	.border-left-xl	Draws very large sized border at left side of element
Border at right	.border-right-xl	Draws very large sized border at right side of element



# WORKING WITH BACKGROUND

Understanding the PieCSS background display modes

## BACKGROUNDS

This sub-section will explore the various possible ways in Pie-CSS to present backgrounds in various activities using elements in CSS.

### Background clips

The background clip is used to specify how far the background (color or image) of an element should extend within the container element.

The pie.css framework supports the following clips:

- Dotty (using class of `.bg-clip-dotty`)
- Padbox (using class of `.bg-clip-padbox`)
- Contentbox (using class of `.bg-clip-contentbox`)

Using the `.arc` `.bg-clip-contentbox` classes will inherit the parent BG clip style

### Reset background clips

Use the `.bg-clip` to reset background clip to an initially declared or undeclared state.



## **Background images**

To specify a background image for an element/container, open the pie css folder and look for the .bg-image class and replace the /\*specify image name\*/ comment with the specific image name and type, for example: my-image.jpg and if image is in a folder, also specify folder name for example: folder/my-image.jpg.

## **Animate background images**

To specify an animated background image for an element/container, open the pie css folder and look for the .bg-image-animate class and replace the /\*specify image name\*/ comment with the specific image name and type, for example: my-image.jpg and if image is in a folder, also specify folder name for example: folder/my-image.jpg.

Background image for any element in piecss v1.0 discourages a repeat so use an image large enough to cover content area if to cover content area is in the intention. Also worthy of note in piecss v1.0 is that image background attachment is set to fixed and size is set to auto.



# WORKING WITH COLOR

PieCSS in colors



## COLORS

Colors are the different shades of beauty that elements are immersed in. The PieCSS framework allows you to define several color shades to be used on any element and together with any CSS classes on your document.

### Color notes

Please note the following when working with colors in Pie-CSS

- A variety of colors are supported in Pie-CSS
- You have a chance to specify your own colors where the `–myPie` appears in a color class
- Colors are meant to be used on elements
- Colors can be used together with other CSS classes where the color effect is intended
- More colors will be supported in future revisions



## COLOR APPLICATION

The following areas have classes that applies color in this version of the pie.css framework.

- Background color and background color on hover
- Text color and text color on hover
- Border color and border color on hover
- Caret color
- Color on focus
- Column color – works with the column gap and other areas where a color is needed in between columns
- To add custom color, look for the class names appended with the `-myPie` suffix and replace the comments inside the declarations bracket with any color name/code or `none` if you want no colors.

## COLOR NAMES

The following color category and color ships with this version of software:

<b>Amber</b>	<b>Black</b>	<b>Blue</b>	<b>Brown</b>	<b>Cyan</b>	<b>Green</b>	<b>Indigo</b>	<b>Khaki</b>	<b>Lime</b>	<b>Orange</b>
Amber	Black	Blue	Brown	Cyan	Green	Indigo	Khaki	Lime	Orange
Amber light	Dark grey/gray	Blue grey	Dark brown	Dark cyan	Dark green	Dark Indigo	Dark khaki	Dark lime	Coral
Amber dark	Grey/gray	Light blue	Light brown	Light cyan	Forest green	Light Indigo	Light khaki	Light lime	Dark orange
		Steel blue	Maroon		Light green				Light orange
		Pale blue	Sand		Olive drab				Orange red
					Pale green				Tomato
					Teal				

## COLOR NAMES II

The following continuation color category and color ships with this version of software:

<b>Pink</b>	<b>Purple</b>	<b>Red</b>	<b>Yellow</b>	<b>White</b>	<b>Your custom color</b>
Pink	Purple	Red	Yellow	White	??
Dark pink	Dark purple	Crimson	Gold	Snow white	??
Hot pink	Lavendar	Indian red	Moccasin		??
Light pink	Orchid	Light red	Pale yellow		??
	Violet	Pale red			??

## COLOR GRADIENTS

Color gradients (more appropriately, linear gradients) are implemented in the piecss as a combination of all the colors in a color category combined to form a color gradient. For example:

In the amber category, **Amber & Amber light & Amber dark** combine to form a linear gradient for amber color category and so forth for all the other colors categories.

Color gradients are in two forms namely: **Horizontal bands combination (class ends with -h and moves from top to bottom)** and **vertical bands combination (class ends with -v and moves from right to left)**.

The gradient classes will format your background with the specified color combination. You still need to include a text color class if needed or background will inherit text color from document.

CSS classes for the color gradients are listed thus:

Amber	Black	Blue	Brown	Cyan	Green	Indigo	Khaki	Lime	Orange
.linear-gradient-amber-h	.linear-gradient-black-h	.linear-gradient-blue-h	.linear-gradient-brown-h	.linear-gradient-cyan-h	.linear-gradient-green-h	.linear-gradient-indigo-h	.linear-gradient-khaki-h	.linear-gradient-lime-h	.linear-gradient-orange-h
.linear-gradient-amber-v	.linear-gradient-black-v	.linear-gradient-blue-v	.linear-gradient-brown-v	.linear-gradient-cyan-v	.linear-gradient-green-v	.linear-gradient-indigo-v	.linear-gradient-khaki-v	.linear-gradient-lime-v	.linear-gradient-orange-v

## COLOR GRADIENTS II

The following continuation color category and color ships with this version of software:

The `.linear-gradient-myPie-h` and `.linear-gradient-myPie-v` classes works by first opening the `pie.css` file or any of the color files and carefully replacing the `/*Specify a color*/` comments with colors of your choice (also can delete up to two of the comments to replace up to two colors) before calling the css classes in your document.

<b>Pink</b>	<b>Purple</b>	<b>Red</b>	<b>Yellow</b>	<b>Your custom color</b>
<code>.linear-gradient-pink-h</code>	<code>.linear-gradient-purple-h</code>	<code>.linear-gradient-red-h</code>	<code>.linear-gradient-yellow-h</code>	<code>.linear-gradient-myPie-h</code>
<code>.linear-gradient-pink-v</code>	<code>.linear-gradient-purple-v</code>	<code>.linear-gradient-red-v</code>	<code>.linear-gradient-yellow-v</code>	<code>.linear-gradient-myPie-v</code>

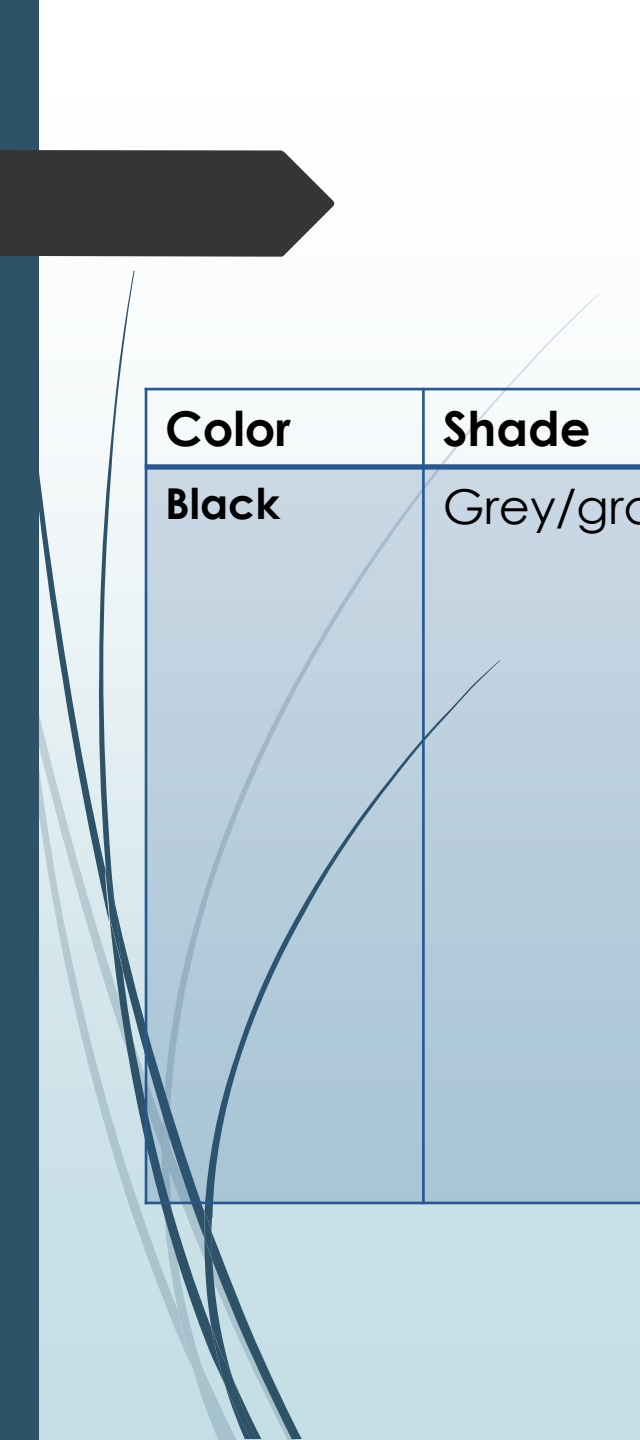


## COLOR CLASSES

Use the following classes to achieve color categories and colors that ships with this version of software:

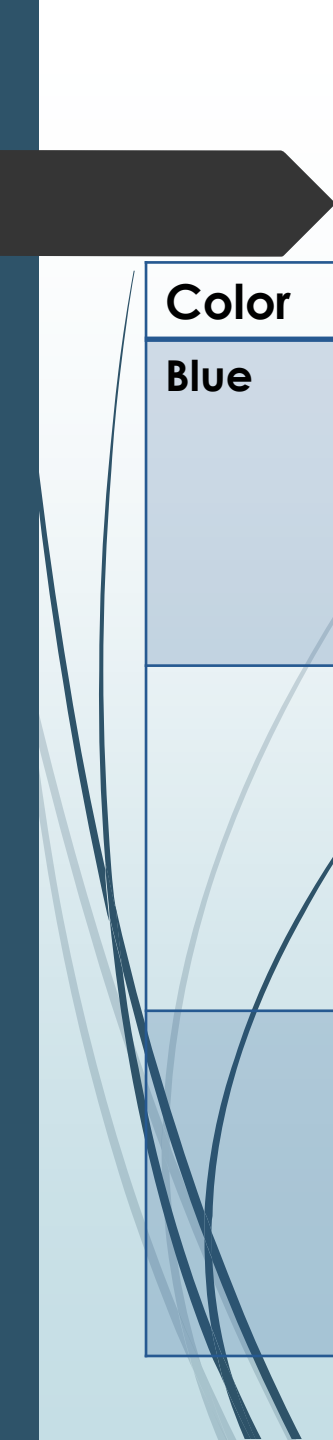
Color	Shade	Class/Id
Amber	Amber	<code>.color-amber, .color-hover-amber</code> <code>.text-color-amber, .text-color-hover-amber</code> <code>.border-color-amber, .border-color-hover-amber</code> <code>.caretcolor-amber</code> <code>.color-focus-amber</code> <code>.col-amber</code>
	Amber dark	<code>.color-amberdark, .color-hover-amberdark</code> <code>.text-color-amberdark, .text-color-hover-amberdark</code> <code>.border-color-amberdark, .border-color-hover-amberdark</code> <code>.caretcolor-amberdark</code> <code>.color-focus-amberdark</code> <code>.col-amberdark</code>
	Amber light	<code>.color-amberlight, .color-hover-amberlight</code> <code>.text-color-amberlight, .text-color-hover-amberlight</code> <code>.border-color-amberlight, .border-color-hover-amberlight</code> <code>.caretcolor-amberlight</code> <code>.color-focus-amberlight</code> <code>.col-amberlight</code>

Color	Shade	Class/Id
Black	Black	.color-black,.color-hover-black .text-color-black,.text-color-hover-black .border-color-black,.border-color-hover-black .caretcolor-black .color-focus-black .col-black
	Dark grey/gray	.color-darkgrey,.color-hover-darkgrey .text-color-darkgrey,.text-color-hover-darkgrey .border-color-darkgrey,.border-color-hover-darkgrey .caretcolor-darkgrey .color-focus-darkgrey .col-darkgrey  GREY WORKS SAME AS GRAY .color-darkgray,.color-hover-darkgray .text-color-darkgray,.text-color-hover-darkgray .border-color-darkgray,.border-color-hover-darkgray .caretcolor-darkgray .color-focus-darkgaey .col-darkgray

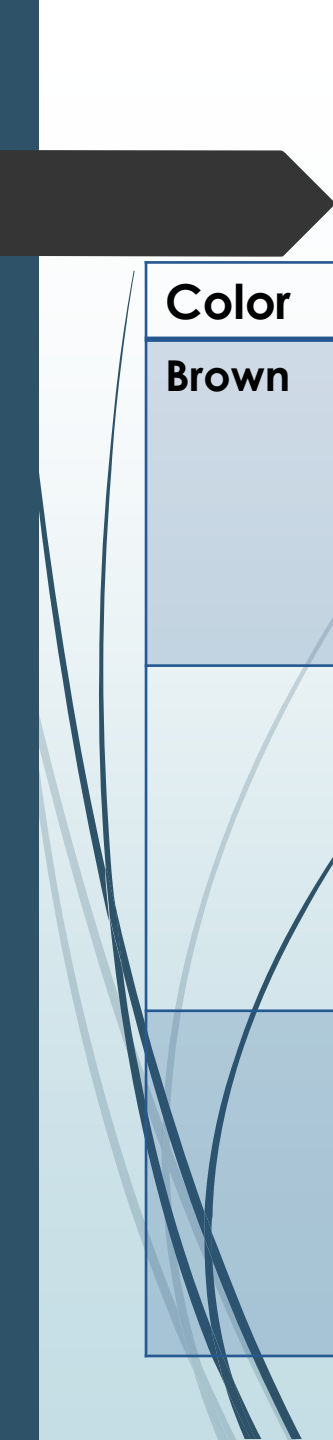


Color	Shade	Class/Id
Black	Grey/gray	<code>.color-grey,.color-hover-grey</code> <code>.text-color-grey,.text-color-hover-grey</code> <code>.border-color-grey,.border-color-hover-grey</code> <code>.caretcolor-grey</code> <code>.color-focus-grey</code> <code>.col-grey</code>  GREY WORKS SAME AS GRAY <code>.color-gray,.color-hover-gray</code> <code>.text-color-gray,.text-color-hover-gray</code> <code>.border-color-gray,.border-color-hover-gray</code> <code>.caretcolor-gray</code> <code>.color-focus-gray</code> <code>.col-gray</code>

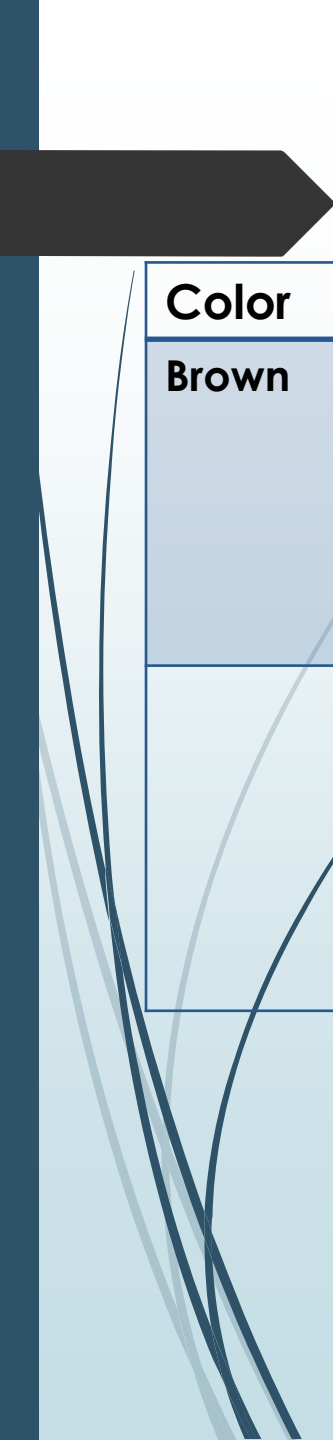
Color	Shade	Class/Id
Blue	Blue	.color-blue,.color-hover-blue .text-color-blue,.text-color-hover-blue .border-color-blue,.border-color-hover-blue .caretcolor-blue .color-focus-blue .col-blue
	Blue grey/gray	.color-bluegray,.color-hover-bluegray .text-color-bluegray,.text-color-hover-bluegray .border-color-bluegray,.border-color-hover-bluegray .caretcolor-bluegray .color-focus-bluegray .col-bluegray  GREY WORKS SAME AS GRAY .color-bluegrey,.color-hover-bluegrey .text-color-bluegrey,.text-color-hover-bluegrey .border-color-bluegrey,.border-color-hover-bluegrey .caretcolor-bluegrey .color-focus-bluegrey .col-bluegrey



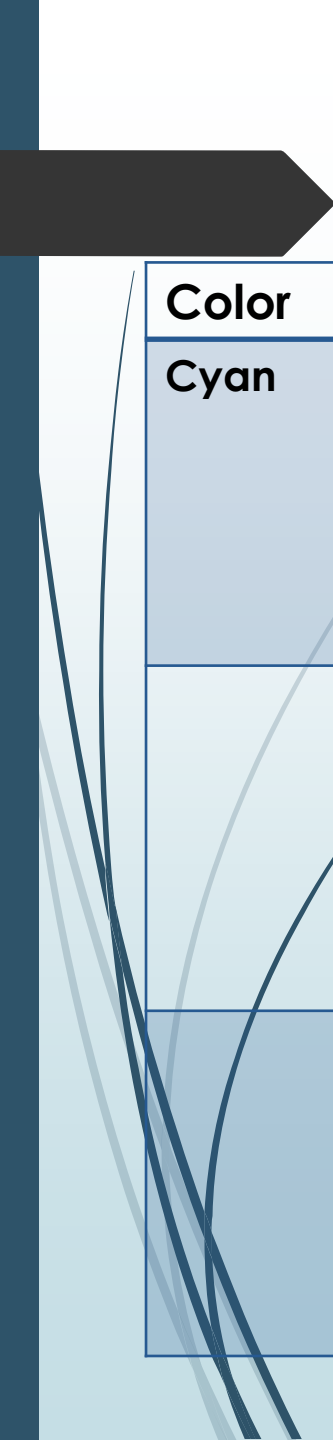
Color	Shade	Class/Id
Blue	Light blue	<code>.color-lightblue, .color-hover-lightblue</code> <code>.text-color-lightblue, .text-color-hover-lightblue</code> <code>.border-color-lightblue, .border-color-hover-lightblue</code> <code>.caretcolor-lightblue</code> <code>.color-focus-lightblue</code> <code>.col-lightblue</code>
	Steel blue	<code>.color-steelblue, .color-hover-steelblue</code> <code>.text-color-steelblue, .text-color-steelhover-steelblue</code> <code>.border-color-steelblue, .border-color-hover-steelblue</code> <code>.caretcolor-steelblue</code> <code>.color-focus-steelblue</code> <code>.col-steelblue</code>
	Pale blue	<code>.color-paleblue, .hover-color-hover-paleblue</code> <code>.text-color-paleblue, .text-color-hover-paleblue</code> <code>.border-color-paleblue, .border-color-hover-paleblue</code> <code>.caretcolor-paleblue</code> <code>.color-focus-paleblue</code> <code>.col-paleblue</code>



Color	Shade	Class/Id
Brown	Brown	<code>.color-brown, .color-hover-brown</code> <code>.text-color-brown, .text-color-hover-brown</code> <code>.border-color-brown, .border-color-hover-brown</code> <code>.caretcolor-brown</code> <code>.color-focus-brown</code> <code>.col-brown</code>
	Dark brown	<code>.color-darkbrown, .color-hover-darkbrown</code> <code>.text-color-darkbrown, .text-color-hover-darkbrown</code> <code>.border-color-darkbrown, .border-color-hover-darkbrown</code> <code>.caretcolor-darkbrown</code> <code>.color-focus-darkbrown</code> <code>.col-darkbrown</code>
	Light brown	<code>.color-lightbrown, .color-hover-lightbrown</code> <code>.text-color-lightbrown, .text-color-hover-lightbrown</code> <code>.border-color-lightbrown, .border-color-hover-lightbrown</code> <code>.caretcolor-lightbrown</code> <code>.color-focus-lightbrown</code> <code>.col-lightbrown</code>

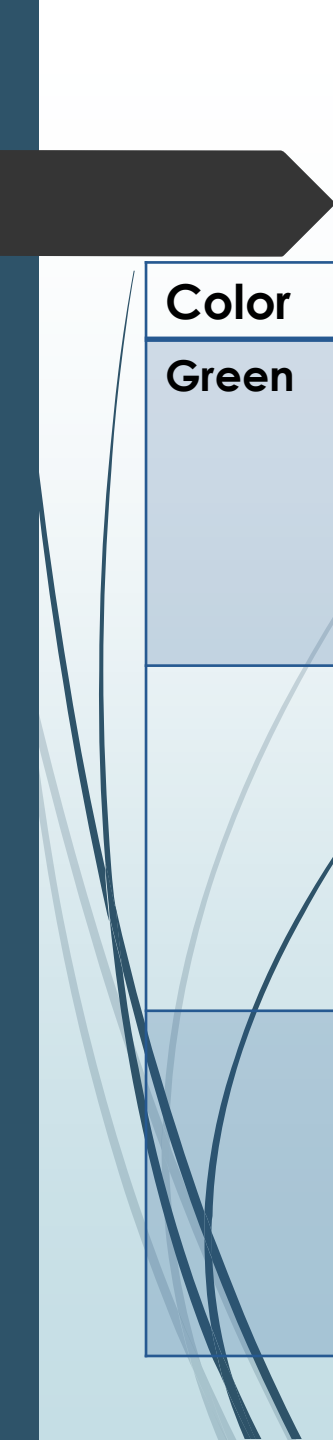


Color	Shade	Class/Id
Brown	Maroon	<code>.color-maroon, .color-hover-maroon</code> <code>.text-color-maroon, .text-color-hover-maroon</code> <code>.border-color-maroon, .border-color-hover-maroon</code> <code>.caretcolor-maroon</code> <code>.color-focus-maroon</code> <code>.col-maroon</code>
	Sand	<code>.color-sand, .color-hover-sand</code> <code>.text-color-sand, .text-color-hover-sand</code> <code>.border-color-sand, .border-color-hover-sand</code> <code>.caretcolor-sand</code> <code>.color-focus-sand</code> <code>.col-sand</code>

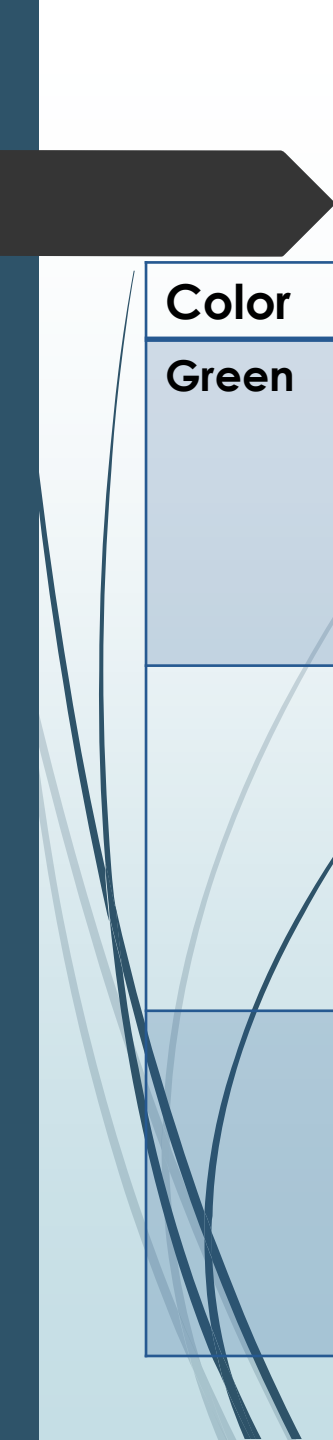


Color	Shade	Class/Id
Cyan	Cyan	<code>.color-cyan, .color-hover-cyan</code> <code>.text-color-cyan, .text-color-hover-cyan</code> <code>.border-color-cyan, .border-color-hover-cyan</code> <code>.caretcolor-cyan</code> <code>.color-focus-cyan</code> <code>.col-cyan</code>
	Dark cyan	<code>.color-darkcyan, .color-hover-darkcyan</code> <code>.text-color-darkcyan, .text-color-hover-darkcyan</code> <code>.border-color-darkcyan, .border-color-hover-darkcyan</code> <code>.caretcolor-darkcyan</code> <code>.color-focus-darkcyan</code> <code>.col-darkcyan</code>
	Light cyan	<code>.color-lightcyan, .color-hover-lightcyan</code> <code>.text-color-lightcyan, .text-color-hover-lightcyan</code> <code>.border-color-lightcyan, .border-color-hover-lightcyan</code> <code>.caretcolor-lightcyan</code> <code>.color-focus-lightcyan</code> <code>.col-lightcyan</code>

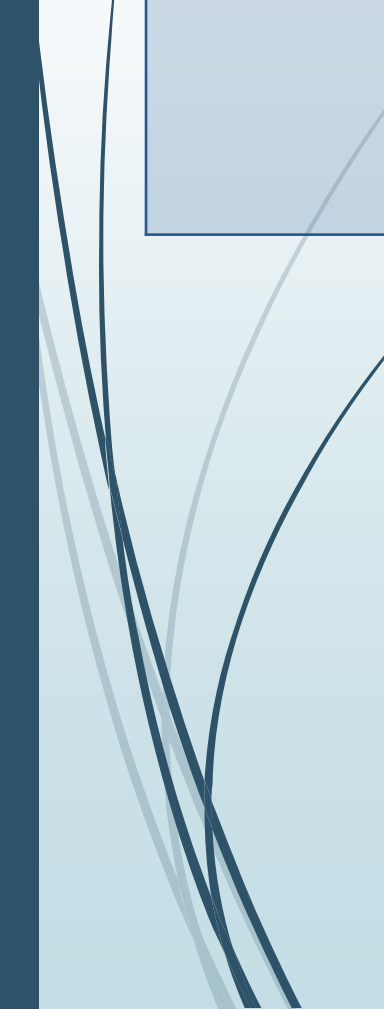





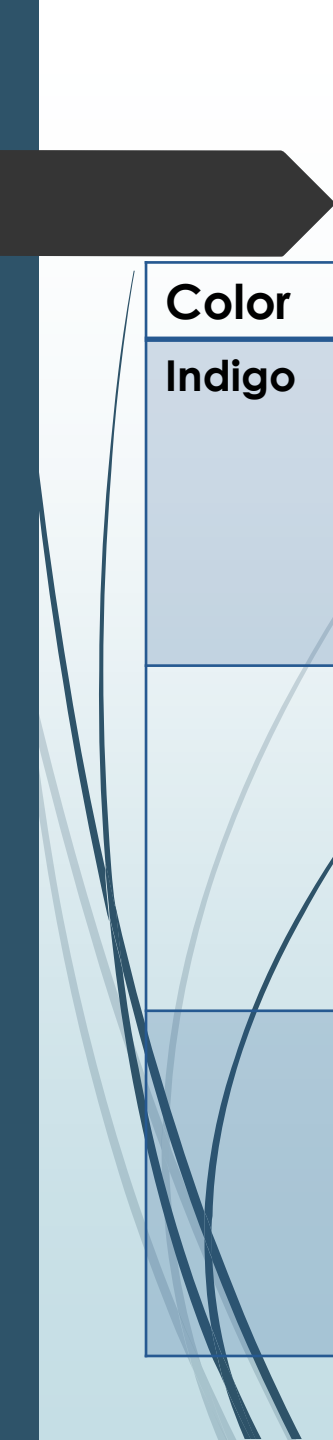
Color	Shade	Class/Id
Green	Green	<code>.color-green, .color-hover-green</code> <code>.text-color-green, .text-color-hover-green</code> <code>.border-color-green, .border-color-hover-green</code> <code>.caretcolor-green</code> <code>.color-focus-green</code> <code>.col-green</code>
	Dark green	<code>.color-darkgreen, .color-hover-darkgreen</code> <code>.text-color-darkgreen, .text-color-hover-darkgreen</code> <code>.border-color-darkgreen, .border-color-hover-darkgreen</code> <code>.caretcolor-darkgreen</code> <code>.color-focus-darkgreen</code> <code>.col-darkgreen</code>
	Forest green	<code>.color-forestgreen, .color-hover-forestgreen</code> <code>.text-color-forestgreen, .text-color-hover-forestgreen</code> <code>.border-color-forestgreen, .border-color-hover-forestgreen</code> <code>.caretcolor-forestgreen</code> <code>.color-focus-forestgreen</code> <code>.col-forestgreen</code>



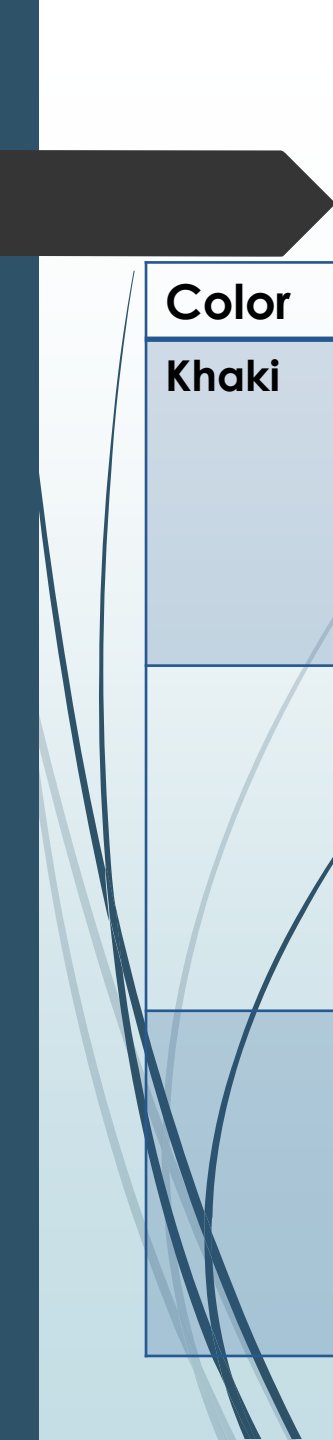
Color	Shade	Class/Id
Green	Light green	<code>.color-lightgreen, .color-hover-lightgreen</code> <code>.text-color-lightgreen, .text-color-hover-lightgreen</code> <code>.border-color-lightgreen, .border-color-hover-lightgreen</code> <code>.caretcolor-lightgreen</code> <code>.color-focus-lightgreen</code> <code>.col-lightgreen</code>
	Olivedrab	<code>.color-olivedrab, .color-hover-olivedrab</code> <code>.text-color-olivedrab, .text-color-hover-olivedrab</code> <code>.border-color-olivedrab, .border-color-hover-olivedrab</code> <code>.caretcolor-olivedrab</code> <code>.color-focus-olivedrab</code> <code>.col-olivedrab</code>
	Pale green	<code>.color-palegreen, .color-hover-palegreen</code> <code>.text-color-palegreen, .text-color-hover-palegreen</code> <code>.border-color-palegreen, .border-color-hover-palegreen</code> <code>.caretcolor-palegreen</code> <code>.color-focus-palegreen</code> <code>.col-palegreen</code>



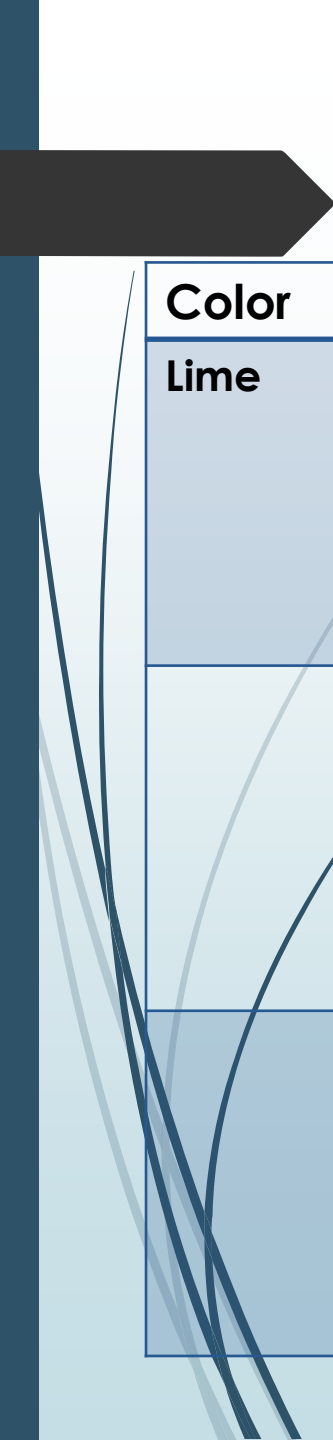
Color	Shade	Class/Id
Green	Teal	<code>.color-teal, .color-hover-teal</code> <code>.text-color-teal, .text-color-hover-teal</code> <code>.border-color-teal, .border-color-hover-teal</code> <code>.caretcolor-teal</code> <code>.color-focus-teal</code> <code>.col-teal</code>



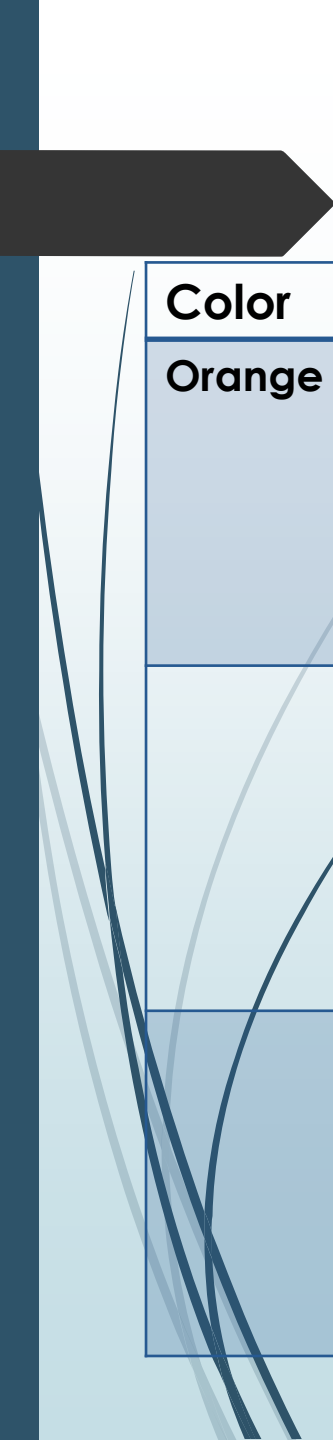
Color	Shade	Class/Id
Indigo	Indigo	<code>.color-indigo, .color-hover-indigo</code> <code>.text-color-indigo, .text-color-hover-indigo</code> <code>.border-color-indigo, .border-color-border-hover-indigo</code> <code>.caretcolor-indigo</code> <code>.color-focus-indigo</code> <code>.col-indigo</code>
	Dark Indigo	<code>.color-darkindigo, .color-hover-darkindigo</code> <code>.text-color-darkindigo, .text-color-hover-darkindigo</code> <code>.border-color-dark-indigo, .border-color-hover-darkindigo</code> <code>.caretcolor-darkindigo</code> <code>.color-focus-darkindigo</code> <code>.col-darkindigo</code>
	Light Indigo	<code>.color-lightindigo, .color-hover-lightindigo</code> <code>.text-color-lightindigo, .text-color-hover-lightindigo</code> <code>.border-color-lightindigo, .border-color-hover-lightindigo</code> <code>.caretcolor-lightindigo</code> <code>.color-focus-lightindigo</code> <code>.col-lightindigo</code>



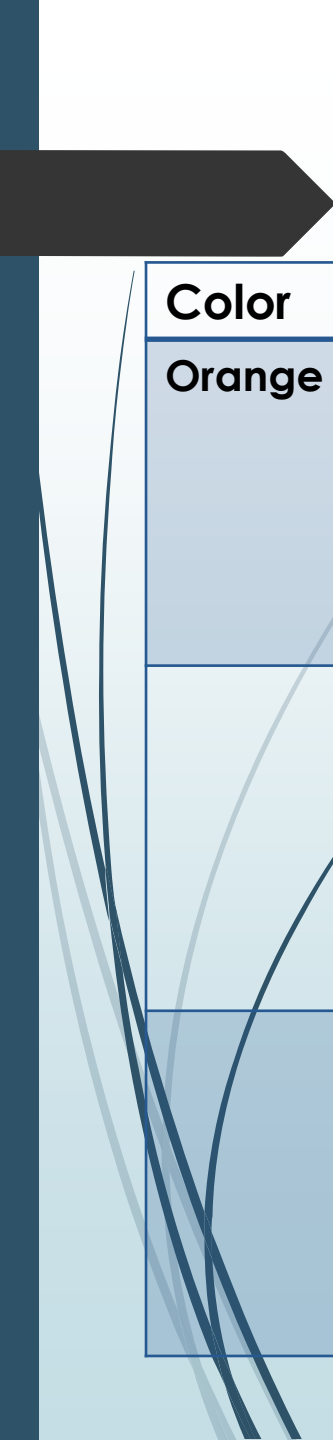
Color	Shade	Class/Id
Khaki	Khaki	.color-khaki , .color-hover-khaki .text-color-khaki , .text-color-hover-khaki .border-color-khaki , .border-color-hover-khaki .caretcolor-khaki .color-focus-khaki .col-khaki
	Dark khaki	.color-dark-khaki , .color-hover-dark-khaki .text-color-dark-khaki , .text-color-hover-dark-khaki .border-color-dark-khaki , .border-color-hover-dark-khaki .caretcolor-dark-khaki .color-focus-dark-khaki .col-dark-khaki
	Light khaki	.color-lightkhaki , .color-hover-lightkhaki .text-color-lightkhaki , .text-color-hover-lightkhaki .border-color-light-khaki , .border-color-hover-lightkhaki .caretcolor-lightkhaki .color-focus-lightkhaki .col-lightkhaki



Color	Shade	Class/Id
Lime	Lime	<code>.color-lime, .color-hover-lime</code> <code>.text-color-lime, .text-color-hover-lime</code> <code>.border-color-lime, .border-color-hover-lime</code> <code>.caretcolor-lime</code> <code>.color-focus-lime</code> <code>.col-lime</code>
	Dark lime	<code>.color-darklime, .color-hover-darklime</code> <code>.text-color-darklime, .text-color-hover-darklime</code> <code>.border-color-darklime, .border-color-hover-darklime</code> <code>.caretcolor-darklime</code> <code>.color-focus-darklime</code> <code>.col-darklime</code>
	Light lime	<code>.color-lightlime, .color-hover-lightlime</code> <code>.text-color-lightlime, .text-color-hover-lightlime</code> <code>.border-color-lightlime, .border-color-hover-lightlime</code> <code>.caretcolor-lightlime</code> <code>.color-focus-lightlime</code> <code>.col-lightlime</code>

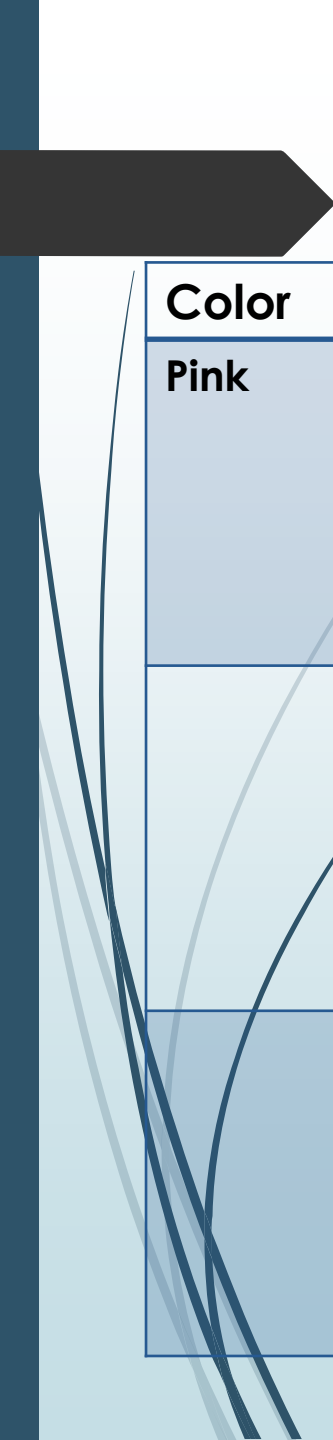


Color	Shade	Class/Id
Orange	Orange	<code>.color-orange, .color-hover-orange</code> <code>.text-color-orange, .text-color-hover-orange</code> <code>.border-color-orange, .border-color-hover-orange</code> <code>.caretcolor-orange</code> <code>.color-focus-orange</code> <code>.col-orange</code>
	Coral	<code>.color-coral, .color-hover-coral</code> <code>.text-color-coral, .text-color-hover-coral</code> <code>.border-color-coral, .border-color-hover-coral</code> <code>.caretcolor-coral</code> <code>.color-focus-coral</code> <code>.col-coral</code>
	Dark Orange	<code>.color-darkorange, .color-hover-darkorange</code> <code>.text-color-darkorange, .text-color-hover-darkorange</code> <code>.border-color-darkorange, .border-color-hover-darkorange</code> <code>.caretcolor-darkorange</code> <code>.color-focus-orange</code> <code>.col-darkorange</code>

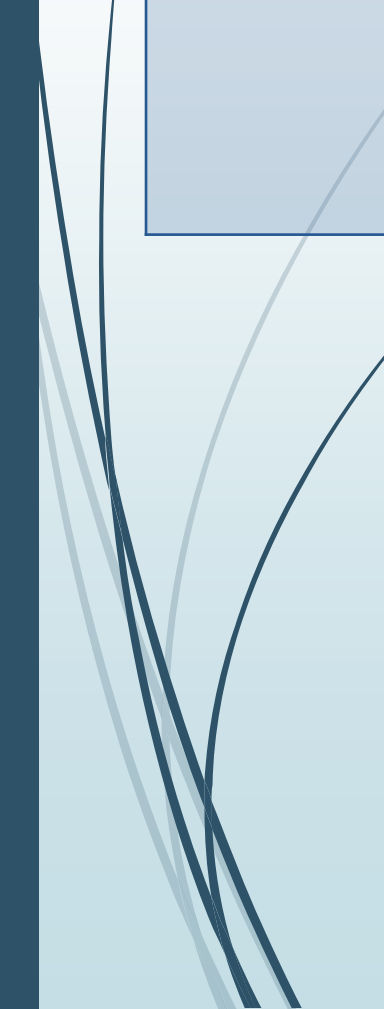



Color	Shade	Class/Id
Orange	Light orange	<code>.color-lightorange, .color-hover-lightorange</code> <code>.text-color-lightorange, .text-color-hover-lightorange</code> <code>.border-color-lightorange, .border-color-hover-lightorange</code> <code>.caretcolor-lightorange</code> <code>.color-focus-lightorange</code> <code>.col-lightorange</code>
	Orange-red	<code>.color-orangered, .color-hover-orangered</code> <code>.text-color-orangered, .text-color-hover-orangered</code> <code>.border-color-orangered, .border-color-hover-orangered</code> <code>.caretcolor-orangered</code> <code>.color-focus-orangered</code> <code>.col-orangered</code>
	Tomato	<code>.color-tomato, .color-hover-tomato</code> <code>.text-color-tomato, .text-color-hover-tomato</code> <code>.border-color-tomato, .border-color-hover-tomato</code> <code>.caretcolor-tomato</code> <code>.color-focus-tomato</code> <code>.col-tomato</code>

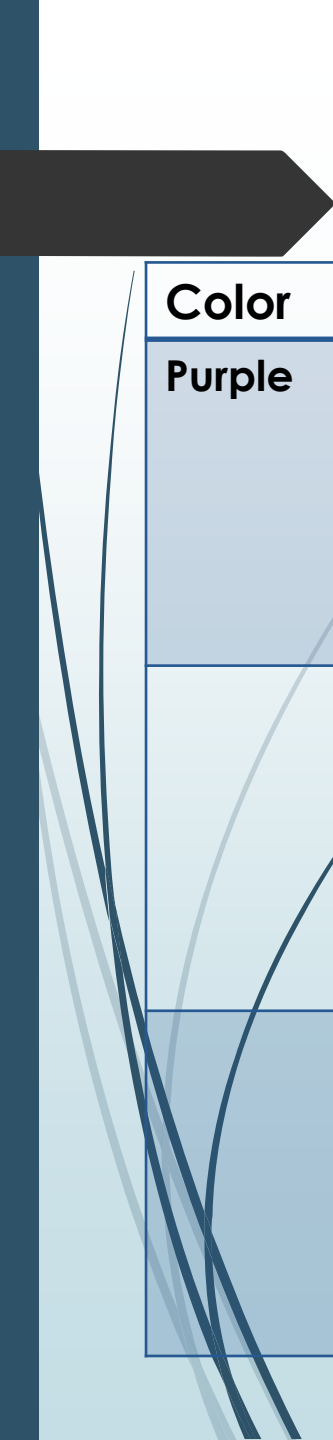




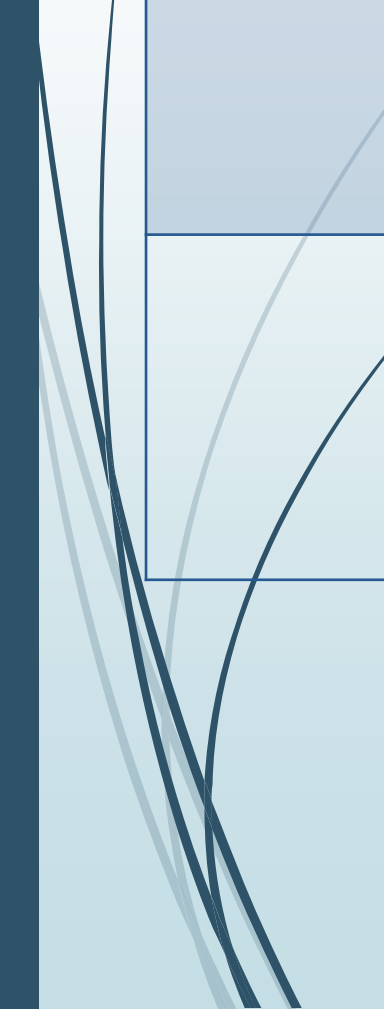

Color	Shade	Class/Id
Pink	Pink	<code>.color-pink, .color-hover-pink</code> <code>.text-color-pink, .text-color-hover-pink</code> <code>.border-color-pink, .border-color-hover-pink</code> <code>.caretcolor-pink</code> <code>.color-focus-pink</code> <code>.col-pink</code>
	Dark pink	<code>.color-darkpink, .color-hover-darkpink</code> <code>.text-color-darkpink, .text-color-hover-darkpink</code> <code>.border-color-darkpink, .border-color-hover-darkpink</code> <code>.caretcolor-darkpink</code> <code>.color-focus-darkpink</code> <code>.col-darkpink</code>
	Hot pink	<code>.color-hotpink, .color-hover-hotpink</code> <code>.text-color-hotpink, .text-color-hover-hotpink</code> <code>.border-color-hotpink, .border-color-hover-hotpink</code> <code>.caretcolor-hotpink</code> <code>.color-focus-hotpink</code> <code>.col-hotpink</code>



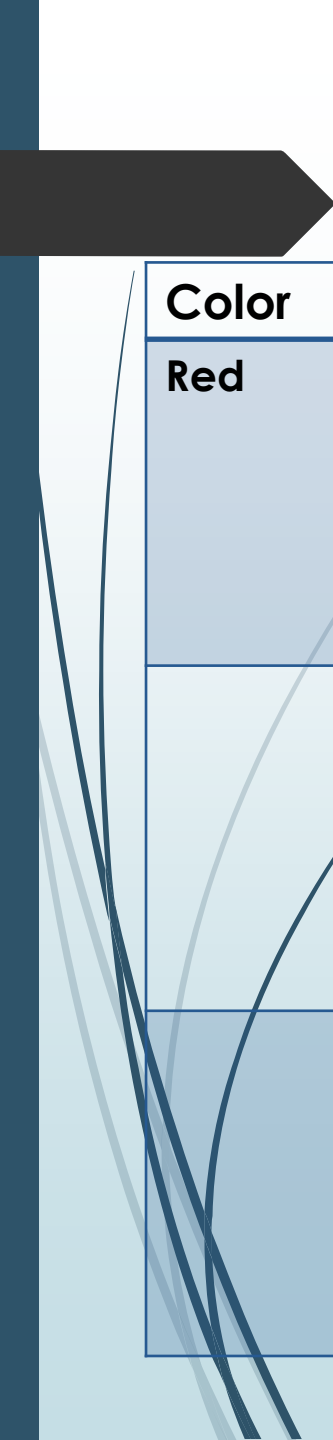
Color	Shade	Class/Id
Pink	Light pink	<code>.color-lightpink, .color-hover-lightpink</code> <code>.text-color-lightpink, .text-color-hover-lightpink</code> <code>.border-color-lightpink, .border-color-hover-lightpink</code> <code>.caretcolor-lightpink</code> <code>.color-focus-lightpink</code> <code>.col-lightpink</code>



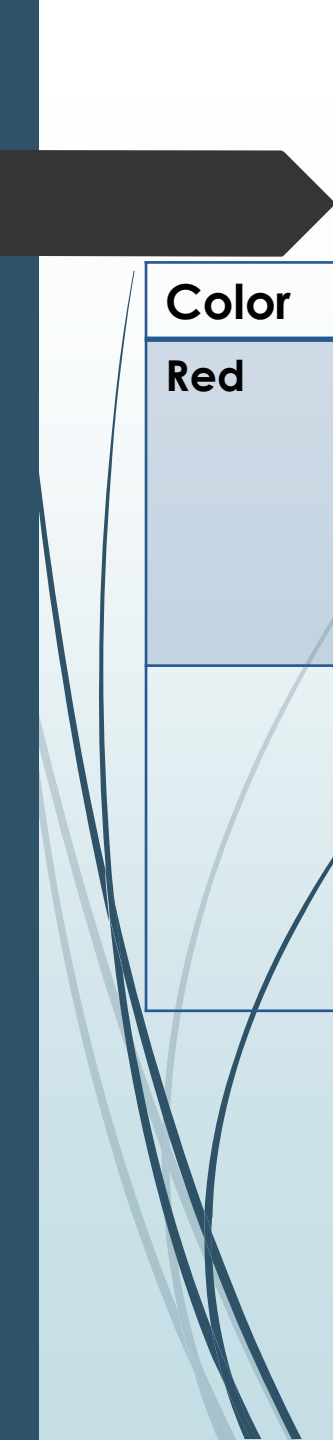
Color	Shade	Class/Id
Purple	Purple	<code>.color-purple, .color-hover-purple</code> <code>.text-color-purple, .text-color-hover-purple</code> <code>.border-color-purple, .border-color-hover-purple</code> <code>.caretcolor-purple</code> <code>.color-focus-purple</code> <code>.col-purple</code>
	Dark purple	<code>.color-darkpurple, .color-hover-darkpurple</code> <code>.text-color-darkpurple, .text-color-hover-darkpurple</code> <code>.border-color-darkpurple, .border-color-hover-darkpurple</code> <code>.caretcolor-darkpurple</code> <code>.color-focus-darkpurple</code> <code>.col-darkpurple</code>
	Lavendar	<code>.color-lavender, .color-hover-lavender</code> <code>.text-color-lavender, .text-color-hover-lavender</code> <code>.border-color-lavender, .border-color-hover-lavender</code> <code>.caretcolor-lavender</code> <code>.color-focus-lavender</code> <code>.col-lavender</code>



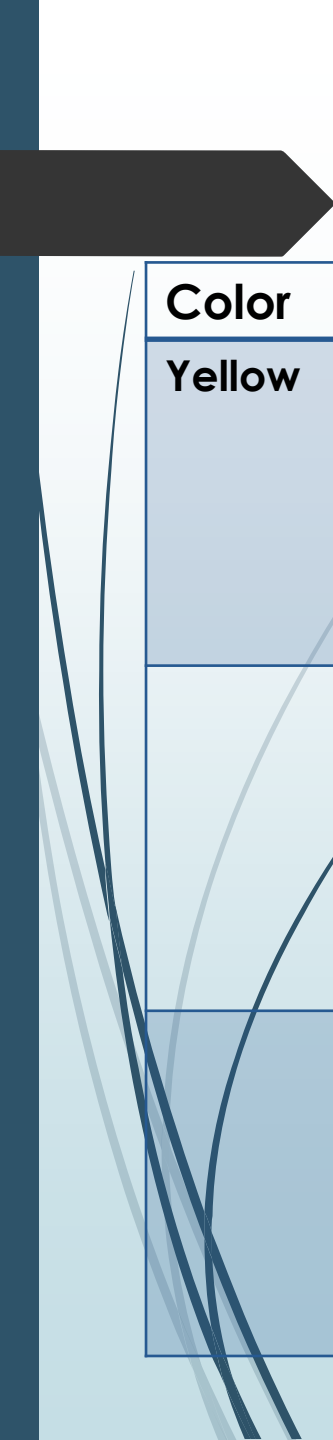
Color	Shade	Class/Id
Purple	Orchid	<code>.color-orchid, .color-hover-orchid</code> <code>.text-color-orchid, .text-color-hover-orchid</code> <code>.border-color-orchid, .border-color-hover-orchid</code> <code>.caretcolor-orchid</code> <code>.color-focus-orchid</code> <code>.col-orchid</code>
	Violet	<code>.color-violet, .color-hover-violet</code> <code>.text-color-violet, .text-color-hover-violet</code> <code>.border-color-violet, .border-color-hover-violet</code> <code>.caretcolor-violet</code> <code>.color-focus-violet</code> <code>.col-violet</code>



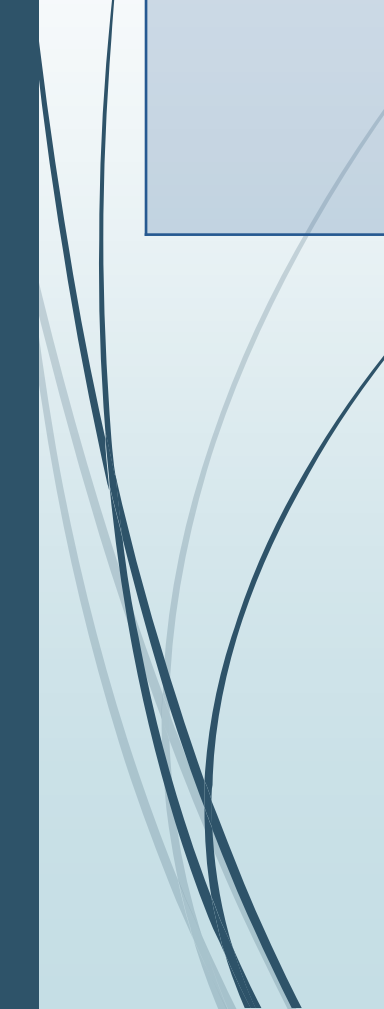

Color	Shade	Class/Id
Red	Red	<code>.color-red, .color-hover-red</code> <code>.text-color-red, .text-color-hover-red</code> <code>.border-color-red, .border-color-hover-red</code> <code>.caretcolor-red</code> <code>.color-focus-red</code> <code>.col-red</code>
	Crimson	<code>.color-crimson, .color-hover-crimson</code> <code>.text-color-crimson, .text-color-hover-crimson</code> <code>.border-color-crimson, .border-color-hover-crimson</code> <code>.caretcolor-crimson</code> <code>.color-focus-crimson</code> <code>.col-crimson</code>
	Indian red	<code>.color-indianred, .color-hover-indianred</code> <code>.text-color-indianred, .text-color-hover-indianred</code> <code>.border-color-indianred, .border-color-hover-indianred</code> <code>.caretcolor-indianred</code> <code>.color-focus-indianred</code> <code>.col-indianred</code>



Color	Shade	Class/Id
Red	Light red	<code>.color-lightred, .color-hover-lightred</code> <code>.text-color-lightred, .text-color-hover-lightred</code> <code>.border-color-lightred, .border-color-hover-lightred</code> <code>.caretcolor-lightred</code> <code>.color-focus-lightred</code> <code>.col-lightred</code>
	Pale red	<code>.color-palered, .color-hover-palered</code> <code>.text-color-palered, .text-color-hover-palered</code> <code>.border-color-palered, .border-color-hover-palered</code> <code>.caretcolor-palered</code> <code>.color-focus-palered</code> <code>.col-palered</code>

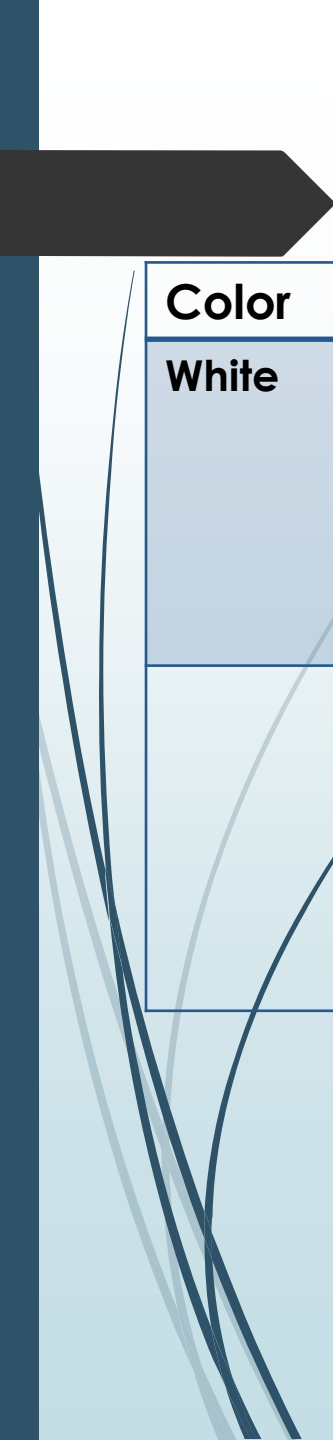


Color	Shade	Class/Id
Yellow	Yellow	<code>.color-yellow, .color-hover-yellow</code> <code>.text-color-yellow, .text-color-hover-yellow</code> <code>.border-color-yellow, .border-color-hover-yellow</code> <code>.caretcolor-palered</code> <code>.color-focus-yellow</code> <code>.col-yellow</code>
	Gold	<code>.color-gold, .color-hover-gold</code> <code>.text-color-gold, .text-color-hover-gold</code> <code>.border-color-gold, .border-color-hover-gold</code> <code>.caretcolor-gold</code> <code>.color-focus-gold</code> <code>.col-gold</code>
	Moccasin	<code>.color-moccasin, .color-hover-moccasin</code> <code>.text-color-moccasin, .text-color-hover-moccasin</code> <code>.border-color-moccasin, .border-color-hover-moccasin</code> <code>.caretcolor-moccasin</code> <code>.color-focus-moccasin</code> <code>.col-moccasin</code>

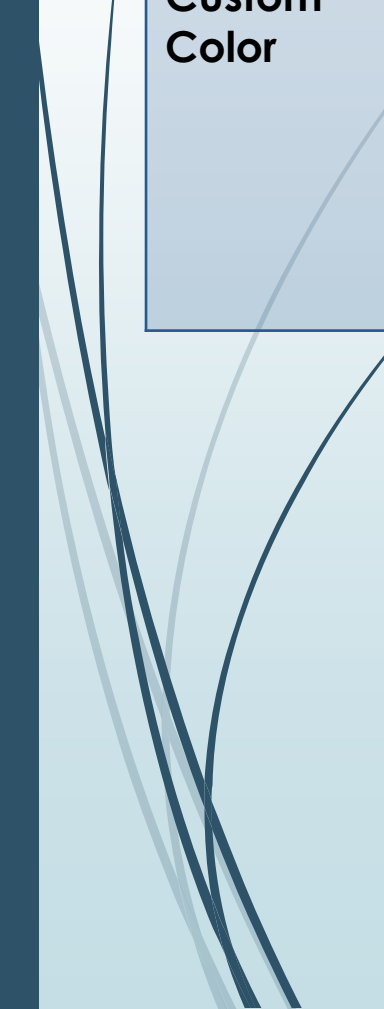



Color	Shade	Class/Id
Yellow	Pale yellow	<code>.color-paleyellow, .color-hover-paleyellow</code> <code>.text-color-paleyellow, .text-color-hover-paleyellow</code> <code>.border-color-paleyellow, .border-color-hover-paleyellow</code> <code>.caretcolor-paleyellow</code> <code>.color-focus-paleyellow</code> <code>.col-paleyellow</code>





Color	Shade	Class/Id
White	White	<code>.color-white, .color-hover-white</code> <code>.text-color-white, .text-color-hover-white</code> <code>.border-color-white, .border-color-hover-white</code> <code>.caretcolor-white</code> <code>.color-focus-white</code> <code>.col-white</code>
	Snow white	<code>.color-snow-white, .color-hover-snow-white</code> <code>.text-color-snow-white, .text-color-hover-snow-white</code> <code>.border-color-snow-white, .border-color-hover-snow-white</code> <code>.caretcolor-snow-white</code> <code>.color-focus-snow-white</code> <code>.col-snow-white</code>



Color	Shade	Class/Id
<b>Your Custom Color</b>	Replace /*comment(s)*/ with any color name/code or none before calling/using classes	<code>.color-myPie, .color-hover-myPie</code> <code>.text-color-myPie, .text-color-hover-myPie</code> <code>.border-color-myPie, .border-color-hover-myPie</code> <code>.caretcolor-myPie</code> <code>.color-focus-myPie</code> <code>.col-myPie</code>



# WORKING WITH CURSORS

Understanding the PieCSS Cursor Presentation Modes

## CURSORS

With PieCSS, it is all about presentation and styling your presentations and cursors being one of the considerations of the pie.css framework, it is included in the presentation systems developed by Pie.

### Define cursors?

Cursors are graphical aids in selecting objects and commands on an output device. Most cursors are controlled by a mouse but can also be controlled by other input devices such as the Joystick and several pointer devices.

Please note the following when working with cursors in Pie-CSS

- A variety of cursor pointers are supported in Pie-CSS
- It is easier to switch to any of the various cursor styles simply by using any of the CSS classes
- Cursor classes are meant to be used on containers and elements though they can also be applied to whole webpage by styling the <body> tag with a relevant cursor class
- There is **no** room to add your own custom cursor style in the PieCSS presentation system
- You can always reset cursors to default style anytime

## CURSOR PRESENTATION SYSTEM

### Manipulating cursors

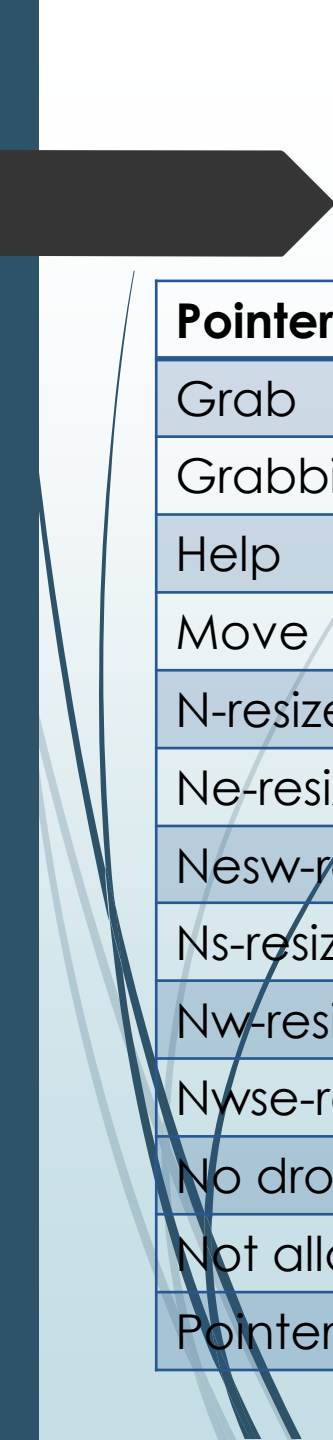
The table shows the different ways of manipulating cursor presentation in PieCSS

Action	Class/Id	Description
Overwrite cursor mode	<code>.pie-overwrite .pie-cursor</code>	Overwrite cursor where used previously
Pointer events	<code>.pie-cursor</code>	Disable pointer events
Adds visibility	<code>.pie-focused section.pie-cursors</code>	Visibility visible – works for other elements as well
Overlap	<code>.pie-big-cursor .pie-cursors</code>	Ensures cursor is placed overlaying the element(s) in focus
Disable cursor	<code>.pie-disabled</code>	Disable mouse(and cursor)
Disable cursor II	<code>.pie-disabled *</code>	Disables all pointer events
Hides cursor	<code>.cursor-none</code>	Hides cursor on a class
Animation	<code>.pie-animate-fat-cursor</code>	Animates cursor
Reset pointer	<code>. cursor-default</code>	Reset mouse pointer shape to default
Hide on print	<code>pie section.hide-cursors</code>	Hides cursor when printing. Works only on @print event

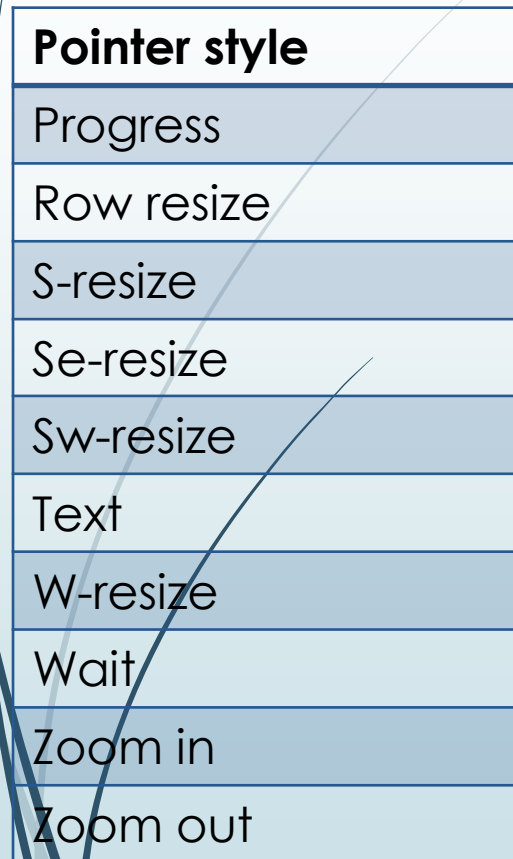

## Different cursor pointer styles

Different cursor pointer styles to use within an HTML element or even for your entire `<body>` element

Pointer style	Class/Id
Alias	<code>.cursor-alias</code>
All scroll	<code>.cursor-all-scroll</code>
Auto	<code>.cursor-auto</code>
Cell	<code>.cursor-cell</code>
Context menu	<code>.cursor-context-menu</code>
Col-resize	<code>.cursor-col-resize</code>
Copy	<code>.cursor-copy</code>
Crosshair	<code>.cursor-crosshair</code>
E-resize	<code>.cursor-e-resize</code>
Ew-resize	<code>.cursor-ew-resize</code>



Pointer style	Class/Id
Grab	.cursor-grab
Grabbing	.cursor-grabbing
Help	.cursor-help
Move	.cursor-move
N-resize	.cursor-n-resize
Ne-resize	.cursor-ne-resize
Nesw-resize	.cursor-nesw-resize
Ns-resize	.cursor-ns-resize
Nw-resize	.cursor-nw-resize
Nwse-resize	.cursor-nwse-resize
No drop	.cursor-no-drop
Not allowed	.cursor-not-allowed
Pointer	.cursor-pointer



Pointer style	Class/Id
Progress	.cursor-progress
Row resize	.cursor-row-resize
S-resize	.cursor-s-resize
Se-resize	.cursor-se-resize
Sw-resize	.cursor-sw-resize
Text	.cursor-text
W-resize	.cursor-w-resize
Wait	.cursor-wait
Zoom in	.cursor-zoom-in
Zoom out	.cursor-zoom-out





## PRELOADERS

Preloaders are what you see on the screen while the rest of the websites' page content is still loading.

Preloaders have been included in the PieCSS framework from a third-party work by Tobia Sahlin.

See more about Tobia Sahlin => <https://tobiasahlin.com/>

Github: <https://github.com/tobiasahlin/>

Please note that jQuery is required to animate the preloaders and this can be found in our piejs folder as preloader.js



Pie is currently maintained by Ojaay and welcomes collaborators and users.

Join pie on Github:

<https://github.com/eugyenocho/pie>