## CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

* Simple selectors (select elements based on name, id, class)
* [Combinator selectors](https://www.w3schools.com/css/css_combinators.asp) (select elements based on a specific relationship between them)
* [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
* [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
* [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)

### All CSS Simple Selectors

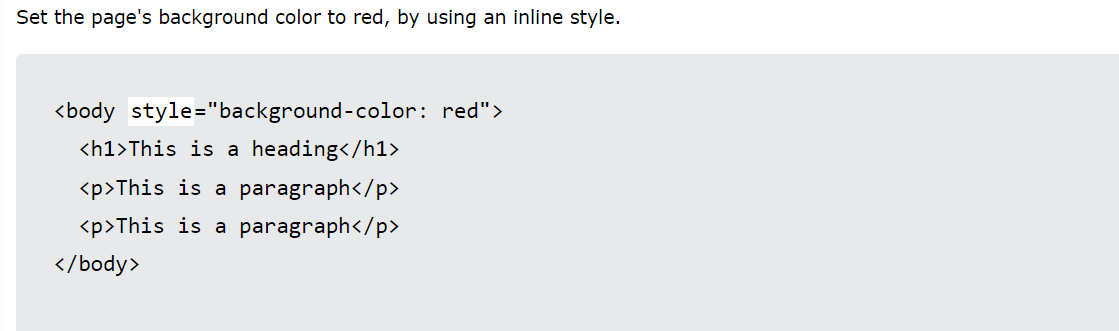
|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [#*id*](https://www.w3schools.com/cssref/sel_id.asp) | #firstname | Selects the element with id="firstname" |
| [.*class*](https://www.w3schools.com/cssref/sel_class.asp) | .intro | Selects all elements with class="intro" |
| [*element.class*](https://www.w3schools.com/cssref/sel_element_class.asp) | p.intro | Selects only <p> elements with class="intro" |
| [\*](https://www.w3schools.com/cssref/sel_all.asp) | \* | Selects all elements |
| [*element*](https://www.w3schools.com/cssref/sel_element.asp) | p | Selects all <p> elements |
| [*element,element,..*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p , h1 , h3, img | Selects all <div> elements and all <p> elements and so on together |

Note : Class is targerted by .classname

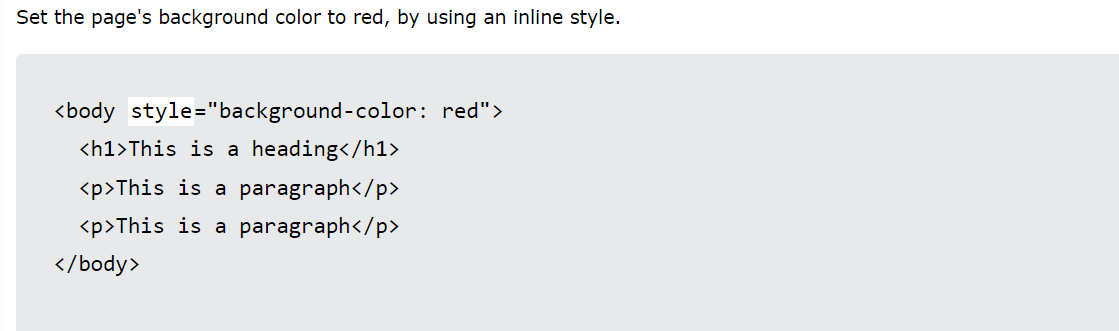
Id is targeted by #id

## Adding external css file

<link rel=”Sytlesheet” href=”mystyle.css”>



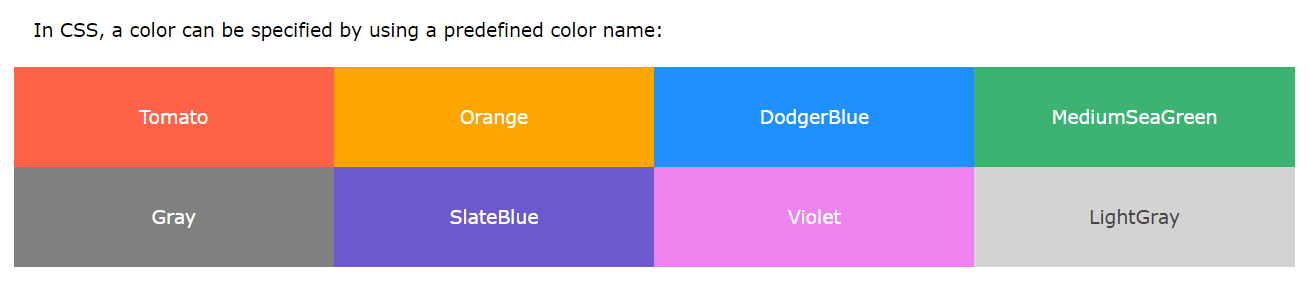
CSS CAN BE APPLIED DIRECTLY INSIDE THE TAG OF THE ELEMENT AS FOLLOWS



### Comments

/\* \*/ are used for commenting in css style sheets.

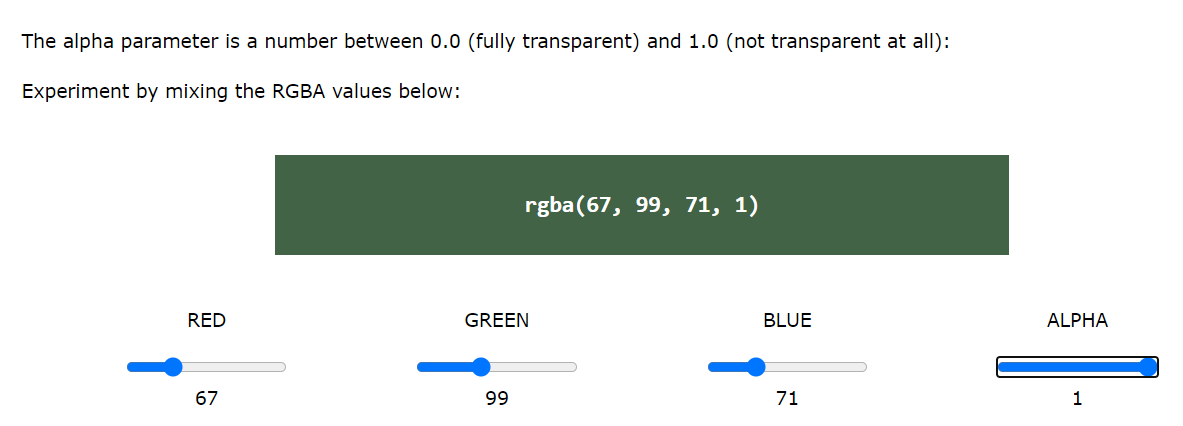
### Color names:



CSS has more than 150 colors’ support.

Color can be given to borders as well.

Alpha defines the intensity of RGBA COLOR in css Style Sheets.



HSL(hue , %saturation , %lightness) another type of colors.

Hex is type of colors that used hexadecimal values for respective colors.

# Backgrounds

## BG COLOR

Div{

Background:navy ;

Opacity:0.3; //opacity is from 0.1 to 1 for 10 to 100 % respectively

}

## BG IMG

image repeats itself over and over again to cover the whole page ;

div{

bacgroung-img:url( here has to be the address of the image).

}

Since image repats itself we can choose which dimension of it has to be repetative

**body {**

**background-image: url("gradient\_bg.png");**

**background-repeat: repeat-x;**

## BG attachment

It tells where the bg img should move with the page or remain sticky like nav bar

|  |  |
| --- | --- |
| Background | |
| body {  background-image: url("gradient\_bg.png");  background-repeat: repeat-x;  } | For setting the image as bg and repeating the image along x, y , z |
| body {   background-image: url("img\_tree.png");   background-repeat: no-repeat;   background-position: right top;   background-attachment: scroll; //if fixed the img will be fixed and other contents of body will be scrollable only } | For attachement of img as scroll or fixed relative to web page . |
|  | No-repeat for not repeating the image.  Margin-top is from 100px top the content of body will start displayin |
| This is shorthand for above properties  body {  background: skyblue url("img\_tree.png") no-repeat right top;  margin-right: 200px;  } |  |
|  | |
| Border styles | |
| p.dotted {border-style: dotted;} p.dashed {border-style: dashed;} p.solid {border-style: solid;} p.double {border-style: double;} p.groove {border-style: groove;} p.ridge {border-style: ridge;} p.inset {border-style: inset;} p.outset {border-style: outset;} p.none {border-style: none;} p.hidden {border-style: hidden;} p.mix {border-style: dotted dashed solid double;} | |
| These are the different border styles for above properties | |
| border-style: dashed;  border-width:10px 10px 10px 10px; /\* 25px top, 10px right, 4px bottom and 35px left \*/ |  |
| border-color: blue; |  |
| border-color: red green blue yellow; | red top, green right, blue bottom and yellow left  clock wise(same can be applied to border-width as well ) |
| border-top-style: dotted;   border-right-style: solid;   border-bottom-style: dotted;   border-left-style: solid; | This of for sepeartely styling each border |
| Short  border: 5px solid red;  here border style(here solid ) is always required | The border property is a shorthand property for the following individual border properties:   * border-width * border-style (required) * border-color |
| border: 2px solid red;   border-radius: 5px; | For rounded borders |
|  | |
| Margins : The CSS margin properties are used to create space around elements, outside of any defined borders.  CSS has properties for specifying the margin for each side of an element:   * margin-top * margin-right * margin-bottom * margin-left   margin can be   * auto - the browser calculates the margin * *length* - specifies a margin in px, pt, cm, etc. * inherit - specifies that the margin should be inherited from the parent element | |
| here this size is the size of the div  margin: auto;  div has a border | width: 300px;   margin: auto;   border: 1px solid red; |
| **CSS Padding** Thee CSS padding properties are used to generate space around an element's content, inside of any defined borders.  With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left). | |
| **CSS Height, Width and Max-width** The height and width properties may have the following values:   * auto - This is default. The browser calculates the height and width * length - Defines the height/width in px, cm, etc. * % - Defines the height/width in percent of the containing block * initial - Sets the height/width to its default value * inherit - The height/width will be inherited from its parent value | |
| CSS LAYOUR FOR PADDING, BORDER , MARGIN , BORDER | |
|  |  |
| CSS Outline An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".  1. Demonstration of the different ***outline styles***:  p.dotted {outline-style: dotted;} p.dashed {outline-style: dashed;} p.solid {outline-style: solid;} p.double {outline-style: double;} p.groove {outline-style: groove;} p.ridge {outline-style: ridge;} p.inset {outline-style: inset;} p.outset {outline-style: outset;}  outline is outside the border.       |  |  | | --- | --- | | [outline-offset](https://www.w3schools.com/cssref/css3_pr_outline-offset.asp) | Specifies the space between an outline and the edge or border  of an element | | |
|  | |
|  |  |

|  |
| --- |
|  |
| **CSS Text Alignment** |
|  |

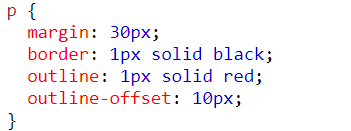
# **Text Styles**

The text-align property is used to set the horizontal alignment of a text.

There are three types of alignments

* 1. left
  2. right
  3. justify (makes sure words are equally distributed to make a line)

The direction and unicode-bidi properties can be used to change the text direction of an element:



## The vertical-align

 property sets the vertical alignment of an element

The text-align-last property specifies how to align the last line of a text.

## text-decoration-line

 property is used to add a decoration line to text.

  text-decoration-line: overline;  
  text-decoration-line: line-through;  
  text-decoration-line: underline;

text-decoration-line: overline underline;

The text-decoration-color property is used to set the color of the decoration line.   text-decoration-thickness: 5px;

Text-decoration-color:Green;

text-decoration-line: overline underline;  
  text-decoration-color: purple;



The Shorthand Property

The text-decoration property is a shorthand property for:

* text-decoration-line (required)
* text-decoration-color (optional)
* text-decoration-style (optional)
* text-decoration-thickness (optional)

p {  
  text-decoration: underline red double 5px;

}

|  |  |  |
| --- | --- | --- |
| **Property** | | **Description** |
| [text-decoration](https://www.w3schools.com/cssref/pr_text_text-decoration.asp) | | Sets all the text-decoration properties in one declaration |
| [text-decoration-color](https://www.w3schools.com/cssref/css3_pr_text-decoration-color.asp) | Specifies the color of the text-decoration | |
| [text-decoration-line](https://www.w3schools.com/cssref/css3_pr_text-decoration-line.asp) | | Specifies the kind of text decoration to be used (underline, overline, etc.) |
| [text-decoration-style](https://www.w3schools.com/cssref/css3_pr_text-decoration-style.asp) | | Specifies the style of the text decoration (solid, dotted, etc.) |
| [text-decoration-thickness](https://www.w3schools.com/cssref/pr_text_text-decoration-thickness.asp) | | Specifies the thickness of the text decoration line |

## Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

### **Example**

  text-transform: uppercase;  
  text-transform: lowercase;

  text-transform: capitalize;

# Text Shadow

The text-shadow property adds shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

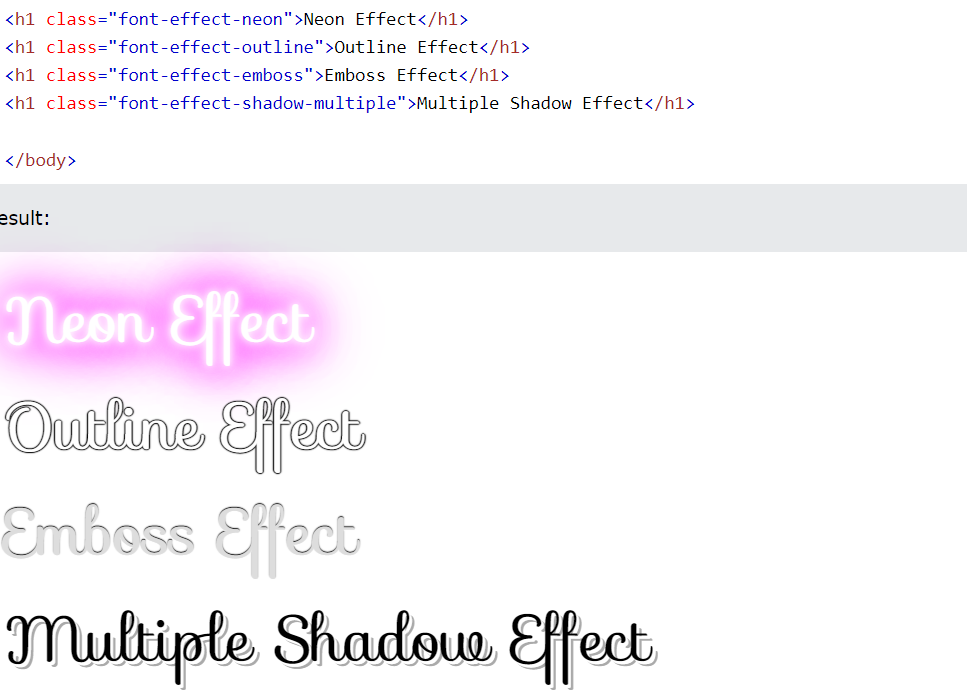
  text-shadow: 2px 2px;

Then, add a blur effect (5px) to the shadow:

## Text shadow effect

Below 5px adds blur effect to shadow   
  text-shadow: 2px 2px 5px red;

# Font effects

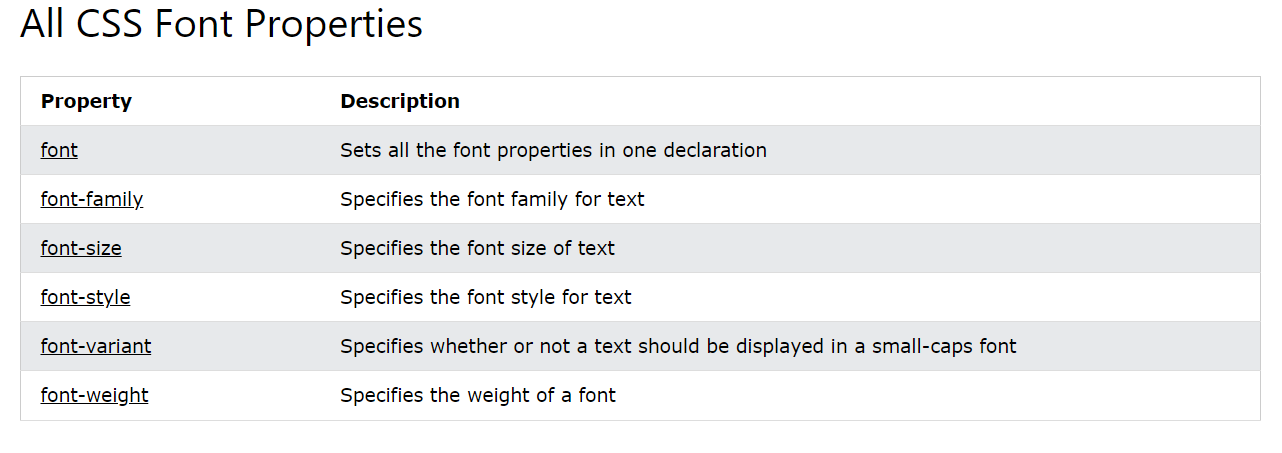


Use font to set several font properties in one declaration:

p.a {  
  font: 20px Arial, sans-serif;

}

# ALL CSS FONT PROPERTIES



# Adding icons

## Font Awesome Icons

To use the Font Awesome icons, go to [fontawesome.com](https://fontawesome.com/), sign in, and get a code to add in the <head> section of your HTML page:

<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>

Read more about how to get started with Font Awesome in our [Font Awesome 5 tutorial](https://www.w3schools.com/icons/fontawesome5_intro.asp).

**Note:** No downloading or installation is required

# Lists

|  |  |
| --- | --- |
| **Property** | **Description** |
| [list-style](https://www.w3schools.com/cssref/pr_list-style.asp) | Sets all the properties for a list in one declaration |
| [list-style-image](https://www.w3schools.com/cssref/pr_list-style-image.asp) | Specifies an image as the list-item marker |
| [list-style-position](https://www.w3schools.com/cssref/pr_list-style-position.asp) | Specifies the position of the list-item markers (bullet points) |
| [list-style-type](https://www.w3schools.com/cssref/pr_list-style-type.asp) | Specifies the type of list-item marker |

## Different List Item Markers

## The list-style-type

The following example shows some of the available list item markers:ul.a {  
  list-style-type: circle;  
  list-style-type: square;  
  list-style-type: upper-roman;  
  list-style-type: lower-alpha;  
}

## Remove Default Settings

The list-style-type:none property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add margin:0 and padding:0 to <ul> or <ol>:

* list-style-type (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
* list-style-position (specifies whether the list-item markers should appear inside or outside the content flow)
* list-style-image (specifies an image as the list item marker)

# RESPONSIVE TABLE

CSS Table Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [border](https://www.w3schools.com/cssref/pr_border.asp) | Sets all the border properties in one declaration |
| [border-collapse](https://www.w3schools.com/cssref/pr_border-collapse.asp) | Specifies whether or not table borders should be collapsed |
| [border-spacing](https://www.w3schools.com/cssref/pr_border-spacing.asp) | Specifies the distance between the borders of adjacent cells |
| [caption-side](https://www.w3schools.com/cssref/pr_tab_caption-side.asp) | Specifies the placement of a table caption |
| [empty-cells](https://www.w3schools.com/cssref/pr_tab_empty-cells.asp) | Specifies whether or not to display borders and background on empty  cells in a table |
| [table-layout](https://www.w3schools.com/cssref/pr_tab_table-layout.asp) | Sets the layout algorithm to be used for a table |

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content:

Add a container element (like <div>) with overflow-x:auto around the <table> element to make it responsive:

<div style="overflow-x:auto;">  
  
<table>  
... table content ...  
</table>  
  
</div>

# The display Property

The display property is used to specify how an element is shown on a web page.

Block-level Elements

A block-level element ALWAYS starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

Examples of block-level elements:

* <div>
* <h1> - <h6>
* <p>
* <form>
* <header>
* <footer>
* <section>

Display:block ;

Inline Elements

An inline element DOES NOT start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements:

* <span>
* <a>
* <img>
* Display:inline ;

The display Property Values

The display property has many values:

|  |  |
| --- | --- |
| **Value** | **Description** |
| inline | Displays an element as an inline element |
| block | Displays an element as a block element |
| contents | Makes the container disappear, making the child elements children of the  element the next level up in the DOM |
| flex | Displays an element as a block-level flex container |
| grid | Displays an element as a block-level grid container |
| inline-block | Displays an element as an inline-level block container. The element itself is  formatted as an inline element, but you can apply height and width values |
| inline-flex | Displays an element as an inline-level flex container |
| inline-grid | Displays an element as an inline-level grid container |
| inline-table | The element is displayed as an inline-level table |
| list-item | Let the element behave like a <li> element |
| run-in | Displays an element as either block or inline, depending on context |
| table | Let the element behave like a <table> element |
| table-caption | Let the element behave like a <caption> element |
| table-column-group | Let the element behave like a <colgroup> element |
| table-header-group | Let the element behave like a <thead> element |
| table-footer-group | Let the element behave like a <tfoot> element |
| table-row-group | Let the element behave like a <tbody> element |
| table-cell | Let the element behave like a <td> element |
| table-column | Let the element behave like a <col> element |
| table-row | Let the element behave like a <tr> element |
| none | The element is completely removed |
| **Remove an Element –**  **display:none**    **Element is remove as if it did’nt exist** | **Hide an Element**  **visibility:hidden**    **Element occupies the space but it is hidden in the page.** |
| initial | Sets this property to its default value |
| inherit | Inherits this property from its parent element |

CSS Display/Visibility Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [display](https://www.w3schools.com/cssref/pr_class_display.php) | Specifies how an element should be displayed |
| [visibility](https://www.w3schools.com/cssref/pr_class_visibility.php) | Specifies whether or not an element should be visible |

## CSS Layout - width and max-width

Max-width is better when web-application is being made for small devices such as mobile phones.It is

MORE REPONSIVE.

div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}

# The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

## Static

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This <div> element has position: static;



## Relative

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}

## Fixed

div.fixed {

position: fixed;

bottom: 20px; //this would be the gap from bottom to div.

right: 30px; //this would be the gap from right side .

width: 300px;

border: 3px solid #73AD21;

}

## Absolute

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Example

div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}  
  
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}

## Sticky :

An element with position: sticky; is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

div.sticky {  
  position: -webkit-sticky; /\* Safari \*/  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}

# The z-index Property

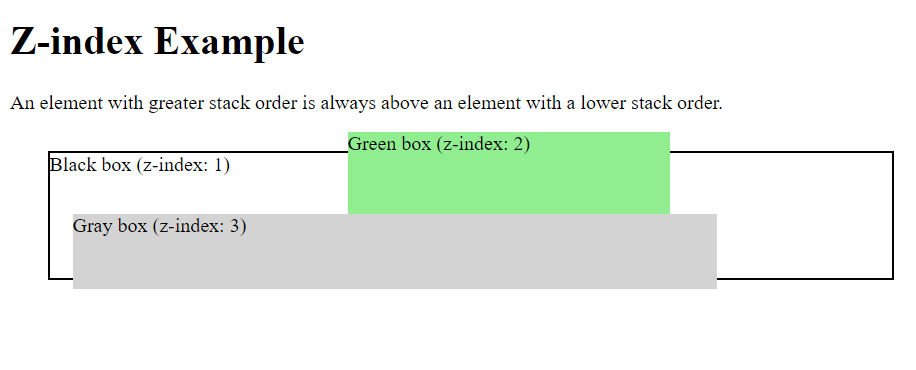
The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order:

1. If image has a z-index of -1, it will be placed behind the text.

img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}

## Example :

.black-box {  
  position: relative;  
  z-index: 1; //here this box will be behind all other boxes.  
  border: 2px solid black;  
  height: 100px;  
  margin: 30px;  
}  
.gray-box {  
  position: absolute;  
  z-index: 3; //this box will be on the top of all boxes.  
  background: lightgray;  
  height: 60px;  
  width: 70%;  
  left: 50px;  
  top: 50px;  
}  
.green-box {  
  position: absolute;  
  z-index: 2; //this box will be between the other two boxes.  
  background: lightgreen;  
  width: 35%;  
  left: 270px;  
  top: -15px;  
  height: 100px;  
} 

# OverFlow and Scroll bar

It decides how does the div behaves wrt to change in size (when increased / decreased) .

1. overflow: auto

The auto value is similar to scroll, but it adds scrollbars only when necessary:

## overflow-x and overflow-y

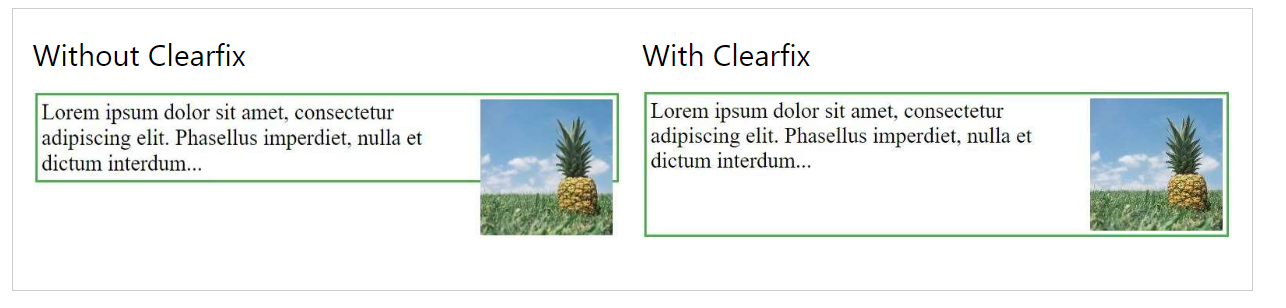
div {  
  overflow-x: hidden; /\* Hide horizontal scrollbar \*/  
  overflow-y: scroll; /\* Add vertical scrollbar \*/  
}

# FLOAT :

1. FLOAT: LEFT
2. FLOAT: RIGHT

## The clearfix Hack

If a floated element is taller than the containing element, it will "overflow" outside of its container. We can then add a clearfix hack to solve this problem:



# CSS - The :lang Pseudo-class

Pseudo-classes in CSS allows you to make the different classes react differently under different conditions.

## <q lang=”no” >

q:lang(no) {  
  quotes: "~" "~";  
}

## All CSS Pseudo Classes

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> elements that is the first child  of its parent |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p>  element of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the <input> element that has focus |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects <input> elements with a value within a  specified range |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all <input> elements with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute  value starting with "it" |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> elements that is the last child of its  parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element  of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links |
| [:not(selector)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element |
| [:nth-child(n)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of  its parent |
| [:nth-last-child(n)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of  its parent, counting from the last child |
| [:nth-last-of-type(n)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p>  element of its parent, counting from the last child |
| [:nth-of-type(n)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p>  element of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p>  element of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child  of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects <input> elements with  no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects <input> elements with a value outside a  specified range |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects <input> elements with a "readonly"  attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects <input> elements with no "readonly"  attribute |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects <input> elements with a "required"  attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | root | Selects the document's root element |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element  (clicked on a URL containing that anchor name) |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all <input> elements with a valid value |
| [:visited](https://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links |

## Pseudo class elements

## All CSS Pseudo Elements

|  |  |  |
| --- | --- | --- |
| Selector | Example | Example description |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert content afterevery <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert content beforeevery <p> element |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element |
| [::marker](https://www.w3schools.com/cssref/sel_marker.asp) | ::marker | Selects the markers of list items |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | p::selection | Selects the portion of an element that isselected by a user |

# Attribute Selectors

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [[*attribute*]](https://www.w3schools.com/cssref/sel_attribute.php) | [target] | Selects all elements with a target attribute |
| [[*attribute*=*value*]](https://www.w3schools.com/cssref/sel_attribute_value.php) | [target="\_blank"] | Selects all elements with target="\_blank" |
| [[*attribute*~=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_contains.php) | [title~="flower"] | Selects all elements with a title attribute that  contains a space-separated list of words, one  of which is "flower" |
| [[*attribute*|=*value*]](https://www.w3schools.com/cssref/sel_attribute_value_lang.php) | [lang|="en"] | Selects all elements with a lang attribute value  starting with "en" |
| [[*attribute*^=*value*]](https://www.w3schools.com/cssref/sel_attr_begin.php) | a[href^="https"] | Selects all <a> elements with a href attribute value  starting with "https" |
| [[*attribute*$=*value*]](https://www.w3schools.com/cssref/sel_attr_end.php) | a[href$=".pdf"] | Selects all <a> elements with a href attribute value  ending with ".pdf" |
| [[*attribute*\*=*value*]](https://www.w3schools.com/cssref/sel_attr_contain.php) | a[href\*="w3schools"] | Selects all <a> elements with a href attribute value  containing the substring "w3schools" |

## Use of attribute selectors :

In forms : e.g : to apply property to element with specific attribute value ;

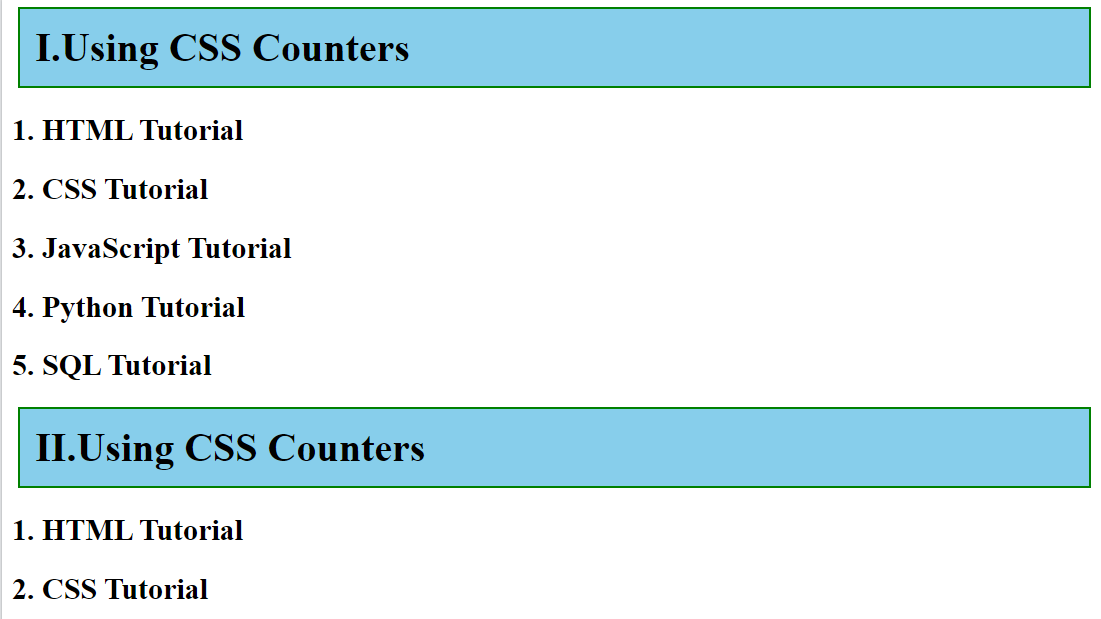
input[type=text] {  
  width: 100%;  
  padding: 12px 20px;  
  margin: 8px 0;  
  box-sizing: border-box;  
}



# **CSS Counters**

Counter are used to add counter using css to heading and sub headings :

ASK CHAT GPT TO WRITE CODE TO ADD COUNTERS TO HEADING (H1 , ETC,…) AND SUBHEADINGS(H2 , ETC ,….)



## CODE

body {  
  counter-reset: section;  
}  
  
h1 {  
  counter-reset: subsection;  
}  
  
h1::before {  
  counter-increment: section;  
  content: "Section " counter(section) ". ";  
}  
  
h2::before {  
  counter-increment: subsection;  
  content: counter(section) "." counter(subsection) " ";  
}

## CSS Counter Properties

|  |  |
| --- | --- |
| Property | Description |
| [content](https://www.w3schools.com/cssref/pr_gen_content.asp) | Used with the ::before and ::afterpseudo-elements, to insert generated content |
| [counter-increment](https://www.w3schools.com/cssref/pr_gen_counter-increment.asp) | Increments one or more counter values |
| [counter-reset](https://www.w3schools.com/cssref/pr_gen_counter-reset.asp) | Creates or resets one or more counters |
| [counter()](https://www.w3schools.com/cssref/func_counter.asp) | Returns the current value of the named counter |

# **Flexbox**

## [Flexbox Cheat sheet](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) 🡨 click here to see the cheat sheet.

## Example Present in Example at the end of Image Gallery

## Properties of flex boxes :

1. Flex direction: 1. Horizontal (ROW) , vertical ( COLUMNS )
2. Justify content (based on direction ) start, end , space-between , space-evenly

IT IS APPLIED INSIDE THE PARENT BUT DEFINES HOW INSIDE ITEMS WILL BE PLACED;

1. Align-items : for aligning items ***vertically***
2. Flex-start: top
3. Center IN THE MID OF THE BOX
4. Flex-end
5. Flex wrap : wrap

If no space is available our items will try to wrap to adjust inside the div.

It’s a property of parent means that it tells the parent container who should its children behave.

1. Align-content(different from align items ) aligns content if flex-wrap is set to wrap depending on property used with it.
2. Gap : used to add space between the elements of flex boxes.
3. Algin-self : used to target the individual items …has properties:
4. Flex basis :

Increased height if used with flex-direction:column and increases width if used with flex-direction:row ;

## FLEX-SIZING

It is used to define how the

1. Surpulus place inside div
2. Space deficity inside the div

Will be resolved and has three properties.

1. Flex grow:1;

IT MEANS THAT FREE SPACE BY THE FACTOR OF 1 WILL BE DISTRIBUTED EQUALLY AMONG ALL ITEMS.

IF FLEX-GROW:2 for certain item(say nth child:2) 2nd item will take 2-times the space of other items

1. Flex-shrink : used if there is less space than required to fix the items.
2. Default : 1
3. Flex-shrink : 0 overflow (don’t make items smaller )
4. It’s opposite of flex-grow;

Example :

.item2{

Flex-shrink:2; //it will set the item2 to 2X less than other (2) items.

}

1. Flex-basis : Correct method of using grow and shrink.

So while using the flex-box we should use flex-basis instead of width and height because it checks the direction of the flex i.e flex-direction and depending upon the factor value it automatically adjusts the items.

Shortcut : flex: 0 1 auto; for flex-grow , flex-shrink and flex-basis;

## Code for responsive flexbox

div.gallery {

border: 1px solid #ccc;

}

div.gallery:hover {

border: 1px solid #777;

}

div.gallery img {

width: 100%;

height: auto;

}

div.desc {

padding: 15px;

text-align: center;

}

\* {

box-sizing: border-box;

}

.responsive {

padding: 0 6px;

float: left;

width: 24.99999%;

}

@media only screen and (max-width: 700px) {

.responsive {

width: 49.99999%;

margin: 6px 0;

}

}

@media only screen and (max-width: 500px) {

.responsive {

width: 100%;

}

}

.clearfix:after {

content: "";

display: table;

clear: both;

}

# Grids a

## Terms for grid layouts

## **Grid**

## **Lines**

## **Cells**

## **Tracks**

## **Container**

## **Items**

Making grids ;

Display : grid ;

Grid-tempelate-columns : 1fr 2fr ; /\*allocates 1 part to one columna and 2 parts to two columns\*/

Grid-tempelate-row : 1fr 3fr ; /\*allocates 1 part height to one row and 3 parts to second row \*/

1. IF WE USED AUTO FOR ANY COLUMNS WIDTH IT WILL AUTOMtically adjust the height to 100 % for that particular columns of remaining space
2. If we want to give more space to some columns if space if available depending upon the size of screen we can do so by using minmax( minwidth , maxwidth) function.

.minmaxcontainer {

    display: grid;

    grid-template-columns: 100px minmax(300px , 800px)  minmax(100px , 500px);

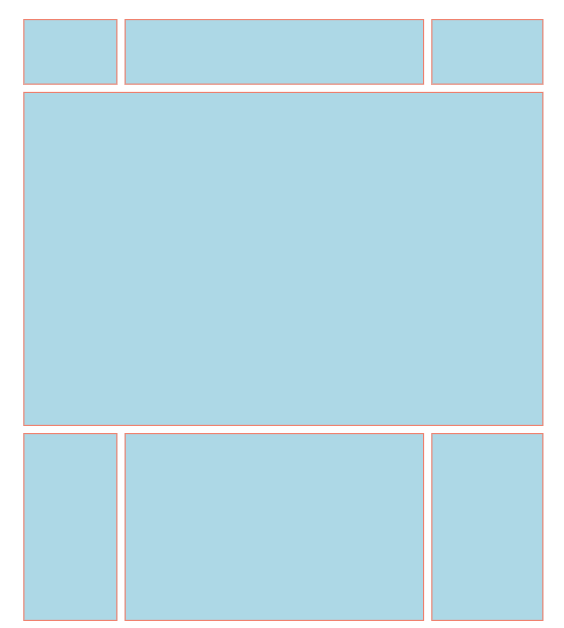
    grid-template-rows:70px minmax(100px,200px)  70px 70px 200px;

    margin: 2rem;

    gap: 0.5rem;

}

This is the output for above code



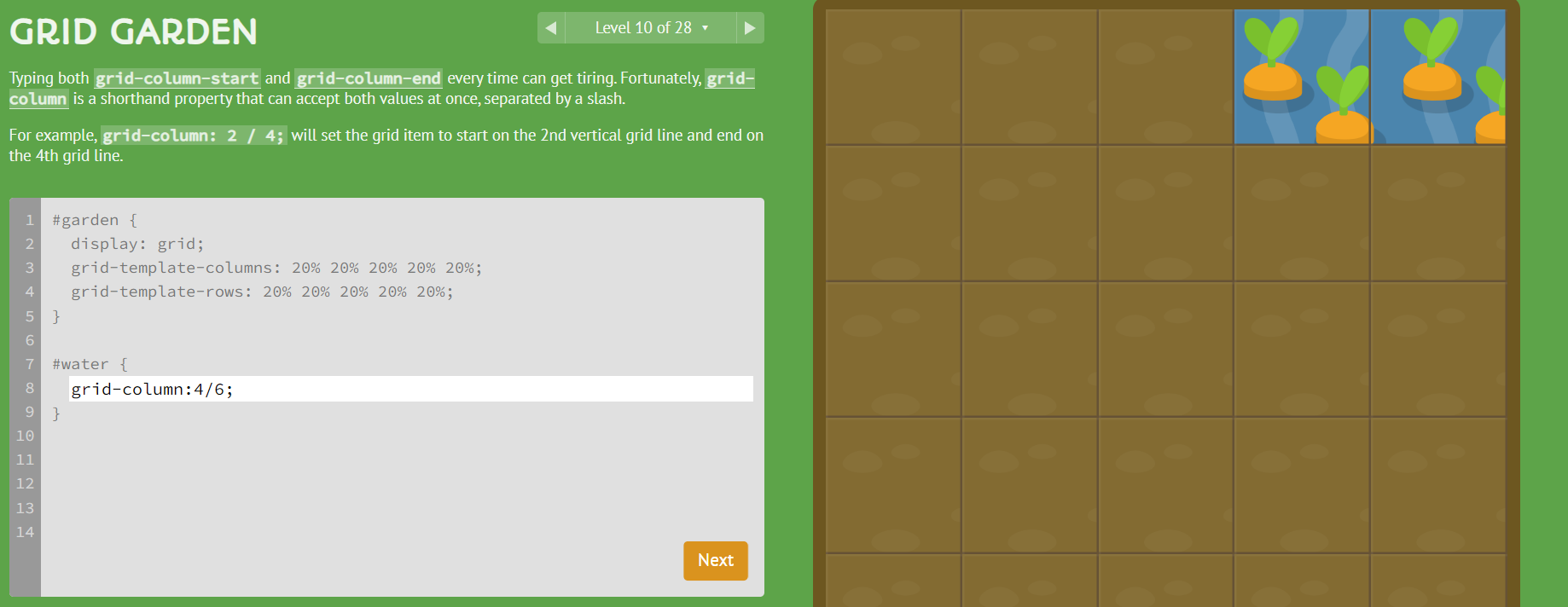
There are there columns and 5 rows .

The center area :

grid-column:span 3 (occu[ies space of three columns )

Grid-row:span 3; (occupies the space of three rows )

Note: If we create rows that don’t fit the mentioned propty for grid we can set grid-auto-rows and grid-auto-columns width



## Css code

Here there are two classes 1. Grid-container 2.grid-items

.grid-container {

display: grid;

grid-template-columns: repeat(3, 1fr); /\* Three equal columns \*/

grid-template-rows: auto; /\* Rows adjust based on content \*/

gap: 10px; /\* Space between grid items \*/

padding: 10px;

background-color: #f5f5f5;

}

.grid-item {

background-color: #2196F3;

color: white;

border-radius: 4px;

padding: 20px;

font-size: 20px;

text-align: center;

}

## Html code

<div class="grid-container">

<div class="grid-item">1</div>

<div class="grid-item">2</div>

<div class="grid-item">3</div>

<div class="grid-item">4</div>

<div class="grid-item">5</div>

<div class="grid-item">6</div> </div >

# CSS CALC ( ) Function

This function is used to calculate the max ,min screen sizes etc,… and make changes in the layout of our webpage based on those calculations.

## Example:

Width: max(80 % , 300px ) …… It means that my div will have size 80%of the screen for screens having width >= 300px . if Screen size is less than 300px scroll bar will be introduced.

## CSS Math Functions

|  |  |
| --- | --- |
| **Function** | **Description** |
| [calc()](https://www.w3schools.com/cssref/func_calc.asp) | Allows you to perform calculations to determine CSS property values |
| [max()](https://www.w3schools.com/cssref/func_max.asp) | Uses the largest value, from a comma-separated list of values, as the property value |
| [min()](https://www.w3schools.com/cssref/func_min.asp) | Uses the smallest value, from a comma-separated list of values, as the property value |

# Media Queries

There are two types ;

1. Mobile-first
2. Desktop-first

Sample and CSS code of both is given.

The mobile-first pre-defined media queries:

/\* xs \*/

/\* @media (min-width: 475px) {} \*/

/\* sm \*/

/\* @media (min-width: 640px) {} \*/

/\* md \*/

/\* @media (min-width: 768px) {} \*/

/\* lg \*/

/\* @media (min-width: 1024px) {} \*/

/\* xl \*/

/\* @media (min-width: 1280px) {} \*/

/\* 2xl \*/

/\* @media (min-width: 1536px) {} \*/

The desktop-first pre-defined media queries:

/\* 2xl \*/

@media (max-width: 1536px) {}

/\* xl \*/

@media (max-width: 1280px) {}

/\* lg \*/

@media (max-width: 1024px) {}

/\* md \*/

@media (max-width: 768px) {}

/\* sm \*/

@media (max-width: 640px) {}

/\* xs \*/

@media (max-width: 475px) {}

Mobile container utility class:

.container {

width: 100%;

margin-left: auto;

margin-right: auto;

padding-left: 0.5rem;

padding-right: 0.5rem;

}

/\* xs \*/

@media (min-width: 475px) {

.container {

max-width: 475px;

}

}

/\* sm \*/

@media (min-width: 640px) {

.container {

max-width: 640px;

}

}

/\* md \*/

@media (min-width: 768px) {

.container {

max-width: 768px;

}

}

/\* lg \*/

@media (min-width: 1024px) {

.container {

max-width: 1024px;

}

}

/\* xl \*/

@media (min-width: 1280px) {

.container {

max-width: 1280px;

}

}

/\* 2xl \*/

@media (min-width: 1536px) {

.container {

max-width: 1536px;

}

}

Desktop container utility class:

.container {

max-width: 1536px;

margin-left: auto;

margin-right: auto;

padding-left: 0.5rem;

padding-right: 0.5rem;

}

/\* 2xl \*/

@media (max-width: 1536px) {

.container {

max-width: 1280px;

}

}

/\* xl \*/

@media (max-width: 1280px) {

.container {

max-width: 1024px;

}

}

/\* lg \*/

@media (max-width: 1024px) {

.container {

max-width: 768px;

}

}

/\* md \*/

@media (max-width: 768px) {

.container {

max-width: 640px;

}

}

/\* sm \*/

@media (max-width: 640px) {

.container {

max-width: 475px;

}

}

/\* xs \*/

@media (max-width: 475px) {

.container {

width: 100%;

}

………………………………………………

…………………………………………………………………….

……………………………………………….

Here is the reorganized version with headers and code sections:

## Mobile-First Media Queries

Breakpoints:

- xs: 475px

- sm: 640px

- md: 768px

- lg: 1024px

- xl: 1280px

- 2xl: 1536px

Code:

/\* xs \*/

@media (min-width: 475px) {}

/\* sm \*/

@media (min-width: 640px) {}

/\* md \*/

@media (min-width: 768px) {}

/\* lg \*/

@media (min-width: 1024px) {}

/\* xl \*/

@media (min-width: 1280px) {}

/\* 2xl \*/

@media (min-width: 1536px) {}

## Desktop-First Media Queries

Breakpoints:

- 2xl: 1536px

- xl: 1280px

- lg: 1024px

- md: 768px

- sm: 640px

- xs: 475px

Code:

/\* 2xl \*/

@media (max-width: 1536px) {}

/\* xl \*/

@media (max-width: 1280px) {}

/\* lg \*/

@media (max-width: 1024px) {}

/\* md \*/

@media (max-width: 768px) {}

/\* sm \*/

@media (max-width: 640px) {}

/\* xs \*/

@media (max-width: 475px) {}

## Mobile Container Utility Class

Default Styles:

- width: 100%

- margin-left: auto

- margin-right: auto

- padding-left: 0.5rem

- padding-right: 0.5rem

Code:

.container {

width: 100%;

margin-left: auto;

margin-right: auto;

padding-left: 0.5rem;

padding-right: 0.5rem;

}

/\* xs \*/

@media (min-width: 475px) {

.container {

max-width: 475px;

}

}

/\* sm \*/

@media (min-width: 640px) {

.container {

max-width: 640px;

}

}

/\* md \*/

@media (min-width: 768px) {

.container {

max-width: 768px;

}

}

/\* lg \*/

@media (min-width: 1024px) {

.container {

max-width: 1024px;

}

}

/\* xl \*/

@media (min-width: 1280px) {

.container {

max-width: 1280px;

}

}

/\* 2xl \*/

@media (min-width: 1536px) {

.container {

max-width: 1536px;

}

}

## Desktop Container Utility Class

Default Styles:

- max-width: 1536px

- margin-left: auto

- margin-right: auto

- padding-left: 0.5rem

- padding-right: 0.5rem

Code:

.container {

max-width: 1536px;

margin-left: auto;

margin-right: auto;

padding-left: 0.5rem;

padding-right: 0.5rem;

}

/\* 2xl \*/

@media (max-width: 1536px) {

.container {

max-width: 1280px;

}

}

/\* xl \*/

@media (max-width: 1280px) {

.container {

max-width: 1024px;

}

}

/\* lg \*/

@media (max-width: 1024px) {

.container {

max-width: 768px;

}

}

/\* md \*/

@media (max-width: 768px) {

.container {

max-width: 640px;

}

}

/\* sm \*/

@media (max-width: 640px) {

.container {

max-width: 475px;

}

}

/\* xs \*/

@media (max-width: 475px) {

.container {

width: 100%;

}

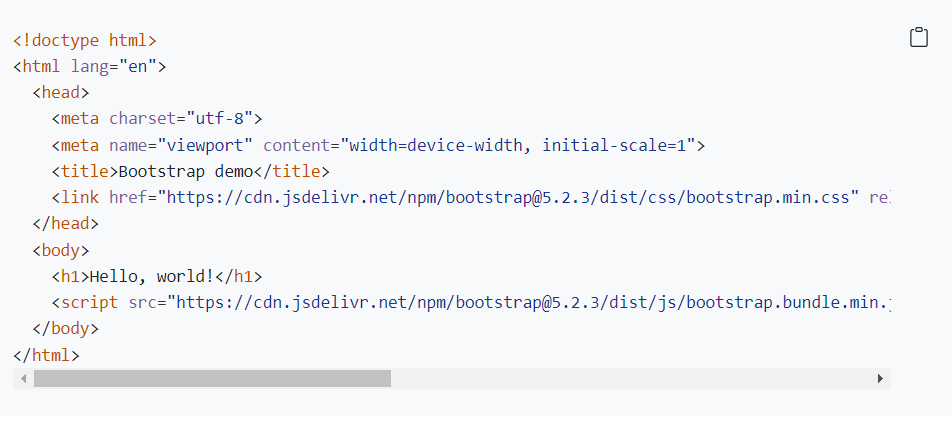
}

# Bootstrap

Open source css framework no need to write css . good for making fast and efficient mobile-first websites websites

If we have add CSS file as well besides we have to make sure that it’s link is after the framework link in the header section.

## How to use;



In the head section :

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">

Just before the end of body section :

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>

We add script if we want to use animation for styles applies as well .

In boostrap we have to use to just the classes inside the html.

## Bootstrap layouts

Boostrap layout used 12 columns by-default . If we want our div to occupy 2 columns inside class we may use col-2 .

If we want to occupy 5-columns . we may use col-5 etc,…

Important :

There can multiple definitions of col inside a single div . If we write

<div **col-sm-12**  **col-md-8 col-lg-4** > it meands that this div will take 12 of 12 columns on mobile , 8/12 columns on tablets and on large divices

Components can be downloaded from their website.

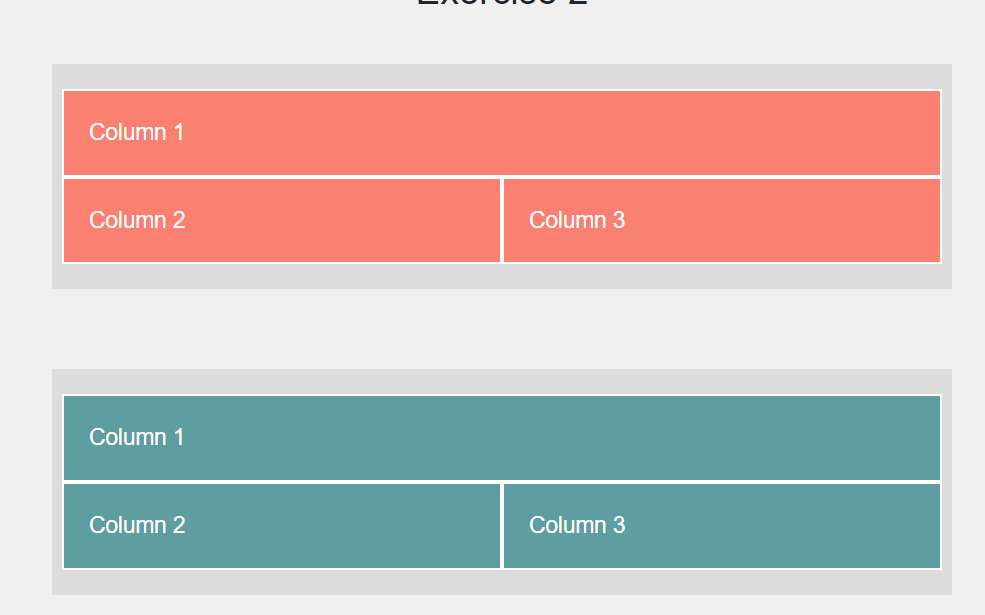
## Layouts

## Link to understand this with practice:

## [Click-me to open github web for practice](https://appbrewery.github.io/bootstrap-layout/)

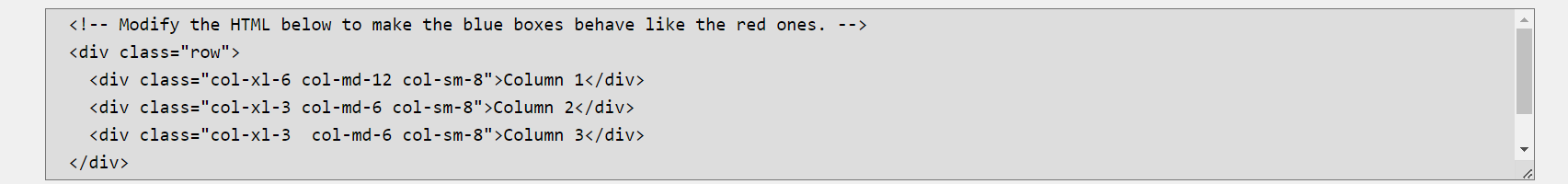
## Large

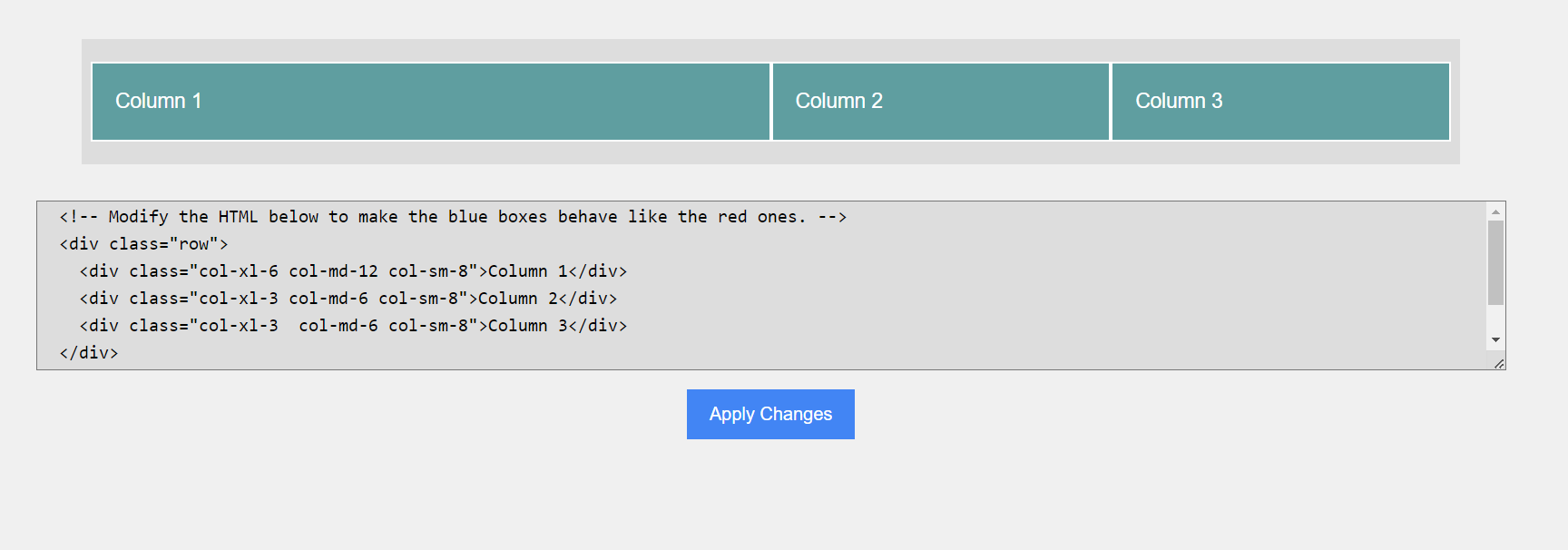


Medium : 

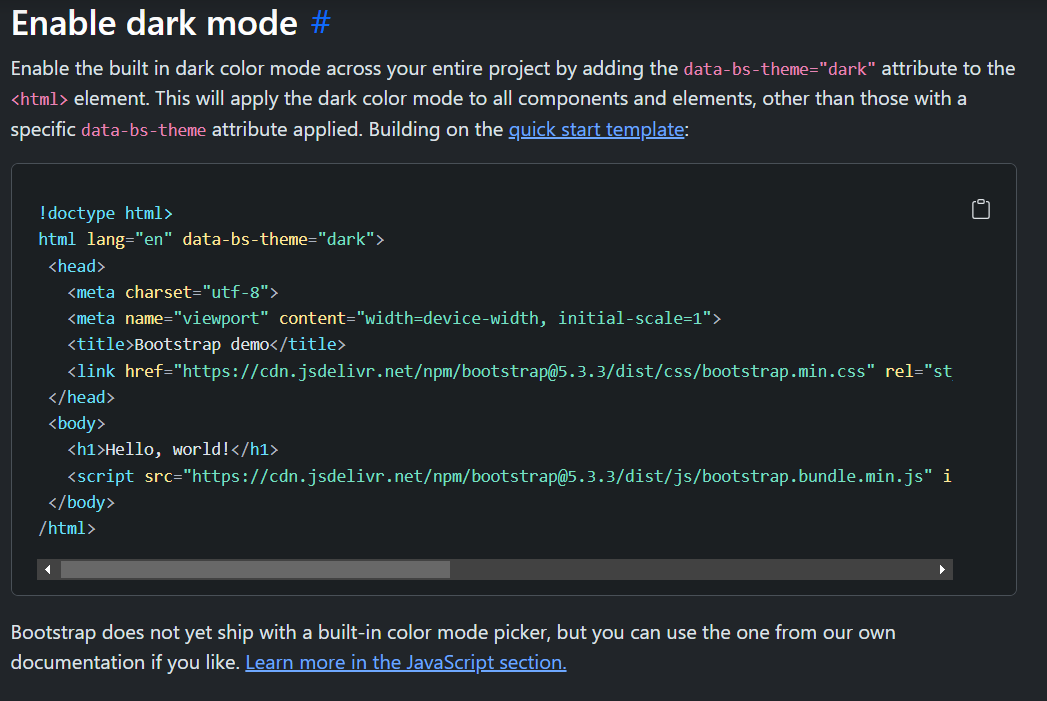
Small :

Code





## Dark mode using Bootstrap



# WEBDESIGN

## Color scheme:

This should be used for logos , headers , the things we want to stand out not every text because it makes text difficult to read.

1. Red conveys love, energy, intensity. Which is why you see a lot of car adverts which have a red.
2. Yellow: logo , headline >>> not for something to be read for large period of time.
3. Green which is all about freshness, growth, safety.

And that's why you see a lot of grocery companies like HelloFresh or Amazon Groceries use this color

as their predominant color palette. Because it's trying to tell you, "Hey look! This thing is edible.

This is what we're selling."

do with food, this is usually a safe bet to go with.

1. Now Blue is about stability, trust, serenity. And so you see a lot of companies which need to convey that

"Hey, we're really trustworthy"

use this color. For example,

look at all the financial companies, look at companies like PayPal or the cryptocurrency companies like

Coinbase. They all choose this color because they want you to know that they're not going to run away

with your money.

In fact, there was a really interesting psychology study where they compared interviewees wearing different colored shirts.

So for example, a group of them would wear a green shirt,

another group would wear a blue shirt and they compared how they were perceived by the interviewers.

And it showed that just by changing the color of your shirt to a blue one you vastly increase the amount

of trust in you

that's given by other people.

So have a think about that the next time you're going for an interview.

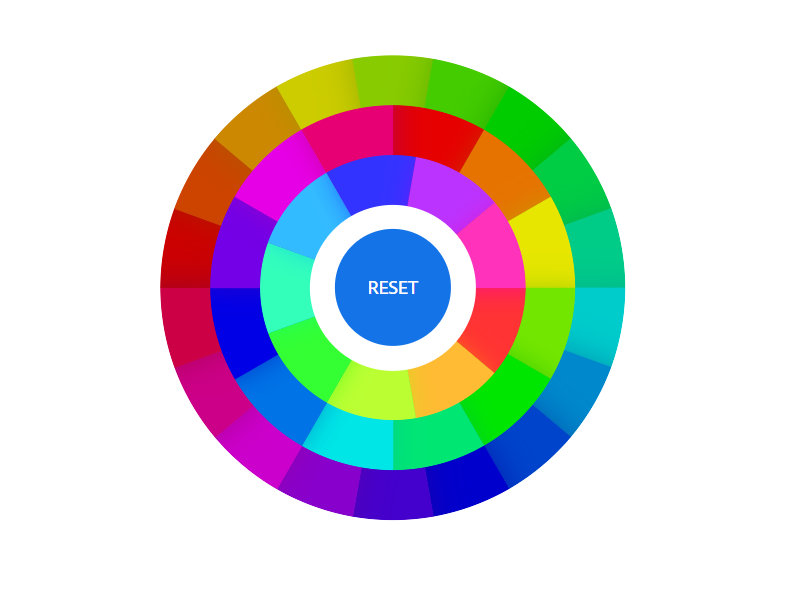
1. Now the final color is purple and it's meant to signify royalty, wealth, femininity. And strangely enough

I've come across a number of payday loan companies which use this color palette especially ones which

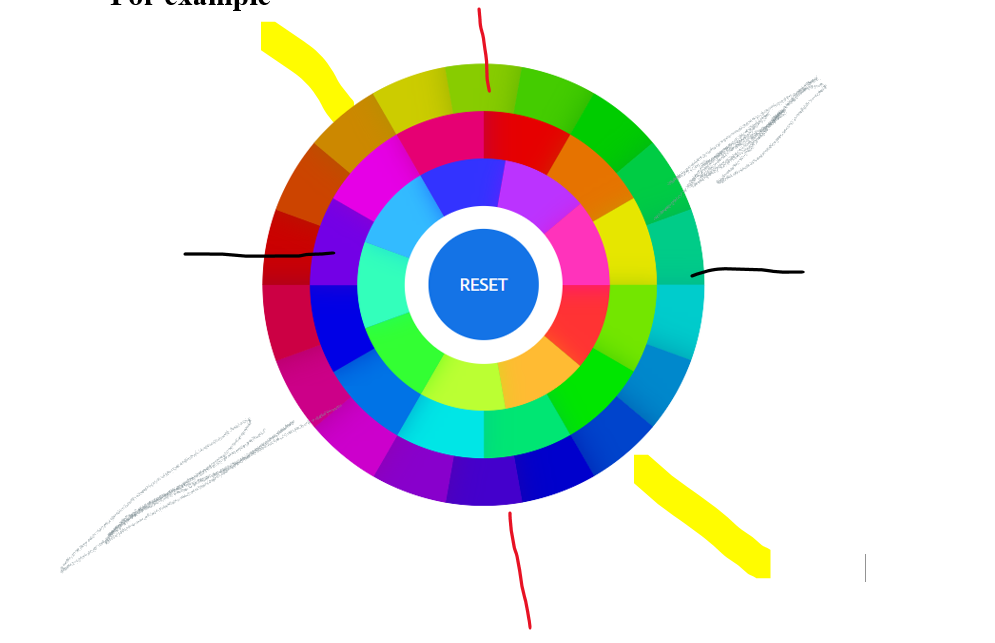
are targeted at women.

So, the important thing to think about is that you're making a conscious decision when you're choosing

a color palette.

It’s always better to select the opposite colors for contrasting along a straight line.

For example



**Here best practice is using the constrasting colors for designing as shown by line …..each color with its exactly opposite on the straight line.**

## Website to select colors pallets ;

## [Adoble-colors](https://color.adobe.com/create/color-wheel)

Make triangle , circle , square of pallets to choose design that fits the best.

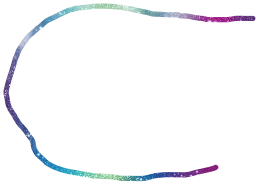
Anoteer website

## [Colorhunt.co](https://colorhunt.co/)

Designed by professionals

# Typography :

Serif : old , authoritative , serious . three types old , average , modern.



What font represent

Font -subfamily

From the article :

<https://medium.com/my-journey-to-the-cloud/css-typography-writing-wholesome-texts-b849265c7278>

## user experience :

Use F- pattern or Z- pattern for placing important items on the page. Important items such as logos , navigation bar , images and heading or thing which we want user to see first is places along the tracing of these alphabets on a full screen.

**Daily-Ui** is website for making a web-design daily to make you a professional Designer.