# /\*\*\*\*\*\*\*1.CSS Examples\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*initial HTML code for static site\*\*\*\*\*\*\*\*/

/\* http://meyerweb.com/eric/tools/css/reset/ 2. v2.0 | 20110126

License: none (public domain) \*/

html, body, div, span, applet, object, iframe,

h1, h2, h3, h4, h5, h6, p, blockquote, pre,

a, abbr, acronym, address, big, cite, code,

del, dfn, em, img, ins, kbd, q, s, samp,

small, strike, strong, sub, sup, tt, var,

b, u, i, center,

dl, dt, dd, ol, ul, li,

fieldset, form, label, legend,

table, caption, tbody, tfoot, thead, tr, th, td,

article, aside, canvas, details, embed,

figure, figcaption, footer, header, hgroup,

menu, nav, output, ruby, section, summary,

time, mark, audio, video {

margin: 0;

padding: 0;

border: 0;

font-size: 100%;

font: inherit;

vertical-align: baseline;

}

/\* HTML5 display-role reset for older browsers \*/

article, aside, details, figcaption, figure,

footer, header, hgroup, menu, nav, section {

display: block;

}

body {

line-height: 1;

}

ol, ul {

list-style: none;

}

blockquote, q {

quotes: none;

}

blockquote:before, blockquote:after,

q:before, q:after {

content: '';

content: none;

}

table {

border-collapse: collapse;

border-spacing: 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Dynamic Table Generator \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<https://divtable.com/table-styler/>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*Example for Media Screen\*\*\*\*\*\*\*\*/

<https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule#Conditional_group_rules>

<https://drafts.csswg.org/css-conditional-3/#cssconditionrule>

<https://www.w3.org/TR/css3-mediaqueries/>

/\*\*\*\*low\*\*/

<https://www.cssmine.com/ebook/css3-media-queries>

/\*\*\*\*\*\*\*\*\*vertical align image and text\*\*\*\*\*\*\*/

<div class="parent">

<div class="child">

<img src="Leaf.svg"/>

<p>text1</p>

</div>

<div class="child">

<img src="Lips.svg"/>

<p>text2</p>

</div>

</div>

.parent{

display:flex;

}

.child{

display: flex;

flex-direction: column;

justify-content: center;

}

.child img { width: 100px; height: 100px; }

/\*\*\*\*\*\*\*\*\*replace read more text into more info\*\*\*\*\*\*\*/

<div>

<a class="btn">Read more</a>

</div>

div a.btn {

visibility: hidden !important;

}

div a.btn:after {

content: 'More info' !important;

visibility: visible !important;

display: block !important;

color: #fff;

background: #007BB2;

font-size: 14px;

padding: 10px 20px;

border: 2px solid #007BB2;

border-radius: 30px;

}

/\*\*\*\*\*\*\*\*\*\*\*Reorder div\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<div class="parent">

<div class="child1">col1</div>

<div class="child2">col2</div>

</div>

.parent {

**display: flex;**

**flex-wrap: wrap;**

}

.child1 {

order: 2;

}

.child2 {

order: 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*Grid vertically equal height\*\*\*\*\*\*\*\*\*\*\*\*/

<div class="parent row">

<div class="grid1 col"></div>

<div class="grid2 col"></div>

<div class="grid3 col"></div>

</div>

.parent {

display: flex;

flex-wrap: wrap;

}

**[OR]**

.row

{

display: -ms-flexbox;

display: -webkit-flex;

display: flex; /\* equal height of the children \*/

}

.col {

-webkit-flex: 1;

-ms-flex: 1;

flex: 1; /\* additionally, equal width \*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*Grid Center\*\*\*\*\*\*\*\*\*\*\*\*/

.cat-grid div {

overflow: hidden;

margin-bottom: 3%;

text-align: center;

}

.cat-grid h2 {

margin: 0;

position: absolute;

top: 50%;

left: 50%;

-ms-transform: translate(-50%, -50%);

transform: translate(-50%, -50%);

color: #ffffff;

white-space: nowrap;

}

/\*\*\*\*\*\*\*\*\*\*\*Grid align center\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

grid-item {

position: relative;

text-align: center;

}

span {

position: absolute;

left: 50%;

top: 50%;

transform: translate(-50%, -50%);

}

<https://www.freecodecamp.org/news/flexbox-the-ultimate-css-flex-cheatsheet/>

## All CSS Flexbox properties

Here is a list of all the CSS flexbox properties that can be used to position elements in CSS. Next, you'll see how they work.

### ****CSS that can be applied to the container****

display: flexbox | inline-flex;

flex-direction: row | row-reverse | column | column-reverse;

flex-wrap: nowrap | wrap | wrap-reverse;

flex-flow: <‘flex-direction’> || <‘flex-wrap’>

justify-content: flex-start | flex-end | center | space-between | space-around;

align-items: flex-start | flex-end | center | baseline | stretch;

align-content: flex-start | flex-end | center | space-between | space-around | stretch;

### ****CSS that can be applied to items/elements in the container****

order: <integer>;

flex-grow: <number>; /\* default 0 \*/

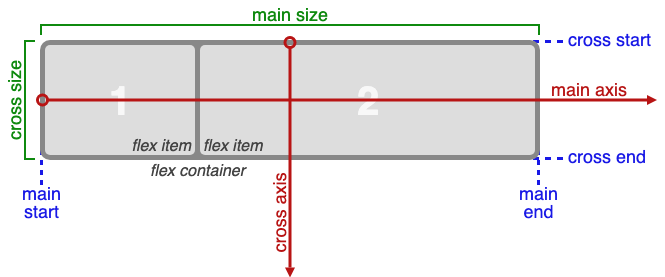
flex-shrink: <number>; /\* default 1 \*/

flex-basis: <length> | auto; /\* default auto \*/

flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]

align-self: auto | flex-start | flex-end | center | baseline | stretch;

**Terminology**

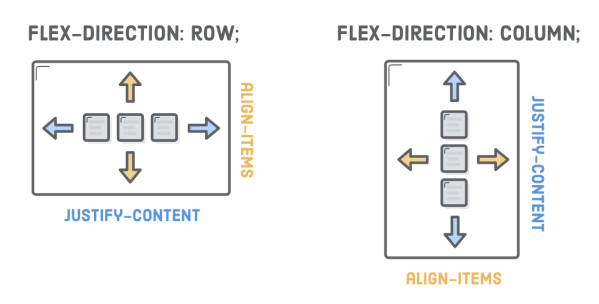


**List of CSS Properties:**

<https://www.geeksforgeeks.org/css-properties-complete-reference/>

**Support properties:**

**Flex:**



* Use display: flex; to create a flex container.
* Use justify-content to define the **horizontal** alignment of items.
* Use align-items to define the **vertical** alignment of items.
* Use flex-direction if you need columns instead of rows.
* Use the row-reverse or column-reverse values to flip item order.
* Use order to customize the order of **individual** elements.
* Use align-self to vertically align **individual** items.
* Use flex to create flexible boxes that can stretch and shrink.

Parent Element (Container)

The flex container becomes flexible by setting the display property to *flex*:

The flex container properties are:

* [flex-direction](https://www.w3schools.com/css/css3_flexbox.asp#flex-direction)
* [flex-wrap](https://www.w3schools.com/css/css3_flexbox.asp#flex-wrap)
* [flex-flow](https://www.w3schools.com/css/css3_flexbox.asp#flex-flow)
* [justify-content](https://www.w3schools.com/css/css3_flexbox.asp#justify-content)
* [align-items](https://www.w3schools.com/css/css3_flexbox.asp#align-items)
* [align-content](https://www.w3schools.com/css/css3_flexbox.asp#align-content)

Child Elements (Items)

The direct child elements of a flex container automatically becomes flexible (flex) items.

The flex item properties are:

* [order](https://www.w3schools.com/css/css3_flexbox.asp#order)
* [flex-grow](https://www.w3schools.com/css/css3_flexbox.asp#flex-grow)
* [flex-shrink](https://www.w3schools.com/css/css3_flexbox.asp#flex-shrink)
* [flex-basis](https://www.w3schools.com/css/css3_flexbox.asp#flex-basis)
* [flex](https://www.w3schools.com/css/css3_flexbox.asp#flex)
* [align-self](https://www.w3schools.com/css/css3_flexbox.asp#align-self)

**Position:**

**position**ed using the **top, bottom, left, and right** properties, and **Z-index, Clip**

These properties will not work unless the position property is set first.

z-index only works on positioned elements

**display: flex;**

**flex-direction: row;**

**flex-wrap: nowrap;**

**justify-content: flex-start;**

**align-items: stretch;**

**align-content: stretch;**

**CSS Combinators:**

* descendant selector (space)
* child selector (>) – (all child elements)
* adjacent sibling selector (+) - (direct sibiling only)
* general sibling selector (~) - (all sibilings)

**CSS Pseudo Classes:**

<https://www.w3schools.com/css/css_pseudo_classes.asp>

a:active

input:checked

input:disabled - input[type=text]:disabled

p:empty

input:enabled - input[type=text]:enabled

p:first-child

p:first-of-type

input:focus

a:hover

input:in-range

input:invalid

p:lang(it)

p:last-child

p:last-of-type

a:link

:not(p)

p:nth-child(2)

p:nth-last-child(2)

p:nth-last-of-type(2)

p:nth-of-type(2)

p:only-of-type

p:only-child

input:optional

input:out-of-range

input:read-only

input:read-write

input:required

root

#news:target

input:valid

a:visited

**CSS Pseudo Elements:**

p::after

p::before

p::first-letter

p::first-line

p::selection

## CSS Attribute Selectors

[target] -Selects all elements with a target attribute

[target=\_blank] -Selects all elements with target="\_blank"

[title~=flower] -Selects all elements with a title attribute **containing** the word "flower"

[lang|=en] -Selects all elements with a lang attribute value **starting** with "en"

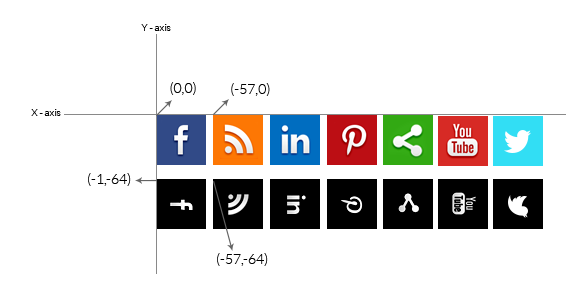
a[href^="https"] -Selects every <a > element whose href attribute value **begins** with "https"

a[href$=".pdf"] -Selects every <a > element whose href attribute value **ends** with ".pdf"

a[href\*="w3schools"] -Selects every <a > element whose href attribute value **contains** the substring "w3schools"

**Image Sprites:**

Graphical position of background.



.class {

background: url('imgs/image\_sprites.png');

background-position: -57px -64px;

width: 50px;

height: 50px;

}

**shorthand property:**

/\*\*\*\*\*\*\*\*\*\*\*Change Placeholder color\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

input::placeholder {

color: red;

}

::-webkit-input-placeholder {

color: #707070 !important;

font-size: 18px;

line-height: 21px !important;

}

/\*\*\*\*\*\*\*\*\*\*\* align li horizontal with full width container equally \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<https://jsfiddle.net/csswizardry/zfSt4/>

<style>

/\*\*

\* Tabs object.

\* 1. Tables (kinda) for layout!

\* 2. This is the magic bit; make all children occupy equal width.

\* 3. Required to make the tabs fill their container.

\* 4. Make each tab pack up horizontally.

\* 5. Ensure the hit area covers the whole tab.

\*/

.tabs {

margin: 0;

padding: 0;

list-style: none;

display: table; /\* [1] \*/

table-layout: fixed; /\* [2] \*/

width: 100%; /\* [3] \*/

}

.tabs\_item {

display: table-cell; /\* [4] \*/

}

.tabs\_link {

display: block; /\* [5] \*/

}

/\*\*

\* Primary nav. Extends `.tabs`.

\* 1. Stop tabs’ corners leaking out beyond our 4px round.

\*/

.primary-nav {

text-align: center;

border-radius: 4px;

overflow: hidden; /\* [1] \*/

}

.primary-nav a {

padding: 1em;

background-color: #BADA55;

color: #fff;

font-weight: bold;

text-decoration: none;

}

.primary-nav a:hover {

background-color: #A3C43B;

}

</style>

<ul class="tabs primary-nav">

<li class="tabs\_item">

<a href="#" class="tabs\_link">Home</a>

</li>

<li class="tabs\_item">

<a href="#" class="tabs\_link">About</a>

</li>

<li class="tabs\_item">

<a href="#" class="tabs\_link">Work</a>

</li>

<li class="tabs\_item">

<a href="#" class="tabs\_link">Contact</a>

</li>

</ul>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FLEXBOX\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\* <https://developer.mozilla.org/en-US/docs/Web/CSS/justify-content>

<https://gedd.ski/post/how-to-pin-element-to-center-of-parent/>

<https://thoughtbot.com/blog/positioning>

\* <https://flexboxfroggy.com/>

* **justify-content**
* **align-items**
* **flex-direction**
* **order**
* **align-self**
* **flex-wrap**
* **flex-flow**
* **align-contenthttps://code.org/api/hour/finish_flexbox_froggy.png**

**justify-content**

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lilypad on the right by using the justify-content property, which aligns items horizontally and accepts the following values:

* **flex-start:** Items align to the left side of the container.
* **flex-end:** Items align to the right side of the container.
* **center:** Items align at the center of the container.
* **space-between:** Items display with equal spacing between them.
* **space-around:** Items display with equal spacing around them.

For example, justify-content: flex-end; will move the frog to the right. https://code.org/api/hour/begin_flexbox_froggy.png

**align-items or align-self**

flex-start: Items align to the left side of the container.

flex-end: Items align to the right side of the container.

center: Items align at the center of the container.

baseline

stretch

**flex-direction:**

* **row:** Items are placed the same as the text direction.
* **row-reverse:** Items are placed opposite to the text direction.
* **column:** Items are placed top to bottom.
* **column-reverse:** Items are placed bottom to top.

/\*\*\*\*\*\*\*\*\*use of selectors\*\*\*\*\*\*\*\*\*\*/

<div class=”img-box”>

<div class=”info-box”><p>test</p></div> <!--(+ symbol pointing immediate next to img-box class)-->

<div class=”info-box”></div>

</div>

**.img-box + .info-box p {**

display: none;

}

/\*\*\*\*\*\*\*\*\*Url Break\*\*\*///////////

a {

**overflow-wrap: break-word !important;**

**word-break: break-all;**

text-decoration: underline;

}

<a href="https://www.emralliance.org/specific\_assessments.pdf"> https://www.emralliance.org/rtuuyhhg/vnmcxnfkjdkf/kdskdkfjirieuri/specific\_assessments.pdf </a>

<div class="container">  
  <div class="vertical-center">  
    <p>I am vertically centered.</p>  
  </div>  
</div>

<style>  
.container {   
  height: 200px;  
  position: relative;  
  border: 3px solid green;   
}  
.vertical-center {  
  margin: 0;  
  position: absolute;  
  top: 50%;  
  -ms-transform: translateY(-50%);  
  transform: translateY(-50%);  
}  
</style>

<div class="container">  
  <div class="center">  
    <p>I am vertically and horizontally centered.</p>  
  </div>  
</div>

<style>  
.container {   
  height: 200px;  
  position: relative;  
  border: 3px solid green;   
}  
.center {  
  margin: 0;  
  position: absolute;  
  top: 50%;  
  left: 50%;  
  -ms-transform: translate(-50%, -50%);  
  transform: translate(-50%, -50%);  
}  
</style>

/\*\*\*\*\*\*\*\*\*\*Justify column middle\*\*\*\*\*\*\*\*\*/

.parent {

**display: flex;**

**justify-content: center;** <!--(for horizondal alignment)-->

align-items: center; <!--(for vertical alignment)-->

}

<div class="grandparent">

<div class="parent">

<div class="childcolumn1">

<div class="grandchild"></div>

</div>

<div class="childcolumn2"></div>

</div>

</div>

/\*\*\*\*\*\*\*\*\*\*\*\*\* (+,> and ~) in CSS selector\*\*\*\*\*\*\*\*\*\*\*\*\*/

<div id="container">

<p>First</p>

<div>

<p>Child Paragraph</p>

</div>

<p>Second</p>

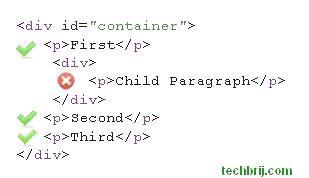
<p>Third</p>

</div>

div#container > p {

border: 1px solid black;

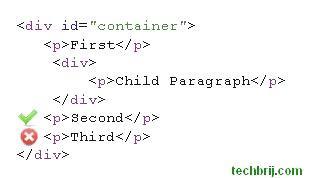
}



div + p {

color: green;

}



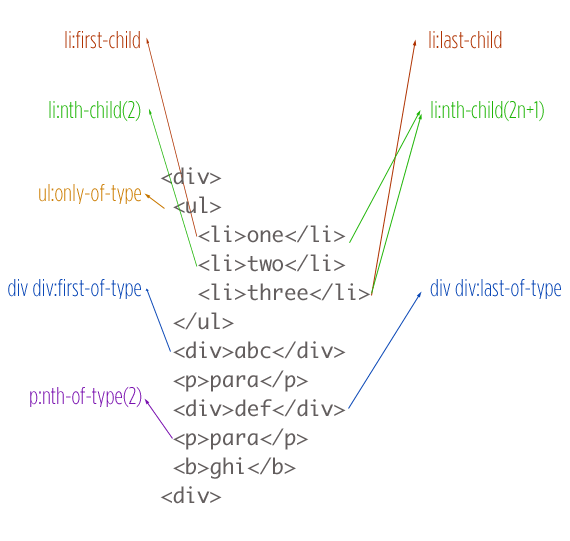
div ~ p {

background-color: blue;

}



<https://css-tricks.com/pseudo-class-selectors/>

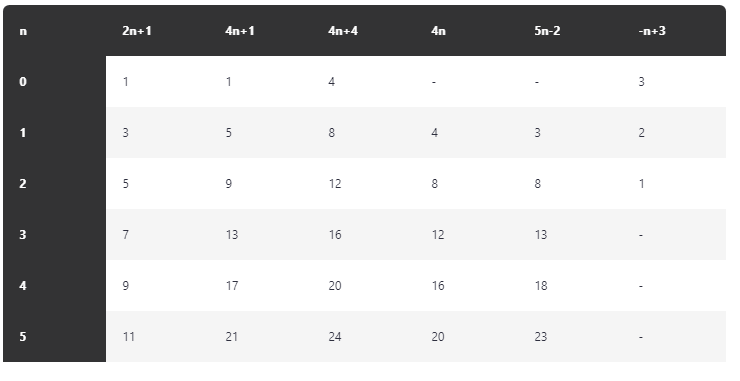


<http://api.jquery.com/category/events/browser-events/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selectors>

<https://www.geeksforgeeks.org/css-indeterminate-selector/>

<https://css-tricks.com/how-nth-child-works/>



**Bootstrap 4** grid system works across different screen sizes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Extra small (<576px) | Small  (>=576px) | Medium  (>=768px) | Large  (>=992px) | Extra Large (>=1200px) |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| Grid behaviour | Horizontal at all times | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints |
| Container width | None (auto) | 540px | 720px | 960px | 1140px |
| Suitable for | Portrait phones | Landscape phones | Tablets | Laptops | Laptops and Desktops |
| # of columns | 12 | 12 | 12 | 12 | 12 |
| Gutter width | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) | 30px (15px on each side of a column) |
| Nestable | Yes | Yes | Yes | Yes | Yes |
| Offsets | Yes | Yes | Yes | Yes | Yes |
| Column ordering | Yes | Yes | Yes | Yes | Yes |

Bootstrap:

<https://getbootstrap.com/docs/4.0/layout/grid/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Extra small <576px | Small ≥576px | Medium ≥768px | Large ≥992px | Extra large ≥1200px |
| Max container width | None (auto) | 540px | 720px | 960px | 1140px |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| # of columns | | | 12 | | |
| Gutter width | | | 30px (15px on each side of a column) | | |
| Nestable | | | Yes | | |
| Column ordering | | | Yes | | |

col-xl-4 col-lg-4 col-md-4 col-sm-4 col-4

<https://getbootstrap.com/docs/3.4/css/>

## /\*\*\*\*\*\*\*\*\*\*\*\*\*Create the CSS Grid \*\*\*\*\*\*\*\*\*\*\*\*/

<https://www.joomlashack.com/blog/tutorials/center-and-align-items-in-css-grid/>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Others\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**white-space: nowrap;**

## /\*\*\*\*\*\*\*\*\*\*\*Box Alignment Properties \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

align-items: baseline;

justify-items: left;

align-self: stretch;

justify-self: start;

/\*\*\*\*\*\*\*\*\*\*\*\*Clip path generator \*\*\*\*\*\*\*\*\*\*\*\*/

<https://bennettfeely.com/clippy/>

<http://www.cssplant.com/clip-path-generator>

<https://css-tricks.com/sketchy-avatars-css-clip-path/> - with points

<https://developer.mozilla.org/en-US/docs/Web/CSS/clip-path>

<https://www.uplabs.com/posts/clip-path-generator>

<http://karenmenezes.com/shapes-polygon/>

/\*\*\*\*\*\*\*\*\*\*\*\*CRABTREE FITNESS SHAPE ALIGNMENT WITH TEXT WRAP \*\*\*\*\*\*\*\*\*\*\*\*/

#accordions-3879 .accr2img {

width: 61%;

margin-top: -148px;

float: right;

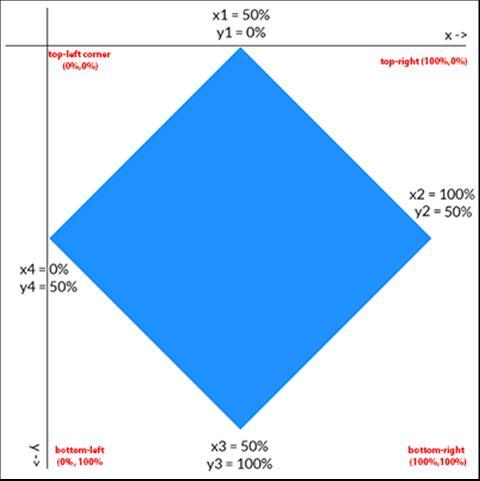
shape-image-threshold: 0.5;

shape-outside: polygon(0 0, 0 0, 42% 100%, 100% 100% );

margin-left: 3px;

margin-bottom: 14px;

}

1. [<basic-shape>](https://developer.mozilla.org/en-US/docs/Web/CSS/basic-shape)
2. [shape-image-threshold](https://developer.mozilla.org/en-US/docs/Web/CSS/shape-image-threshold)
3. [shape-margin](https://developer.mozilla.org/en-US/docs/Web/CSS/shape-margin)
4. [shape-outside](https://developer.mozilla.org/en-US/docs/Web/CSS/shape-outside)

/\* Keyword values \*/

shape-outside: none;

shape-outside: margin-box;

shape-outside: content-box;

shape-outside: border-box;

shape-outside: padding-box;

/\* Function values \*/

shape-outside: circle();

shape-outside: ellipse();

shape-outside: inset(10px 10px 10px 10px);

shape-outside: polygon(10px 10px, 20px 20px, 30px 30px);

/\* <url> value \*/

shape-outside: url(image.png);

/\* <gradient> value \*/

shape-outside: linear-gradient(45deg, rgba(255, 255, 255, 0) 150px, red 150px);

/\* Global values \*/

shape-outside: initial;

shape-outside: inherit;

shape-outside: unset;

/\*\*\*\*\*\*\*\*\*\*\*\***FONT FORMULA**\*\*\*\*\*\*\*\*\*\*\*\*\*/

body {

**font-size: calc([min size] + ([max size] - [min size]) \* ((100vw - [min viewport width]) / ([max viewport width] - [min viewport width])));**

font-size: calc(14px + (26 - 14) \* ((100vw - 300px) / (1600 - 300)));

line-height: calc(1.3em + (1.5 - 1.2) \* ((100vw - 300px)/(1600 - 300)));

font-size: calc(1.5rem + 3vw);

}

<link rel='stylesheet' media='screen and (min-width: 701px) and (max-width: 900px)' href='css/medium.css' />

<link rel='stylesheet' media='screen and (min-width: 901px) and (max-width: 1600px)' href='css/large.css' />

<link rel="stylesheet" media="(max-width: 640px)" href="max-640px.css">

<link rel="stylesheet" media="(min-width: 640px)" href="min-640px.css">

<link rel="stylesheet" media="(orientation: portrait)" href="portrait.css">

<link rel="stylesheet" media="(orientation: landscape)" href="landscape.css">

<script>

// Media wise css append

function adjustStyle(width) {

width = parseInt(width);

if (width < 701) {

$("#size-stylesheet").attr("href", "css/narrow.css");

} else if (width < 900) {

$("#size-stylesheet").attr("href", "css/medium.css");

} else {

$("#size-stylesheet").attr("href", "css/wide.css");

}

}

$(function () {

adjustStyle($(this).width());

$(window).resize(function () {

adjustStyle($(this).width());

});

});

// Chaining Method

$('div').not('.type, .collection').parent();

</script>

**Default Page Widths**

* 320px
* 480px
* 600px
* 768px
* 900px
* 1024px
* 1200px

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

// Meta

<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />

/\* When the width is between 600px and 900px OR above 1100px - change the appearance of <div> \*/

@media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {

}

@media only screen and (min-width: 600px) {

/\* For tablets: \*/

.col-s-1 {

width: 8.33%;

}

.col-s-2 {

width: 16.66%;

}

}

@media only screen and (min-width: 768px) {

/\* For desktop: \*/

.col-1 {

width: 8.33%;

}

.col-2 {

width: 16.66%;

}

}

**img[class="hide"]{**

**display:none;**

**}**

**/\* For mobile phones: \*/**

**[class\*="col-"] {**

**width: 100%;**

**}**

**.row::after {**

**content: "";**

**clear: both;**

**display: table;**

**}**

/\*\*\*\* Update Content in responsive mode \*\*\*\*\*\*/

@media all and (min-width: 1001px) {

#sidebar ul li a:after {

content: " (" attr(data-email) ")";

font-size: 11px;

font-style: italic;

color: #666;

}

}

@media all and (max-width: 1000px) and (min-width: 700px) {

#sidebar ul li a:before {

content: "Email: ";

font-style: italic;

color: #666;

}

}

@media all and (max-width: 699px) and (min-width: 520px), (min-width: 1151px) {

#sidebar ul li a {

padding-left: 21px;

background: url(../images/email.png) left center no-repeat;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\***MEDIA SCREEN EXAMPLES**\*\*\*\*\*\*\*\*\*\*\*\*\*/

@media only screen and (min-width: 100px),

not all and (min-width: 100px),

not print and (min-height: 100px),

(color),

(min-height: 100px) and (max-height: 1000px),

handheld and (orientation: landscape) {

html {

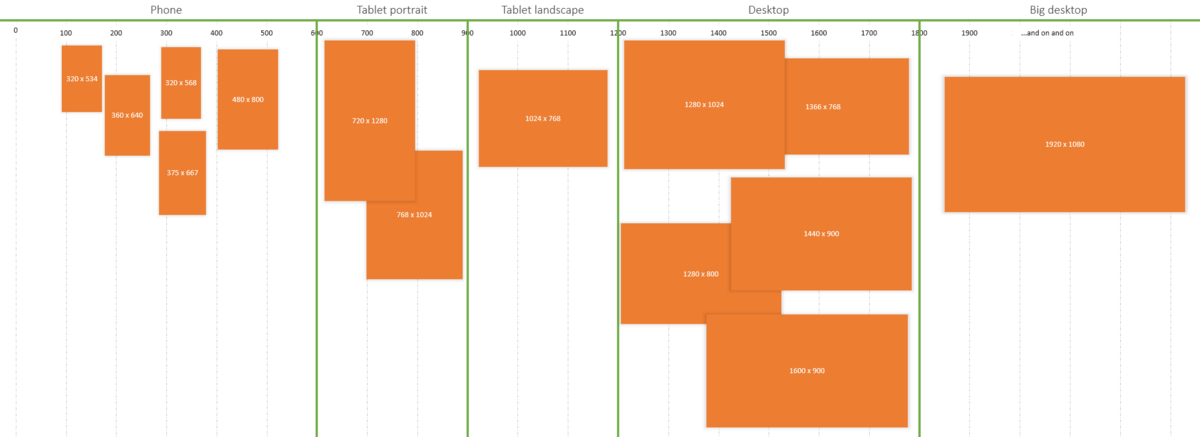
background: red;

}

}

Note:

only keyword was intended to prevent non-media-query supporting browsers to not load the stylesheet or use the styles.



Min-width: 320px (smaller phone viewpoints)

Min-width: 480px (small devices and most phones)

Min-width: 768px (most tablets)

Min-width: 992px (smaller desktop viewpoints)

Min-width: 1200px (large devices and wide screens)

# /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1.1 Table Responsive using css\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

<https://css-tricks.com/responsive-data-tables/>

<table role="table">

<thead role="rowgroup">

<tr role="row">

<th role="columnheader">First Name</th>

<th role="columnheader">Last Name</th>

<th role="columnheader">Job Title</th>

</tr>

</thead>

<tbody role="rowgroup">

<tr role="row">

<td role="cell">James</td>

<td role="cell">Matman</td>

<td role="cell">Chief Sandwich Eater</td>

</tr>

<tr role="row">

<td role="cell">The</td>

<td role="cell">Tick</td>

<td role="cell">Crimefighter Sorta</td>

</tr>

<tr role="row">

<td role="cell">Jokey</td>

<td role="cell">Smurf</td>

<td role="cell">Giving Exploding Presents</td>

</tr>

</tbody>

</table>

<style>

/\*Max width before this PARTICULAR table gets nasty. This query will take effect for any screen smaller than 760px and also iPads specifically. \*/

@media only screen and (max-width: 760px), (min-device-width: 768px) and (max-device-width: 1024px) {

/\* Force table to not be like tables anymore \*/

table, thead, tbody, th, td, tr {

display: block;

}

/\* Hide table headers (but not display: none;, for accessibility) \*/

thead tr {

position: absolute;

top: -9999px;

left: -9999px;

}

tr {

margin: 0 0 1rem 0;

}

tr:nth-child(odd) {

background: #ccc;

}

td {

/\* Behave like a "row" \*/

border: none;

border-bottom: 1px solid #eee;

position: relative;

padding-left: 50%;

}

td:before {

/\* Now like a table header \*/

position: absolute;

/\* Top/left values mimic padding \*/

top: 0;

left: 6px;

width: 45%;

padding-right: 10px;

white-space: nowrap;

}

/\*Label the data

You could also use a data-\* attribute and content for this. That way "bloats" the HTML, this way means you need to keep HTML and CSS in sync. Lea Verou has a clever way to handle with text-shadow.

\*/

td:nth-of-type(1):before {

content: "First Name";

}

td:nth-of-type(2):before {

content: "Last Name";

}

td:nth-of-type(3):before {

content: "Job Title";

}

}

</style>

# /\*\*\*\*\*\*\*\*\*#######\*\*\*\* 1.2CSS ATTRIBUTE SELECTORS \*\*\*\*#####\*\*\*/

[**https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-child**](https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-child)

[**http://nthmaster.com/**](http://nthmaster.com/)

/\*

ATTRIBUTE SELECTORS

Three new kinds of attribute selectors are a part of CSS3 specs:

[att^="value"] Matches elements to an attribute that starts with the specified value.

[att$="value"] Matches elements to an attribute that ends with the specified value.

[att\*="value"] Matches elements to an attribute that contains the specified value.

\*/

a[title$="tweetCC"] {

position: absolute;

top: 0;

right: 0;

display: block;

width: 140px;

height: 140px;

text-indent: -9999px;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*http s://www.smashingmagazine.com/2016/05/an-ultimate-guide-to-css-pseudo-classes-and-pseudo-elements/\*/

/\*===================== first(3)=====================\*/

ul li:nth-child(-n + 3) {

background: red;}

/\*===================== last(3)=====================\*/

ul li:nth-last-child(-n + 3) {

background: red;}

/\*===================== after-first(5) ==============\*/

ul li:nth-child(n+6) {

background: red;}

/\*===================== from-end(3) ===============\*/

ul li:nth-last-child(3) {

background: red;

}

/\*===================== between(3, 6) ==============\*/

ul li:nth-child(n+3):nth-child(-n+6) {

background: red;}

/\*================== even-between(3, 12) ===============\*/

ul li:nth-child(even):nth-child(n + 3):nth-child(-n + 12) {

background: red;}

/\*===================== odd-between(3, 13) ===============\*/

ul li:nth-child(odd):nth-child(n + 3):nth-child(-n + 13) {

background: red;}

/\*================ n-between(3, 10, 20)=================\*/

ul li:nth-child(3n):nth-child(n + 10):nth-child(-n + 20) {

background: red;}

/\*===================== all-but(3)=================\*/

ul li:not(:nth-child(3)) {

background: red;}

/\*===================== each(3)=====================\*/

ul li:nth-child(3n) {

background: red;}

/\*===================== every(3)=====================\*/

ul li:nth-child(3n) {

background: red;}

/\*===================== from-first-last(2) ===============\*/

ul li:nth-child(2), ul li:nth-last-child(2) {

background: red;}

/\*===================== middle(11) =============\*/

ul li:nth-child(6) {

background: red;}

/\*===================== all-but-first-last(2) ============\*/

ul li:nth-child(n+2):nth-last-child(n+2) {

background: red;}

/\*===================== first-of(10) ===============\*/

ul li:nth-last-child(10):first-child {

background: red;}

/\*===================== last-of(10) ==================\*/

ul li:nth-of-type(10):nth-last-of-type(1) {

background: red;}

/\*===================== at-least(10) ==================\*/

ul li:nth-last-child(n + 10), ul li:nth-last-child(n + 10) ~ li {

background: red;}

/\*===================== at-most(10) =====================\*/

ul li:nth-last-child(-n + 10):first-child, ul li:nth-last-child(-n + 10):first-child ~ li {

background: red;}

/\*===================== in-between(5, 10) ================\*/

ul li:nth-last-child(n + 5):nth-last-child(-n + 10):first-child, ul li:nth-last-child(n + 5):nth-last-child(-n + 10):first-child ~ li {

background: red;}

/\*===================== even()=====================\*/

ul li:nth-child(even) {

background: red;}

/\*===================== odd()=====================\*/

ul li:nth-child(odd) {

background: red;}

/\*===================== first-last() ================\*/

ul li:first-child, ul li:last-child {

background: red;}

/\*===================== unique()=====================\*/

ul li:only-child {

background: red;}

/\*===================== not-unique()=====================\*/

ul li:not(:only-child) {

background: red;}

/\*===================== child-index(10) ==================\*/

ul li:nth-child(1) {

z-index: 1;}

ul li:nth-child(2) {

z-index: 2;}

……...

……..

ul li:nth-child(10) {

z-index: 10;}

/\*===================== child-index(5, backward, 10)=====================\*/

ul li:nth-last-child(1) {

z-index: 9;}

ul li:nth-last-child(2) {

z-index: 8;}

ul li:nth-last-child(3) {

z-index: 7;}

ul li:nth-last-child(4) {

z-index: 6;}

ul li:nth-last-child(5) {

z-index: 5;}

**If root:16px**

Body (default) -

H1 - 35px - 2.1875rem

H2 - 28px - 1.75rem

H3 - 25px - 1.5625rem

H4 - 21px - 1.3125rem

H5 - 17px - 1.0625rem

H6 - 14px - 0.875rem

P -

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

html { font-size: 1rem;}

@include media-breakpoint-up(sm) {

html { font-size: 1.2rem; }

}

@include media-breakpoint-up(md) {

html { font-size: 1.4rem; }

}

@include media-breakpoint-up(lg) {

html { font-size: 1.6rem; }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* For mobile phones: \*/  
[class\*="col-"] {  
  width: 100%;  
}  
  
@media only screen and (min-width: 600px) {  
  /\* For tablets: \*/  
  .col-s-1 {width: 8.33%;}  
  .col-s-2 {width: 16.66%;}  
  .col-s-3 {width: 25%;}  
  .col-s-4 {width: 33.33%;}  
  .col-s-5 {width: 41.66%;}  
  .col-s-6 {width: 50%;}  
  .col-s-7 {width: 58.33%;}  
  .col-s-8 {width: 66.66%;}  
  .col-s-9 {width: 75%;}  
  .col-s-10 {width: 83.33%;}  
  .col-s-11 {width: 91.66%;}  
  .col-s-12 {width: 100%;}  
}  
  
@media only screen and (min-width: 768px) {  
  /\* For desktop: \*/  
  .col-1 {width: 8.33%;}  
  .col-2 {width: 16.66%;}  
  .col-3 {width: 25%;}  
  .col-4 {width: 33.33%;}  
  .col-5 {width: 41.66%;}  
  .col-6 {width: 50%;}  
  .col-7 {width: 58.33%;}  
  .col-8 {width: 66.66%;}  
  .col-9 {width: 75%;}  
  .col-10 {width: 83.33%;}  
  .col-11 {width: 91.66%;}  
  .col-12 {width: 100%;}  
}

Example:

<div class="row">  
  <div class="col-3 col-s-3">...</div>  
  <div class="col-6 col-s-9">...</div>  
  <div class="col-3 col-s-12">...</div>  
</div>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Extra small devices (phones, 600px and down) \*/  
@media only screen and (max-width: 600px) {...}   
  
/\* Small devices (portrait tablets and large phones, 600px and up) \*/  
@media only screen and (min-width: 600px) {...}   
  
/\* Medium devices (landscape tablets, 768px and up) \*/  
@media only screen and (min-width: 768px) {...}   
  
/\* Large devices (laptops/desktops, 992px and up) \*/  
@media only screen and (min-width: 992px) {...}   
  
/\* Extra large devices (large laptops and desktops, 1200px and up) \*/  
@media only screen and (min-width: 1200px) {...}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

An **at-rule** is a [CSS statement](https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax#CSS_statements) that instructs CSS how to behave. They begin with an at sign, '@' (U+0040 COMMERCIAL AT), followed by an identifier and includes everything up to the next semicolon, ';' (U+003B SEMICOLON), or the next [CSS block](https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax#CSS_declarations_blocks), whichever comes first.

/\* General structure \*/

@IDENTIFIER (RULE);

/\* Example: tells browser to use UTF-8 character set \*/

@charset "utf-8";

There are several at-rules, designated by their identifiers, each with a different syntax:

* [@charset](https://developer.mozilla.org/en-US/docs/Web/CSS/@charset) — Defines the character set used by the style sheet.
* [@import](https://developer.mozilla.org/en-US/docs/Web/CSS/@import) — Tells the CSS engine to include an external style sheet.
* [@namespace](https://developer.mozilla.org/en-US/docs/Web/CSS/@namespace) — Tells the CSS engine that all its content must be considered prefixed with an XML namespace.
* ***Nested at-rules*** — A subset of nested statements, which can be used as a statement of a style sheet as well as inside of conditional group rules:
  + [@media](https://developer.mozilla.org/en-US/docs/Web/CSS/@media) — A conditional group rule that will apply its content if the device meets the criteria of the condition defined using a *media query*.
  + [@supports](https://developer.mozilla.org/en-US/docs/Web/CSS/@supports) — A conditional group rule that will apply its content if the browser meets the criteria of the given condition.
  + [@document](https://developer.mozilla.org/en-US/docs/Web/CSS/@document)  — A conditional group rule that will apply its content if the document in which the style sheet is applied meets the criteria of the given condition. *(deferred to Level 4 of CSS Spec)*
  + [@page](https://developer.mozilla.org/en-US/docs/Web/CSS/@page) — Describes the aspect of layout changes that will be applied when printing the document.
  + [@font-face](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face) — Describes the aspect of an external font to be downloaded.
  + [@keyframes](https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes) — Describes the aspect of intermediate steps in a CSS animation sequence.
  + [@viewport](https://developer.mozilla.org/en-US/docs/Web/CSS/@viewport)  — Describes the aspects of the viewport for small screen devices. *(currently at the Working Draft stage)*
  + [@counter-style](https://developer.mozilla.org/en-US/docs/Web/CSS/@counter-style) — Defines specific counter styles that are not part of the predefined set of styles. *(at the Candidate Recommendation stage, but only implemented in Gecko as of writing)*
  + [@font-feature-values](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-feature-values) (plus @swash, @ornaments, @annotation, @stylistic, @styleset and @character-variant)  
    — Define common names in [font-variant-alternates](https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant-alternates) for feature activated differently in OpenType. *(at the Candidate Recommendation stage, but only implemented in Gecko as of writing)*

CSS Flexbox model is optimised for UI Design. It is developed primarily to avoid overflowing parent element. It will shrink children when ancestor element size is constricted. It will fill the space by expanding child element's size when ancestor element size extends. Flex container shrink and expand behaviour can break with min and max width / height property.

**CSS FlexBox versions by version**

W3 2009 : display: box;

box-align start | end | center | baseline | stretch

box-direction normal | reverse | inherit

box-flex <number> 0.0

box-flex-group <integer> 1

box-lines single | multiple

box-ordinal-group <integer> 1 visual

box-orient horizontal | vertical | inline-axis | block-axis | inherit inline-axis box elements no no visual

box-pack start | end | center | justify

W3 2011 : display flexbox | inline-flexbox

flex-align auto | baseline auto

flex-direction lr | rl | tb | bt | inline | inline-reverse | block | block-reverse flexboxes no lr | rl | tb | bt

flex-order <integer> 1

flex-pack start | end | center | justify

W3 2012 : display flex | inline-flex

align-content flex-start | flex-end | center | space-between | space-around | stretch

align-items flex-start | flex-end | center | baseline | stretch

align-self auto | flex-start | flex-end | center | baseline | stretch

flex-direction row | row-reverse | column | column-reverse

flex-flow <'flex-direction'> || <'flex-wrap'>

flex-grow <number> ‘0’

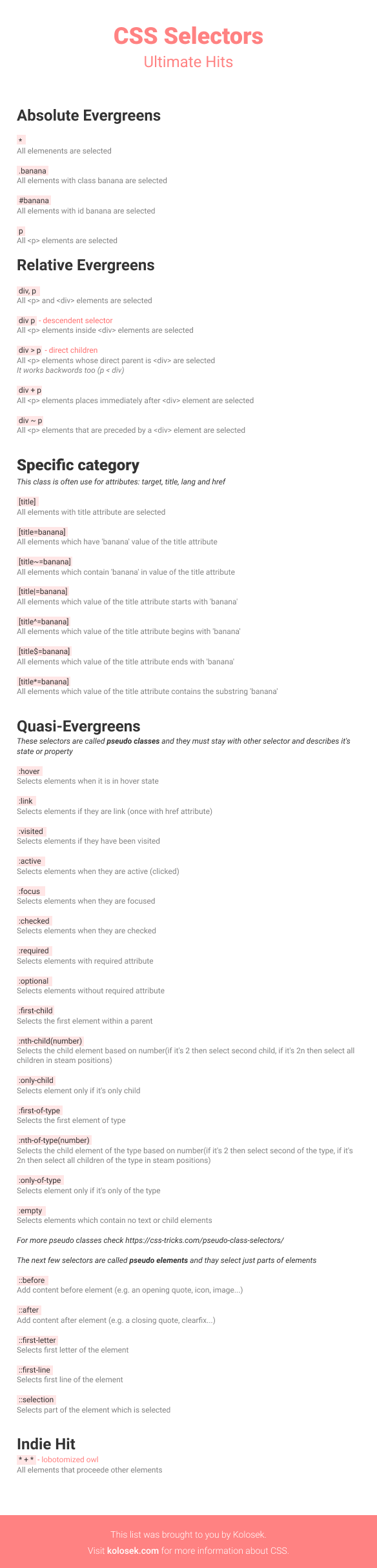
flex-shrink <number> ‘1’

flex-wrap nowrap | wrap | wrap-reverse

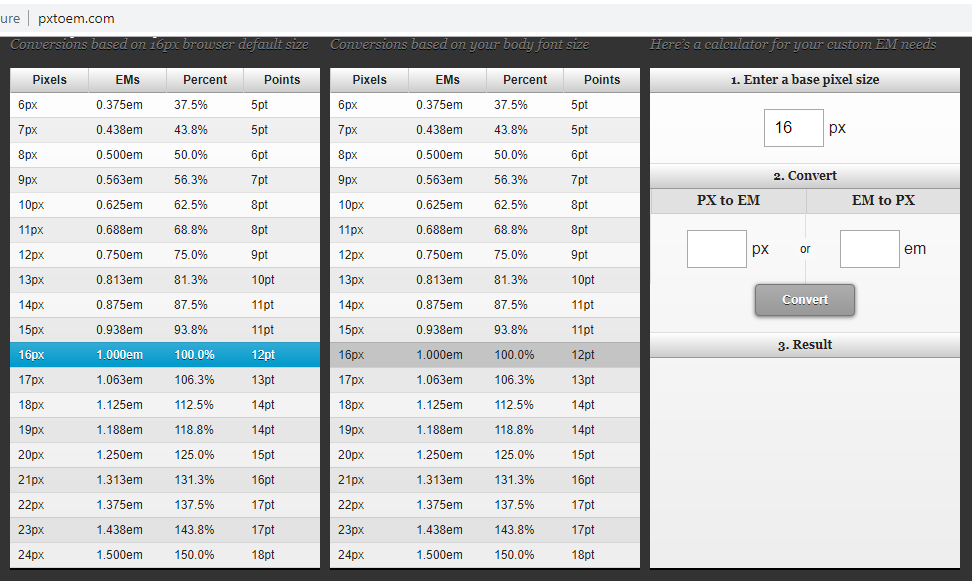
justify-content flex-start | flex-end | center | space-between | space-around

order <number> 0

<https://kolosek.com/css-selectors/>



<https://simplecss.eu/pxtoems.html>



/\*\*\*\*\*font size in percentage\*\*\*\*\*\*/

<https://developer.mozilla.org/en-US/docs/Web/CSS/font-size>

/\* font-size 1em = 10px on default browser settings \*/

/\* Browser default considered as: root {font-size:16px} \*/

font-size: 62.5%;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.parent .column\_leftcontent {

width: 50%;

padding-right: 4%;

margin: 0;

}

.parent .column\_rightimg {

width: 50vw !important;

position: relative;

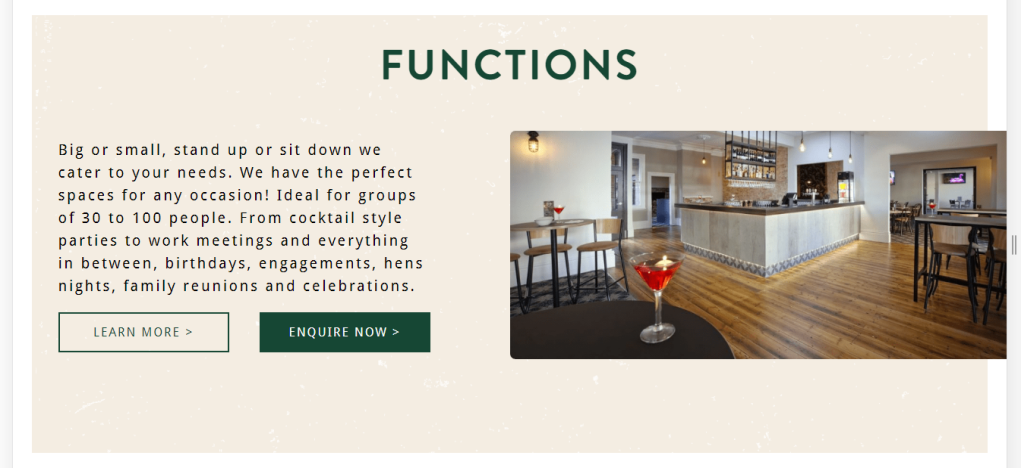
left: 50%;

right: 50%;

margin-left: -52%;

margin-right: -50vw;

}



/\*\*\*\*Content CSS \*\*\*\*\*/

.btnmenu\_tel {

background-image: url('/2019/06/phone-icon-white.png');

background-repeat: no-repeat;

background-position: left;

}

.fusion-main-menu li:nth-last-child(2):after {

content: '';

position: absolute;

height: 115px;

width: 1px;

background: #fff;

top: 0;

right: 0;

z-index: 99;

overflow: hidden;

}

.btn\_services .fusion-column-wrapper div:nth-of-type(3n+2) a,

#bttn\_attract .fusion-button:hover .fusion-button-text,

.contact-frm input[type="text"],

.wpcf7-form input[type="submit"] {

font-size: 20px;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Min-width: 320px (smaller phone viewpoints)

Min-width: 480px (small devices and most phones)

Min-width: 768px (most tablets)

Min-width: 992px (smaller desktop viewpoints)

Min-width: 1200px (large devices and wide screens)

Body{

Font-size:20px;

Line-height:10px

Color:#000;

Weight:normal;

}

h1{

Font-size:60

Line-height:20px

Color:#fff;

Weight:600

}

h2{

Font-size:45

Line-height:15px

Color:#fff;

Weight:600

}

h3{

Font-size:35

Line-height:15px

Color:#000;

Weight:normal

}

h4{

Font-size:30

Line-height:10px

Color:#000;

Weight:normal

}

h5{

Font-size:25

Line-height:10px

Color:#000;

Weight:normal

}

h6{

Font-size:20

Line-height:20px

Color:#000;

Weight:normal

}

paragraph{

Font-size:20px;

Line-height:10px

Color:#000;

Weight:normal;

}

#### Relative Viewport Lengths

* vw - Viewports width
* vh - Viewports height
* vmin - Minimum of the viewport’s height and width
* vmax - Maximum of the viewport’s height and width

#### Omitting a Media Type

When using the not and only logical operators the media type may be left off. In this case the media type is defaulted to all.

### Media Features in Media Queries

#### Height & Width Media Features

@media all and (min-width: 320px) and (max-width: 780px) {...}

#### Orientation Media Feature

@media all and (orientation: landscape) {...}

#### Aspect Ratio Media Features

@media all and (min-device-aspect-ratio: 16/9) {...}

#### Pixel Ratio Media Features

@media only screen and (-webkit-min-device-pixel-ratio: 1.3), only screen and (min-device-pixel-ratio: 1.3) {...}

#### Resolution Media Feature

@media print and (min-resolution: 300dpi) {...}

@media (min-resolution: 300dpi) {

/\* \*/

}

* **Nokia Lumia 640**: 332dpi
* **Apple iPhone 6+**: 401dpi
* **Google Nexus 5**: 445dpi
* **HTC One**: 469dpi
* **Samsung Galaxy S6**: 577dpi

#### Resolution

#### Viewport Resolution

<meta name="viewport" content="target-densitydpi=device-dpi">

#### Combining Viewport Values

<meta name="viewport" content="width=device-width, initial-scale=1">

$('.feature'); // Class selector

$('li strong'); // Descendant selector

$('em, i'); // Multiple selector

$('a[target="\_blank"]'); // Attribute selector

$('p:nth-child(2)'); // Pseudo-class selector

**properties of flexbox**

* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

<https://learn.shayhowe.com/advanced-html-css/jquery/>

<https://www.geeksforgeeks.org/css-media-rule/>

<https://www.tutorialbrain.com/css_tutorial/css_media_queries/>

[https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)

@media (min-width:320px) { /\* smartphones, iPhone, portrait 480x320 phones \*/ }

@media (min-width:481px) { /\* portrait e-readers (Nook/Kindle), smaller tablets @ 600 or @ 640 wide. \*/ }

@media (min-width:641px) { /\* portrait tablets, portrait iPad, landscape e-readers, landscape 800x480 or 854x480 phones \*/ }

@media (min-width:961px) { /\* tablet, landscape iPad, lo-res laptops ands desktops \*/ }

@media (min-width:1025px) { /\* big landscape tablets, laptops, and desktops \*/ }

@media (min-width:1281px) { /\* hi-res laptops and desktops \*/ }

/\*CSS with variables\*/

:root {

--main-bg-color: brown;

}

.one {

color: white;

background-color: var(--main-bg-color);

}

.two {

color: var(--my-var, red); /\* Red if --my-var is not defined \*/

}

.three {

background-color: var(--my-var, var(--my-background, pink)); /\* pink if my-var and --my-background are not defined \*/

}

// Get or set CSS variables using Javascript

// get variable from inline style

element.style.getPropertyValue("--my-var");

// get variable from wherever

getComputedStyle(element).getPropertyValue("--my-var");

// set variable on inline style

element.style.setProperty("--my-var", jsVar + 4);

**FREQUENTLY using CSS**

/\*To hide wonderplugin watermark:\*/

.wonderplugin-gridgallery-list .wonderplugin-gridgallery-item > a, #html5-watermark {

display: none !important;

}

/\*To hide wonder carousel plugin watermark:\*/

.amazingcarousel-list .amazingcarousel-item .amazingcarousel-image a ~ div a {

display: none !important;

visibility: hidden !important;

}

/\*Form input box FOCUS issue:\*/

.wpcf7-form-control-wrap {

position: unset;

}

/\*WP Add Mime Types Admin Settings\*/

otf = font/otf

sfnt = font/sfnt

ttf = font/ttf

woff = font/woff

woff2 = font/woff2

/\*Image from right edge\*/

@media only screen and (min-width: 1000px) {

.sec\_function .secfunc\_rightimg {

width: 50vw !important;

position: relative;

left: 50%;

right: 50%;

margin-left: -52%;

margin-right: -50vw;

}

.sec\_function .secfunc\_leftcontent {

width: 50%;

padding-right: 4%;

margin: 0;

}

}

/\*csdgrampians\*/

.tbllinkhover a {

overflow-wrap: break-word;

word-break: break-all;

}