**CSS**

#BB43BE

**Selectors**

Selectors: Patterns used to select and style elements in a CSS rule.

.class: Selects elements with a specific class attribute.

.highlight {

  background-color: yellow;

  color: black;

}

#id: Selects a single element with a specific ID attribute.

#header {

  font-size: 24px;

  color: blue;

}

element: Selects elements based on their tag name.

p {

  font-style: italic;

}

\*: Selects all elements on a page.

\* {

  margin: 0;

  padding: 0;

}

**Colors**

Text Color:

color:  blue;

Background Color:

background-color: lightblue;

Opacity Color:

opacity: 0.9;

Border Color:

border-color: red;

**Units and Sizes**

Units and Sizes: Specify measurements for various CSS properties.

px: Pixels

em: Ems

rem: Rems

%: Percentages

vw, vh, vmin, vmax: Viewport Units

in, cm, mm, pt, pc: Absolute Units

fr: fraction

auto: Auto

calc(): Calculation

**Box Models**

Box Models: Define the layout and spacing of elements.

width, height: Width and Height.

padding: Padding

border: Border

margin: Margin

box-sizing: control the weight, height, border and padding of an element.

outline: Outline

display: Display

align-items: align items

justify-content: vertical align items.

.centered-box {

  width: 300px;

  height: 200px;

  margin: 0 auto;

  border: 2px solid black;

  padding: 20px;

box-sizing: content-box/border-box/

  display: flex;

  align-items: center/flex-star/flex-end/;

  justify-content: center/flex-star/flex-end/;

}

**Typography**

Typography: Control the appearance of text on your webpage.

font-family: type of font.

font-size: size of the font.

font-weight: weight of the font.

font-style: Font-Style

line-height: height separation

text-align: ubication

color: Color

text-decoration: decoration of text.

letter-spacing: space between letters.

word-spacing: space between words.

p {

  font-family: "Arial", sans-serif/cursive/fantasy;

  font-style: italic/oblique/normal;

  font-size: 24px;

  font-weight: bold/lighter/100 to 900/bolder/normal;

  text-align: center/left/right/justify;

  text-indent: 2em;

  color: #333;

  text-decoration: underline/none/;

  text-transform: capitalize/lowercase/uppercase;

  letter-spacing: 1px/0.1;

  line-height: 1.5;

  word-spacing: 0.5;

}

**Styling Links**

a: anchor tag.

a:visited: for already visited links.

a:hover: when the mousse is on the link.

a:active: when the mousse clicked the link and try to move it to another place.

a:focus: when navigates to an interactive element using the keyboard.

a {

  text-decoration: underline/none;

  cursor: pointer/not-allowed;

  color: blue;

}

a:visited {

  color: purple;

}

a:hover , a:focus{

  color: dodgerblue;

  background-color: gold;

}

a:active{

  color: red;

}

**Lists Styles**

HTML’s List Tags:

ol: order lists. (numeric list)

ul: unorder lists. (circle list)

li: list item.

CSS Styles:

list-style-type: to change the type of markers.

text-align: position of the list without markers.

list-style-position: position of the list and markers.

line-height: height distances between lists.

list-style-image: change de bullets adding an image.

CSS Element:

content: you can swap the list-style-type for markers whit a text.

CSS Pseudo-Class:

ul li:nth-child(2): break down the unorder lists, item list number 2 or even or odd.

CSS Pseudo-Element:

::marker: for set de markers

ol {

  list-style-type: lower-alpha/decimal/circle/upper-roman/none;

}

ul {

list-style-type: square;

text-align: center;

list-style-position: inside/outside;

color: blue;

line-height: 1.6;

list-style-image: url(“../images/checkmark.png”);

}

ul li:nth-child(2) {

color: red;

}

::marker {

color: red;

font-family: fantasy;

font-size: 1em;

content: “Only $4 >>”;

**Displays**

display: Type of presentation

display: block; Arrange the blocks one below the other respecting the height and width.

display: inline; Align the blocks one below the other without respecting height and width.

display: inline-block; Aling the blocks one below the other respecting the height and width.

display: flow-root; Creates a block that does not interfere with other design elements.

display: flex;

display: grid;

**Columns**

. columns {

column-count: 4; Number of columns.

column-width: 250px; width (max 250px)

columns: 4 250px; (amount and width)

column-rule: 3px solid black; divider column.

column-count: 4;

column-width: 250px (max 250px)

or

columns: 4 250px;

column-rule: 3px solid black;

}

. columns p {

margin-top: 0;

}

. columns h2 {

padding: 1rem;

margin-top: 0;

background-color: black;

color: white;

break-inside: avoid; prevent an element from being split into multiple parts during a page or column break in multi-column or multi-page layouts.

}

. columns .quote {

margin-top: 2rem;

font-size: 3rem;

text-align: center;

color: black;

column-span: all; Allows an element to span all available columns, ignoring column boundaries and creating a "collapsed" effect on the column layout.

}

. nowrap {

white-space: nowrap; Used to prevent line wrapping and collapsing whitespace within the element.

}

**Position**

position: static; default position, don’t need directions.

position: absolute; is positioned relative to the nearest positioned ancestor.

position: relative; is the normal position, set the directions to adjust the position.

position: fixed; it always stays in the same place even if the page is scrolled.

position: sticky; toggles between relative and fixed.

html { scroll-behavior: smooth; Scroll when you click in a: <a href="#one">One</a>

}

div.static {

position: static;

border: 3px solid #73AD21;

}

div.relative {

position: relative;

left: 30px;

border: 3px solid #73AD21;

}

div.fixed {

position: fixed;

bottom: 0;

right: 0;

width: 300px;

border: 3px solid #73AD21;

}

div.sticky {

position: -webkit-sticky; /\* Safari \*/

position: sticky;

top: 0;

background-color: green;

border: 2px solid #4CAF50;

}

Absolute position:

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;

z-index: 1; The absolute box will be on the fixed box.

}

**Flexbox**

display: flex; to set flexboxes (Block horizontal from default).

gap: 10px; gap between the boxes.

flex-direction: row (from default) /row-reverse /column/column-reverse; to choose column or rows position.

flex-wrap: wrap (from default) /wrap/nowrap; flex boxes inside a div.

or

flex-flow: row wrap; It’s the mix of flex-direction and flex-wrap.

justify-content: stretch (from default) fills the horizontal items

justify-content: start/center/end; choose an horizontal side of the boxes.

or for a distribution:

justify-content: space-around/space-between/space-evenly; align horizontal distribution.

align-items: stretch (from default, fills the vertical items)

or for vertical distribution:

align-items: flex-end/flex-start/center: Choose a vertical side of the boxes, start to the top, center to center and end for bottom.

or

align-items: baseline; if the divs have different sizes it aligns them from baseline.

align-content: start/center/end/; if the height is taller, choose a vertical side.

or for a vertical distribution:

align-content: space-between/space-around/space-evenly; align vertical distribution.

|  |  |
| --- | --- |
| **Property** | **Description** |
| [align-content](https://www.w3schools.com/cssref/css3_pr_align-content.asp) | Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines |
| [align-items](https://www.w3schools.com/cssref/css3_pr_align-items.asp) | Vertically aligns the flex items when the items do not use all available space on the cross-axis |
| [display](https://www.w3schools.com/cssref/pr_class_display.asp) | Specifies the type of box used for an HTML element |
| [flex-direction](https://www.w3schools.com/cssref/css3_pr_flex-direction.asp) | Specifies the direction of the flexible items inside a flex container |
| [flex-flow](https://www.w3schools.com/cssref/css3_pr_flex-flow.asp) | A shorthand property for flex-direction and flex-wrap |
| [flex-wrap](https://www.w3schools.com/cssref/css3_pr_flex-wrap.asp) | Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line |
| [justify-content](https://www.w3schools.com/cssref/css3_pr_justify-content.asp) | Horizontally aligns the flex items when the items do not use all available space on the main-axis |

Child elements (Items)

order: Specify the order of the boxes in the CSS styles or it HTML styles.

div: nth-child(1) {

order: 3;

}

flex-grow: Specify how much a flex item will grow relative to the rest of flex items.

flex-grow: 1; to set all the flex items in the same size (1 is the default value).

div:nth-child(1) {

flex-grow: 2; to set the only the firs flex item a bigger size.

}

flex-shrink: specifies how much a flex item will shrink relative to the rest of the flex items.

shrink == reduces. flex-shrink: 1; is the default value.

div:nth-child(1) {

flex-shrink: 2; to set the only the firs flex item a shrinker size than the others.

}

**Grid Layout**

position: grid /inline-grid; To set grids.

Grid Rows and Columns

grid-template-columns: auto auto auto/100px 100px 100px; assign number of columns and the width.

grid-template-rows: auto auto auto/100px 100px 100px; assign number of rows and the height.

Gaps

gap: 10px;/ 10px 20px gap between the columns and rows.

columns-gaps: 10px; gap between columns.

rows-gaps: 10px; gap between rows.

Lines

.item1 {

grid-column-start: 1; grid number 1 starts in column line 1.

grid-column-end: 3; grid number 1 ends in column line 3.

}

.item3 {

grid-row-start: 1; grid number three starts in row line 1.

grid-row-end: 3; grid number three ends in row line 3.

}

Positions

justify-content: space-around/space-between/space-evenly; horizontal position.

justify-content: start/center/end; horizontal side.

align-content: space-around/space-between/space-evenly; vertical position.

align-content: start/center/end; horizontal side.

Items

grid-column: 1 / 5; Make "item1" start on column 1 and end before column 5.

grid-column: 1 / span 3; Make "item1" start on column 1 and span 3 columns.

grid-row: 1 / 4; defines on which row the item will starts and ends.4

grid-row: 1 / span 2 ; defines on which row the item will starts and ends.

grid-area: Is a shorthand of grid-row-start, grid-row-end, grid-column-start and grid-column-end.

grid-area: 1 / 2 / 5 / 6; Start on row-line 1 and column-line 2, and end on row-line 5 and column line 6.

grid-area: 2 / 1 / span 2 / span 3; Start on row-line 2 and column-line 1, and span 2 rows and 3.

Naming grid items

Item1 gets the name "myArea" and spans all five columns in a five columns grid layout:

.item1 {

grid-area: myArea;

}

.grid-container {

grid-template-areas: 'myArea myArea myArea myArea myArea';

}

pseudo selectors:

.box:first-child {

background-color: blue;

grid-column-start: 1;

grid-column-end: 4;

grid-row-start; 1;

grid-row-end: 3;

or

grid-columns: 1 / 4;

grid-row: 1 / 3;

display: grid;

place-content: center;

}

**Images**

background-image: url(‘../img/photo1.png’); for background image.

background-repeat: repeat/no-repeat; repeat a background image.

background-size: cover; focus the background image.

background-color: hsla(0, 0%, 100%, 0.403); lighten background color.

background-position: left/right/top/bottom/center/left top/right center/; background position.

background-image: linear-gradient (to left, green, white); The background color will be green on the left side and will turn white on the right.

background-image: url(‘../img/phoyo1.png’), linear-gradient (to left, green, white); The transparent background image will be an image whit background color green on the left side and will turn white on the right.

object-fit: cover: ensures the image fits in the size box without stretching.

object-position: center center: ensures that the center of the image is shown in the box.

.h1 {

text-shadow: 2px 2px 5px black; for add shadow effect to the text.

}

**Medias Query**

Sizes:

Common Media Query breakpoints:

| Breakpoint | Description |

| -------- | ---------- |

| < 481px | Mobile devices |

| 481px — 768px | iPads, Tablets |

| 769px — 1024px | Small screens, laptops |

| 1025px — 1200px | Desktops, large screens |

| 1201px and greater | Extra large screens, TV |

Bootstrap breakpoints:

| Breakpoint | Description |

| -------- | ---------- |

| < 576px | xs |

| >=576px | small |

| >=768px | medium |

| >=992px | large |

| >=1200px | xl |

| >=1400px | 2xl |

Tailwind breakpoints:

| Breakpoint | Description |

| < 640px | xs |

| >=640px | small |

| >=768px | medium |

| >=1024px | large |

| >=1280px | xl |

| >=1536px | 2xl |

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

body {

background-color: gold;

background-image: radial-gradient(whitesmoke, gold)

}

}

**Cards Projects**

.card {

scroll-margin-top: 8rem;

width: min(100%, 350px);

background-color: #cbd5e1;

border: 2px solid #000;

border-radius: 15px;

padding: 1rem;

display: flex;

flex-direction: column;

align-items: center;

}

.card figure {

display: flex;

flex-flow: column nowrap;

}

.card img {

border: 5px double #333;

border-radius: 50%;

}

.card figcaption {

font-weight: bolder;

font-size: 2rem;

margin: 1rem;

text-align: center;

}

**Useful Keyboard & Shortcuts**

!+Enter: Create an HTML document.

Ctrl+U: View the HTML code in a web site.

Ctrl+B: Hide the VS Code Explorator.

Ctrl+Shift+I or Windows+F12: Element inspection in a web site.

Ctrl+S: Save changes in VS Code.

Alt+Shift+downArrow: Copy and paste selected code.

Ctrl+D: Select already highlighted code lines.

Shift+Alt+A: Comment the selected code.

Ctrl+Shift+M: Open Inspection to check the medias querys.

p\*5<lorem20: Create five paragraphs with Lorem Ipse Text.