CSS

[CSS INTRODUCTION 3](#_Toc94119663)

[USEFUL TOOLS AND PLATFORMS 3](#_Toc94119664)

[ADD CSS TO AN HTML PAGE 4](#_Toc94119665)

[CSS SELECTORS 4](#_Toc94119666)

[PRIMARY SELECTORS 5](#_Toc94119667)

[COMBINATORS (NESTED SELECTORS) 5](#_Toc94119668)

[PSEUDO-CLASSES AND PSEUDO-ELEMENTS 5](#_Toc94119669)

[CSS UNITS AND MATH FUNCTIONS 4](#_Toc94119670)

[CSS SHORTHAND PROPERTIES 5](#_Toc94119671)

[COMMON CSS PROPERTIES 6](#_Toc94119672)

[BACKGROUND 6](#_Toc94119673)

[SOLID COLOR AND GRADIENT 6](#_Toc94119674)

[IMAGE BACKGROUND 7](#_Toc94119675)

[BACKGROUND STARTING POINT 8](#_Toc94119676)

[BORDER 9](#_Toc94119677)

[DRAW BORDER 10](#_Toc94119678)

[USE IMAGE 10](#_Toc94119679)

[ROUNDED CORNERS 10](#_Toc94119680)

[TABLE BORDER 10](#_Toc94119681)

[SHADOW 9](#_Toc94119682)

[LIST STYLING 9](#_Toc94119683)

[RESIZING 9](#_Toc94119684)

[CURSOR 9](#_Toc94119685)

[CSS COLORS 9](#_Toc94119686)

[COLORABLE PARTS OF AN ELEMENT 9](#_Toc94119687)

[SOLID COLORS 10](#_Toc94119688)

[COLORS & OPACITY 11](#_Toc94119689)

[CURRENT COLOR 12](#_Toc94119690)

[CSS GRADIENTS 12](#_Toc94119691)

[LINEAR GRADIENTS 12](#_Toc94119692)

[RADIAL GRADIENTS 12](#_Toc94119693)

[CONIC GRADIENTS 13](#_Toc94119694)

[TEXT FORMATTING 13](#_Toc94119695)

[FONT FORMATTING 13](#_Toc94119696)

[GENERAL FONT FORMATTING 13](#_Toc94119697)

[TEXT COLOR AND SHADOW 13](#_Toc94119698)

[FONT PAIRINGS 13](#_Toc94119699)

[GOOGLE FONTS 14](#_Toc94119700)

[FONT AWSOME ICONS 14](#_Toc94119701)

[TEXT LAYOUT 14](#_Toc94119702)

[TEXT ALIGNMENT 14](#_Toc94119703)

[LINE LAYOUT & WORD BREAKING 14](#_Toc94119704)

[TEXT DIRECTION 14](#_Toc94119705)

[CSS IMAGES 14](#_Toc94119706)

[TRANSITIONS 14](#_Toc94119707)

[ANIMATIONS 14](#_Toc94119708)

[CSS COUNTERS 14](#_Toc94119709)

[CSS WEBSITE LAYOUT 14](#_Toc94119710)

[BOX MODEL 15](#_Toc94119711)

[POSITION & FLOAT 15](#_Toc94119712)

[FLEFBOX 15](#_Toc94119713)

[ABOUT FLEXBOX 15](#_Toc94119714)

[PROPERTIES FOR THE PARENT 15](#_Toc94119715)

[PROPERTIES FOR THE CHILDREN 15](#_Toc94119716)

[CSS LAYOUT EXAMPLES 15](#_Toc94119717)

[HORIZONTAL ALIGNMENT 15](#_Toc94119718)

[VERTICAL ALIGNMENT 15](#_Toc94119719)

[HORIZONTAL AND VERTICAL ALIGNMENT 15](#_Toc94119720)

[2D TRANSFORMATIONS 15](#_Toc94119721)

[3D TRANSFORMATIONS 15](#_Toc94119722)

[DISPLAYING IMAGES 15](#_Toc94119723)

[DROPDOWN MENUS 15](#_Toc94119724)

[RESPONSIVE WEB DESIGN 15](#_Toc94119725)

[MEDIA QUERIES 15](#_Toc94119726)

[CSS ARCHITECTURE 15](#_Toc94119727)

[ITCSS LAYERS 15](#_Toc94119728)

[BEM NAMING CONVENTION 15](#_Toc94119729)

[OBJECT ORIENTED CSS 15](#_Toc94119730)

[CSS MODULES 15](#_Toc94119731)

[CSS STYLED COMPONENTS 15](#_Toc94119732)

# CSS INTRODUCTION

|  |  |
| --- | --- |
| About CSS | CSS stands for Cascading Style Sheets and defines the visual presentation of HTML elements. Every CSS document is a collection of CSS rules. Every rule contains a selector (an identifier of the HTML element/s to be styled) and one or more declarations separated by semicolon. "Cascading" means that every rule for an HTML element will also be applied to all its child elements. |
| CSS Syntax | selector {  property: value; } |

# USEFUL TOOLS AND PLATFORMS

|  |  |
| --- | --- |
| View CSS in the Browser | CSS Viewer extension for Chrome |
| Check Actual CSS Properties of an Element in the Browser | right click > Inspect > Computed |
| Online Code Editor for HTML, CSS and JS | codepen.io |
| Make a Screenshot of an Entire Web Page | GoFullPage extension for Chrome |
| Create Design | Figma |
| Pick Exact Color | Just Color Picker |
| CSS Gradient Generator | cssgradient.io |
| Check Browser Support for a Specific Selector/Element | caniuse.com |
| CSS Selector Game | https://flukeout.github.io/ |
| Design Guidelines | Material Design (material.io) |
| Get a Random Image | Lorem Picsum (picsum.photos) |
| Book on Writing Good Code | *Code Complete (2)* by Steve McConnell |
| CSS Tricks Articles | css-tricks.com |

# ADD CSS TO AN HTML PAGE

|  |  |
| --- | --- |
| External Style Sheet (in <head>) | <link rel="stylesheet" type="text/css" href="site.css"> <!-- recommended --> |
| Internal Style Sheet  (in <head>) | <style>...</style> <!-- when a single document has a unique style --> |
| Inline Style | <p *style*="color:red">Some red text</p> |
| Style Inheritance | When the same element is styled differently, the last style overwrites the previous ones. In increasing order of priority: external styles; internal styles (if declared after the external ones); inline styles; **!important**. |
| Selector Specificity | In increasing order of specificity: elements and pseudo-elements; classes, pseudo-classes and attribute selectors; IDs; **!important**. |

# CSS UNITS, NUMBERS AND MATH FUNCTIONS

|  |  |
| --- | --- |
| Absolute Lengths | cm, mm, in, px (1/96th of an inch), pt (1/72nd of an inch), pc (picas; 12 points) |
| Relative Lengths | em: relative to the font-size of the element  ex: relative to the x-height of the current font  ch: relative to the width of "0"  rem: relative to the font-size of the root element  vw: relative to 1% of the width of the viewport  vh: relative to 1% of the height of the viewport  vmin: relative to 1% of the viewport smaller dimension  vmax: relative to 1% of the viewport larger dimension  %: relative to the parent element |
| Leading 0 | line-height: 0.7em; /\* the value has the potential to exceed 1.0 \*/  opacity: .5; /\* the value can never exceed 1.0 \*/ |
| CSS Math Functions | width: calc(100% - 20px); /\* 100% of the container's width minus 20px \*/  width: max(50%, 300px); /\* whichever is larger, 50% or 300px \*/  width: min(50%, 300px); /\* whichever is smaller, 50% or 300px \*/ |

# CSS SELECTORS

## PRIMARY SELECTORS

|  |  |  |
| --- | --- | --- |
| All HTML Elements | \* { ... } |  |
| Tag | h1 { ... } |  |
| ID  (NOT Recommended) | #a { ... } |  |
| Tag and ID  (NOT Recommended) | p#a { ... } |  |
| Class | .b { ... } |  |
| Tag and Class | p.b { ... } |  |
| Multiple Classes | .b.c { ... } |  |
| Multiple Selectors | h1, p { ... } |  |

## ATTRIBUTE SELECTORS

|  |  |  |
| --- | --- | --- |
| Attribute | [title] { ... } |  |
| Attribute with a Specified Value | [title="a"] { ... } |  |
| Attribute with Value Containing a Specified String as a Whole Word | [title~="a"] { ... } |  |
| Attribute with a Specified Value  (+ Hyphen) | [title|="a"] { ... } |  |
| Attribute with Value Starting with a Specified String | [title^="a"] { ... } |  |
| Attribute with Value Ending with a Specified String | [title$="a"] { ... } |  |
| Attribute with Value Containing a Specified String | [title\*="a"] { ... } |  |

## COMBINATORS (NESTED SELECTORS)

|  |  |  |
| --- | --- | --- |
| Descendant | .a p { ... } |  |
| Direct Child | .a > p { ... }  /\* no <p> is a direct clild of an element with class "a" \*/ |  |
| Adjacent Sibling | h1 + p { ... } |  |
| General Next Sibling | h1 ~ p { ... } |  |

## PSEUDO-CLASSES

### ORDER-RELATED PSEUDO-CLASSES

|  |  |  |
| --- | --- | --- |
| First Child | h1:first-child { ... } |  |
| p:first-child { ... }  /\* no <p> is the first clild of its parent \*/ |  |
| Last Child | p:last-child { ... } |  |
| h1:last-child { ... }  /\* no <h1> is thelast clild of its parent \*/ |  |
| First of Type | p:first-of-type { ... } |  |
| Last of Type | div:last-of-type { ... } |  |
| Nth of Type | p:nth-of-type(2) { ... } |  |
| Nth Child | p:nth-child(2) { ... } |  |
| Even Elements | p:nth-child(even) { ... }  /\* every second <p> starting from #2 \*/ |  |
| p:nth-child(2n) { ... } |
| p:nth-child(2n + 2) { ... } |
| Odd Elements | p:nth-child(odd) { ... }  /\* every second <p> starting from #1 \*/ |  |
| p:nth-child(2n + 1) { ... } |
| Odd Elements Starting from the Third | p:nth-child(2n + 3) { ... }  /\* every second <p> starting from #3 \*/ |  |
| Every Third Element | p:nth-child(3n) { ... } |  |
| Every Third Element Starting from the Second | p:nth-child(3n - 1) { ... } |  |
| Nth Child from the End | p:nth-last-child(2) { ... } |  |

### CHILD-PARENT PSEUDO-CLASSES

|  |  |  |
| --- | --- | --- |
| Only of Type | div:only-of-type { ... }  /\* the only <div> child of its parent \*/ |  |
| Only Child | p:only-child { ... }  /\* the only child of their parent \*/ |  |
| Empty | div:empty { ... }  /\* <div> with no children nor text content \*/ |  |

### LINK-RELATED PSEUDO-CLASSES

|  |  |  |
| --- | --- | --- |
| Target | :target { ... } |  |
| Visited & Unvisited Links | a:link { ... } /\* unvisited links \*/  a:visited { ... } /\* visited links \*/ |  |
| Hover | a:hover { ... } |  |
| Active Links | a:active { ... } |  |

### INPUT-RELATED PSEUDO-CLASSES

|  |  |  |
| --- | --- | --- |
| Focused Input | input:focus { ... } |  |
| Disabled Input | input:disabled { ... }  /\* attribute "disabled" \*/ |  |
| Enabled Input | input:enabled { ... }  /\* no attribute "disabled" \*/ |  |
| Required Input | input:required { ... }  /\* attribute "required" \*/ |  |
| Optional Input | input:optional { ... }  /\* no attribute "required" \*/ |  |
| Invalid Input | input:invalid { ... } |  |
| Valid Input | input:valid { ... } |  |
| Number Input with Value in Range | input:in-range { ... } |  |
| Number Input with Value out of Range | input:out-of-range { ... } |  |
| Read-Only Input | input:read-only { ... }  /\* attribute "readonly" or "disabled" \*/ |  |
| Read-Write Input | input:read-write { ... }  /\* no attribute "readonly" nor "disabled" \*/ |  |
| Checked Input | input:checked + label { ... }  /\* when input is checked, label is selected \*/ |  |

### GENERAL PSEUDO-CLASSES

|  |  |  |
| --- | --- | --- |
| Root | :root { ... }  /\* same as \*, if nothing specified for the children \*/ |  |
| html { ... } |
| Not | :not(p) { ... } |  |
| Use Several Pseudo-Classes at Once | div:nth-child(2):not(:hover) p { ... }  /\* all <p> in the 2nd <div> when the user's mouse is NOT over the <div> \*/ |  |

## PSEUDO-ELEMENTS

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | |
| Pseudo-Elements Used to Insert Something | ::after /\* insert content after an element; always with property "content":  content: url("image.png"); or content: ''; \*/  ::before /\* insert content before an element; always with property "content" \*/ | |
| Pseudo-Elements Used to Select Part of an Element | ::first-letter /\* only applied to block-level elements \*/  ::first-line /\* only applied to block-level elements \*/  ::marker /\* selects the markers of list items \*/  ::selection /\* matches the portion of an element that is selected by a user \*/ | |

# CSS SHORTHAND PROPERTIES

## EXAMPLES

|  |  |  |
| --- | --- | --- |
| About Shorthand Properties | They let you set the values of multiple CSS properties simultaneously. A value which is not specified is set to its default value, so it will override previously set values. | |
| Background | background: url('i.jpg') center/80%  no-repeat fixed border-box border-box green; | background-image: url('i.jpg');  background-position: center;  background-size: 80%;  background-repeat: no-repeat;  background-attachment: fixed;  background-origin: border-box;  background-clip: border-box;  background-color: gray; |
| Font | font: italic small-caps bold 12px/30px  Georgia, serif; | font-style: italic;  font-variant: small-caps;  font-weight: bold;  font-size: 12px;  line-height: 30px;  font-family: Georgia, serif; |
| List Style | list-style: disc inside url('sq.jpg'); | list-style-type: disc; /\* backup \*/  list-style-position: inside;  list-style-image: url('sq.jpg'); |

## 1-TO-4-VALUE SYNTAX FOR EDGES

|  |  |  |  |
| --- | --- | --- | --- |
| 1 Value | border-style: dashed; | border-top-width: dashed;  border-right-width: dashed;  border-bottom-width: dashed;  border-left-width: dashed; |  |
| 2 Values | margin: 20px 10px; | margin-top: 20px;  margin-right: 10px;  margin-bottom: 20px;  margin-left: 10px; |  |
| 3 Values | padding: 20px 5px 10px; | padding-top: 20px;  padding-right: 5px;  padding-bottom: 10px;  padding-left: 5px; |  |
| 4 Values | border-width: 2px 5px 3px 1px; | border-top-width: 2px;  border-right-width: 5px;  border-bottom-width: 3px;  border-left-width: 1px; |  |

## 1-TO-4-VALUE SYNTAX FOR CORNERS

|  |  |  |  |
| --- | --- | --- | --- |
| 1 Value | border-radius: 1em; | border-top-left-radius: 1em;  border-top-right-radius: 1em;  border-bottom-right-radius: 1em;  border-bottom-left-radius: 1em; |  |
| 2 Values | border-radius: 1em 2em; | border-top-left-radius: 1em;  border-top-right-radius: 2em;  border-bottom-right-radius: 1em;  border-bottom-left-radius: 2em; |  |
| 3 Values | border-radius: 1em 3em 2em; | border-top-left-radius: 1em;  border-top-right-radius: 3em;  border-bottom-right-radius: 2em;  border-bottom-left-radius: 3em; |  |
| 4 Values | border-radius: 1px 3px 2px 4px; | border-top-left-radius: 1px;  border-top-right-radius: 3px;  border-bottom-right-radius: 2px;  border-bottom-left-radius: 4px; |  |

# COMMON CSS PROPERTIES

## BACKGROUND

### SOLID COLOR AND GRADIENT

|  |  |  |
| --- | --- | --- |
| Solid Color Background | background-color: #fff | transparent | navy | rgb(125, 125, 125); | |
| background-color: #2E74B5; /\* default: transparent \*/ |  |
| background-color: rgb(46, 116, 181); |
| Semi-Transparent Background | background-color: #2E74B5;  opacity: 0.5; |  |
| background-color: rgba(46, 116, 181, 0.5); |
| Linear Gradient Background | background-image: linear-gradient(#2E74B5, gray); |  |
| Sphere Effect | background-image: radial-gradient(circle at 20px 20px,  #2E74B5, rgba(0, 0, 0, 0.7)); |  |
| Background Shorthand Property | background: #fff url('image.png') no-repeat fixed -47px 0;  /\* color, image, repeat, attachment, position \*/ | |

### IMAGE BACKGROUND

|  |  |  |
| --- | --- | --- |
| Image Background | background-image: url('tree.jpg');  (opacity: 1;) /\* default \*/  (background-position: left top;) /\* default \*/  (background-size: auto;) /\* default \*/  (background-attachment: scroll;) /\* default \*/ |  |
| Semi-Transparent Image Background | background-image: url('tree.jpg');  opacity: 0.5; /\* inherited by children, affects text \*/ |  |
| background-image: linear-gradient(rgba(255, 255, 255, 0.5),  rgba(255, 255, 255, 0.5)), url('tree.jpg');  /\* does not affect text \*/ |  |
| Image Starting 20px Down from the Top | background-position: left -20px; |  |
| Image Starting at the Bottom of the Element | background-position: bottom (center); |  |
| background-position: center bottom; |
| background-position: 50% 100%; |
| Image Starting in the Middle of the Element | background-position: center; |  |
| background-position: 50%; |
| Image to Cover 100% of the Element | background-size: 100%; |  |
| background-size: cover; |
| Centered Image to Cover 100% of the Element | background-size: cover;  background-position: center; |  |
| Image to Cover 100% of the Element's Height | background-size: auto 100%;  (background-repeat: repeat;) /\* default \*/ |  |
| Image Displayed as Large as Possible | background-size: contain; |
| Image to Stretch over the Whole Element | background-size: 100% 100%; |  |
| Image to Cover 34% of the Element's Width | background-size: 34%; |  |
| Non-Repeated Image to Cover 34% of the Element's Width | background-size: 34%;  background-repeat: no-repeat; |  |
| Image 34% of the Element's Width & Repeated on the X-Axis | background-size: 34%;  background-repeat: repeat-x; |  |
| Image 34% of the Element's Width & Repeated on the Y-Axis | background-size: 34%;  background-repeat: repeat-y; |  |
| Single Image Centered | background-size: contain;  background-repeat: no-repeat;  background-position: center; |  |
| Image to Cover 34% of the Element in Width & the Whole in Height | background-size: 34% 100%; |  |
| Image to Cover 34% of the Element in Height & the Whole in Width | background-size: 100% 34%; |  |
| Image to Remain Fixed by Scrolling | background-attachment: fixed; |  |
| Several Image Backgrounds | background: url('tree.png') no-repeat bottom left,  url('texture.jpg'); /\* first one is displayed on top \*/  background-size: 30%, auto; /\* tree: 30%, texture: auto \*/ |  |

### BACKGROUND STARTING POINT

|  |  |  |
| --- | --- | --- |
| Image Starting from the Upper Left Corner of the Padding Edge | background-origin: padding-box; /\* default \*/ |  |
| Image Starting from the Upper Left Corner of the Border | background-origin: border-box; |  |
| Image Starting from the Upper Left Corner of the Content | background-origin: content-box; |  |
| Image Painted to the Outside Edge of the Border | background-clip: border-box; /\* default \*/ |  |
| Image Painted to the Outside Edge of the Padding | background-clip: padding-box; |  |
| Image Painted within the Content Box | background-clip: content-box; |  |

## BORDER

### DRAW BORDER

|  |  |  |
| --- | --- | --- |
| Solid Color Border | border-style: solid;  (border-color: black;) /\* default \*/  (border-width: 3px;) /\* default \*/ |  |
| Set Border Color | border-style: solid;  border-color: gray; |  |
| Set Border Width | border-style: solid;  border-width: 5px; or border-width: thick; /\* thin | medium \*/ |  |
| Dotted Border | border-style: dotted; |  |
| Dashed Border | border-style: dashed; |  |
| Double Bprder | border-style: double; |  |
| Carved Border | border-style: groove; |  |
| 3D Inset Border | border-style: inset; |  |
| 3D Outset Border | border-style: outset; |  |
| No Border | border-style: hidden; /\* border included in the width \*/ |  |
| border-style: none; /\* no border at all \*/ |
| Only Style One Side of the Border | border-left-style: solid; |  |
| border-style: none none none solid; /\* top, right, bottom, left \*/ |
| border-left: 3px solid gray; /\* width, style, color \*/ |
| Style Two Sides of the Border Differently | border-top-width: 5px;  border-right-width: 2px;  border-bottom-width: 5px;  border-left-width: 2px; |  |
| border-width: 5px 2px 5px 2px; /\* top, right, bottom, left \*/ |
| border-width: 5px 2px; /\* top & bottom, right & left \*/ |
| Style Three Sides of the Border Differently | border-top-color: pink;  border-right-color: #2E74B5;  border-bottom-color: gray;  border-left-color: #2E74B5; |  |
| border-color: pink #2E74B5 gray; /\* top, right & left, bottom \*/ |
| Border Shorthand Property | border: 2px solid black; /\* width, style, color \*/ | |

### USE IMAGE

|  |  |  |
| --- | --- | --- |
| Border Image | border-image-source: url('tree.jpg');  border-image-width: 20px;  border-image-slice: 10%;  (border-image-repeat: stretch;) /\* default \*/ |  |
| 30% of the Image Displayed in the Border | border-image-slice: 30%; /\* all sides \*/ |  |
| 30% of the Image Displayed in the Border & Rescaled to Fit | border-image-slice: 30%; /\* all sides \*/  border-image-repeat: round; |  |
| 30% of the Image Displayed Repeatedly | border-image-slice: 30%; /\* all sides \*/  border-image-repeat: repeat; |  |
| 30% of the Image Displayed, Extra Space Distributed Around | border-image-slice: 30%; /\* all sides \*/  border-image-repeat: space; |  |
| 30% of the Image Displayed, Inside of the Border Filled | border-image-slice: 30% fill; /\* all sides + inside \*/  border-image-repeat: round; |  |
| 30% of the Image's Height & 50% of the Image's Width | border-image-slice: 30% 50%; /\* vertical, horizontal \*/  border-image-repeat: round; |  |
| 30% of the Image's Width & 50% of the Image's Height | border-image-slice: 50% 30%; /\* vertical, horizontal \*/  border-image-repeat: round; |  |
| 50% of the Image Displayed | border-image-slice: 50%; /\* all sides \*/ |  |
| Border Image Shorthand Property | border-image: url('border.png') 30% round; /\* source, slice, repeat \*/ | |

### ROUNDED CORNERS

|  |  |  |
| --- | --- | --- |
| Rounded Corners | border-radius: 15px; /\* works with no border \*/ |  |
| Only Round Two Corners | border-top-right-radius: 15px;  border-bottom-right-radius: 15px; |  |
| border-radius: 0 15px 15px 0;  /\* top left, top right, bottom right, bottom left \*/ |
| Ellipse or Circle | border-radius: 50%;  /\* if width and height are equal: a circle \*/ |  |

### TABLE BORDER

|  |  |  |
| --- | --- | --- |
| Separate Borders for Table and Cells | border-collapse: separate; /\* default \*/ |  |
| Set Spacing between Table and Cell Borders | border-collapse: separate;  border-spacing: 10px; |  |
| Single Border for Table and Cells | border-collapse: collapse; |  |
| Rounded Corners for Table with Collapsed Border | border-collapse: collapse;  border-radius: 15px;  overflow: hidden; |  |

## SHADOW

|  |  |  |
| --- | --- | --- |
| Text Shadow | text-shadow: 2px 2px lightblue;  /\* horizontal, vertical, color \*/ |  |
| Blurred Shadow | text-shadow: 2px 2px 4px black;  /\* horizontal, vertical, blur, color \*/ |  |
| Neon Glow | text-shadow: 0 0 3px blue;  /\* horizontal, vertical, blur, color \*/ |  |
| Multiple Shadows | text-shadow: 0 0 3px red, 0 0 5px yellow;  /\* horizontal, vertical, blur, color \*/ |  |
| Outlined Text | text-shadow: -1px 0 gray, 0 -1px gray, 1px 0 gray, 0 1px gray; |  |
| Box Shadow | box-shadow: 10px 10px gray;  /\* horizontal, vertical, color \*/ |  |
| Multiple Box Shadows | box-shadow: 2px 2px 8px black, 8px 8px 5px 2px gray;  /\* horizontal, vertical, blur, spread, color \*/ |  |
| Card Effect | box-shadow: 0 4px 8px 0 rgba(0, 0, 0, .2),  0 6px 20px 0 rgba(0, 0, 0, .19);  /\* horizontal, vertical, blur, spread, color + opacity \*/ |  |

## LIST STYLING

|  |  |  |
| --- | --- | --- |
| List Styles | list-style: square inside url('../image'); | |
| Unordered Lists | ul { list-style-type: none; }  /\* to remove default settings add margin: 0; padding: 0; \*/ |  |
|  | ul { list-style-type: disc; } /\* default \*/ |  |
|  | ul { list-style-type: circle; } |  |
|  | ul { list-style-type: square; } |  |
|  | ul { list-style-image: url('sqpurple.jpg'); } |  |
| Ordered Lists | ol { list-style-type: lower-alpha; } |  |
| ol { list-style-type: upper-alpha; } |  |
| ol { list-style-type: upper-roman; } |  |
| List Bullet/Number Position | ol { list-style-position: outside; } /\* default \*/ |  |
| ol { list-style-position: inside; } |  |
| List Style Shorthand Property | list-style: square inside url('sqpurple.jpg');  /\* backup type, position, image \*/ | |

## RESIZING

|  |  |  |
| --- | --- | --- |
| Non-Resizable Elements | resize: none; /\* default \*/ |  |
| Resizable Width | resize: horizontal; |  |
| Resizable Height | resize: vertical; |  |
| Resizable Width and Height | resize: both;  /\* default for textarea in many browsers \*/ |  |

## CURSOR

|  |  |  |
| --- | --- | --- |
| General | cursor: auto | default;  /\* default; cursor based on the current context \*/ |  |
| cursor: default;  /\* the platform default cursor \*/ |  |
| cursor: none; |  |
| Links & Status | cursor: pointer;  /\* indicates a link \*/ |  |
| cursor: help;  /\* help information is available \*/ |  |
| cursor: progress;  /\* program busy in the background, but the user can still interact with the interface \*/ |  |
| cursor: wait;  /\* program busy and the user cannot interact with the interface \*/ |  |
| Selection | cursor: text;  /\* text can be selected \*/ |  |
| cursor: vertical-text;  /\* vertical text can be selected \*/ |  |
| cursor: crosshair; |  |
| cursor: cell; |  |
| Drag & Drop | cursor: grab; |  |
| cursor: grabbing; |  |
| cursor: move;  /\* all-scroll is the same in Windows \*/ |  |
| cursor: copy; |  |
| cursor: alias; |  |
| cursor: not-allowed; |  |
| Resizing & Scrolling | cursor: ew-resize; |  |
| cursor: ns-resize; |  |
| cursor: ne-resize; |  |
| cursor: nw-resize; |  |
| cursor: col-resize; |  |
| cursor: row-resize; |  |
| Zooming | cursor: zoom-in; |  |
| cursor: zoom-out; |  |
| Enable/Disable Pointer Events | pointer-events: auto;  /\* default; the element reacts to events like hover and click \*/ |  |
| pointer-events: none;  /\* the element does not react to pointer events \*/ |  |

# CSS COLORS

## COLORABLE PARTS OF AN ELEMENT

|  |  |  |
| --- | --- | --- |
| Background | background-color: gray; |  |
| Text | color: white; |  |
| Border | border-color: lightskyblue; |  |

## SOLID COLORS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Color Name | Red, Green, Blue | Hex Value | Hue, Saturation, Lightness |  |
| white | rgb(255, 255, 255) | #ffffff /\* #fff \*/ | hsl(0, 0%, 100%) |  |
| /\* 100% red, 100% green, 100% blue \*/ | | /\* red hue, 100% gray, 50% lighter \*/ |
| black | rgb(0, 0, 0) | #000000 /\* #000 \*/ | hsl(0, 0%, 0%) |  |
| /\* 0% red, 0% green, 0% blue \*/ | | /\* red hue, 100% gray, 50% darker \*/ |
| lightgray | rgb(211, 211, 211) | #d3d3d3 | hsl(0, 0%, 83%) |  |
| /\* 82.8% red, 82.8% green, 82.8% blue \*/ | | /\* red hue, 100% gray,  33% lighter \*/ |
| gray | rgb(128, 128, 128) | #808080 | hsl(0, 0%, 50%) |  |
| /\* 50% red, 50% green, 50% blue \*/ | | /\* red hue, 100% gray,  neutral lightness \*/ |
| red | rgb(255, 0, 0) | #ff0000 /\* #f00 \*/ | hsl(0 | 360, 100%, 50%) |  |
| /\* 100% red, 0% green, 0% blue \*/ | | /\* red hue, no gray,  neutral lightness \*/ |
| green | rgb(0, 255, 0) | #00ff00 /\* #0f0 \*/ | hsl(120, 100%, 50%) |  |
| /\* 0% red, 100% green, 0% blue \*/ | | /\* green hue, no gray,  neutral lightness \*/ |
| blue | rgb(0, 0, 255) | #0000ff /\* #00f \*/ | hsl(240, 100%, 50%) |  |
| /\* 0% red, 0% green, 100% blue \*/ | | /\* blue hue, no gray,  neutral lightness \*/ |
| yellow | rgb(255, 255, 0) | #ffff00 /\* #ff0 \*/ | hsl(60, 100%, 50%) |  |
| /\* 100% red, 100% green, 0% blue \*/ | | /\* yellow hue (red + 60),  no gray, neutral lightness \*/ |
| pink | rgb(255, 192, 203) | #ffc0cb | hsl(350, 100%, 88%) |  |
| /\* 100% red, 75.3% green, 79.6% blue \*/ | | /\* almost red hue (- 10),  no gray, 38% lighter \*/ |
| magenta | rgb(255, 0, 255) | #ff00ff /\* #f0f \*/ | hsl(300, 100%, 50%) |  |
| /\* 100% red, 0% green, 100% blue \*/ | | /\* magenta hue (blue + 60),  no gray, neutral lightness \*/ |
| tomato | rgb(255, 99, 71) | #ff6347 | hsl(9, 100%, 64%) |  |
| /\* 100% red, 38.8% green, 27.8% blue \*/ | | /\* reddish hue (+ 9),  no gray, 14% lighter \*/ |
| mediumseagreen | rgb(60, 179, 113) | #3cb371 | hsl(147, 50%, 47%) |  |
| /\* 23.5% red, 70.2% green, 44.3% blue \*/ | | /\* greenish hue (+ 27),  50% gray, 3% darker \*/ |

## COLORS & OPACITY

|  |  |  |  |
| --- | --- | --- | --- |
| Fully Opaque | rgba(255, 0, 0, 1) | hsla(0, 100%, 50%, 1) |  |
| /\* 100% red, 0% green,  0% blue, 100% alpha \*/ | /\* red hue, no gray,  neutral lightness, 100% alpha \*/ |
| 30% Transparent | rgba(255, 0, 0, .7) | hsla(0, 100%, 50%, .7) |  |
| /\* 100% red, 0% green,  0% blue, 70% alpha \*/ | /\* red hue, no gray,  neutral lightness, 70% alpha \*/ |
| 50% Transparent | rgba(255, 0, 0, .5) | hsla(0, 100%, 50%, .5) |  |
| /\* 100% red, 0% green,  0% blue, 50% alpha \*/ | /\* red hue, no gray,  neutral lightness, 50% alpha \*/ |
| 80% Transparent | rgba(255, 0, 0, .2) | hsla(0, 100%, 50%, .2) |  |
| /\* 100% red, 0% green,  0% blue, 20% alpha \*/ | /\* red hue, no gray,  neutral lightness, 20% alpha \*/ |
| Fully Transparent | rgba(255, 0, 0, 0) | hsla(0, 100%, 50%, 0) |  |
| /\* 100% red, 0% green,  0% blue, 0% alpha \*/ | /\* red hue, no gray,  neutral lightness, 0% alpha \*/ |

## CURRENT COLOR

|  |  |  |
| --- | --- | --- |
| Use the Current Value of the Element's Color Property | div {  color: gray;  border: 10px solid currentColor;  } |  |
| Use the Current Value of the Parent's Color Property | body { color: gray; }  div { background-color: currentColor; } |  |

# CSS GRADIENTS

## LINEAR GRADIENTS

|  |  |  |
| --- | --- | --- |
| Top to Bottom | linear-gradient((to bottom), #2E74B5, gray);  linear-gradient(180deg, #2E74B5, gray);  /\* direction, first color, second color \*/ |  |
| Left to Right | linear-gradient(to right, #2E74B5, gray);  linear-gradient(90deg, #2E74B5, gray); |  |
| Three Colors, Diagonally | linear-gradient(to bottom right,  #2E74B5 50%, lightgray, gray);  /\* direction, first color + stop point, second, third color \*/ |  |
| Repeating Stripes | repeating-linear-gradient(#2E74B5,  lightgray 25%, gray 25%); |  |

## RADIAL GRADIENTS

|  |  |  |
| --- | --- | --- |
| Centered Ellipse | radial-gradient((ellipse) (at center),  gray 10%, lightgray, #2E74B5);  /\* shape, position, first color + stop point, second, third color \*/ |  |
| Circle Reaching the Farthest Corner | radial-gradient((farthest-corner) circle at 70% 20%,  gray 10%, lightgray 30%, #2E74B5); |  |
| Circle Reaching the Closest Edge | radial-gradient(closest-side at 20% 30%,  gray 10%, lightgray 30%, #2E74B5); |  |
| Repeating Concentric Circles | repeating-radial-gradient(closest-side at 20% 30%,  gray 10%, lightgray 30%, #2E74B5); |  |
| Two Circles in an Opacity Gradient | radial-gradient(at top,  rgba(46, 116, 181, 1), rgba(46, 116, 181, 0)),  radial-gradient(at bottom,  rgba(150, 150, 150, 1), rgba(150, 150, 150, 0)); |  |

## CONIC GRADIENTS

|  |  |  |
| --- | --- | --- |
| Colors Changing Gradually | conic-gradient(gray 45deg, lightgray 90deg,  #2E74B5 270deg);  /\* color + stop point \*/ |  |
| Pie Chart | conic-gradient(gray 0deg 90deg, lightgray 90deg 180deg, #2E74B5 180deg);  /\* color + start & stop point \*/ |  |
| Cone with Gradually Changing Colors | conic-gradient(from 90deg at 60% 70%,  gray 90deg, lightgray 180deg, #2E74B5 270deg);  /\* start point, position, color + stop point \*/ |  |
| Pie Chart with Gradually Changing Repeating Pieces | repeating-conic-gradient(gray 10%, lightgray 10%,  #2E74B5 20%); |  |
| Pie Chart with Repeating Pieces | repeating-conic-gradient(gray 0deg 10deg,  lightgray 10deg 20deg, #2E74B5 20deg 30deg; |  |

# CSS TEXT

## FONT FORMATTING

### GENERAL FONT FORMATTING

|  |  |  |
| --- | --- | --- |
| Font Family | font-family: 'Times New Roman';  /\* default for Chrome \*/ |  |
| font-family: Verdana, Arial, sans-serif; /\* the first one the browser has: Verdana, Arial or default non-serif font \*/ |  |
| Generic Font Families | font-family: serif; /\* default serif font, Times New Roman for Chrome \*/ |  |
| font-family: sans-serif; /\* default sans-serif font, Arial for Chrome \*/ |  |
| font-family: cursive; /\* default cursive font,  Comic Sans MS for Chrome \*/ |  |
| font-family: monospace; /\* default monospace font, Consolas for Chrome \*/ |  |
| font-family: fantasy; /\* default fantasy font, Impact for Chrome \*/ |  |
| Font Style | font-style: italic; /\* default: normal \*/ |  |
| Font Weight | font-weight: bold | 600; /\* default: normal or 400 \*/ |  |
| Font Variant | font-variant: small-caps; /\* default: normal \*/ |  |
| Lower/Upper Case or Capitalized Words | .first-line { text-transform: lowercase; }  .second-line { text-transform: uppercase; }  .third-line { text-transform: capitalize; }  /\* default: none \*/ |  |
| Line under, over or through Text | .first-line { text-decoration: overline; }  .second-line { text-decoration: underline; }  .third-line { text-decoration: line-through; }  .fourth-line { text-decoration: none; }  /\* default: none; for links: underline \*/ |  |
| Font Size | font-size: 16px | normal | 1em | 100%;  /\* default for Chrome \*/ |  |
| font-size: 19.2px | larger; |  |
| font-size: 13.333px | smaller; |  |
| font-size: 5vw;  /\* responsive: 5% of the screen width \*/ |  |
| Line Height | line-height: 1.15(em) | normal;  /\* default for Chrome \*/ |  |
| line-height: 0.5 | 8px; |  |
| line-height: 2 | 32px; |  |
| Letter Spacing | letter-spacing: normal | 0.01px;  /\* default for Chrome \*/ |  |
| letter-spacing: 2px; |  |
|  | q:lang(no) { quotes: "~" "~" } /\* quotation marks for NO language: ~~ \*/ |  |
| Font Shorthand Property | font: italic small-caps bold 12px/30px Georgia, serif;  /\* style, variant, weight, size/line-height, family \*/ | |

### TEXT COLOR AND SHADOW

|  |  |  |
| --- | --- | --- |
| Text Color | (text-)color: rgb(46, 116, 181);  /\* default: black \*/ |  |
| Simple Shadow | text-shadow: 2px 2px lightblue;  /\* horizontal, vertical, color \*/ |  |
| Blurred Shadow | text-shadow: 2px 2px 4px black;  /\* horizontal, vertical, blur, color \*/ |  |
| Neon Glow | text-shadow: 0 0 3px blue;  /\* horizontal, vertical, blur, color \*/ |  |
| Multiple Shadows | text-shadow: 0 0 3px red, 0 0 5px yellow;  /\* horizontal, vertical, blur, color \*/ |  |
| Outlined Text | text-shadow: -1px 0 gray, 0 -1px gray, 1px 0 gray, 0 1px gray; |  |

### FONT PAIRINGS

|  |  |
| --- | --- |
| Heading: Georgia,  Text: Verdana |  |
| Heading: Merriweather,  Text: Open Sans |  |
| Heading: Abril Fatface,  Text: Poppins |  |
| Heading: Spectral,  Text: Rubik |  |

### GOOGLE FONTS

|  |  |  |
| --- | --- | --- |
| About Google Fonts | Google Fonts is a place where we can check and download a font (type something in cyrillic!). | |
| Import Google Font in the HTML File | <link *rel*="stylesheet" *href*="https://fonts.googleapis.com/css?family=Sofia"> | |
| Import Multiple Fonts in the HTML File | <link *rel*="stylesheet"  *href*="https://fonts.googleapis.com/css?family=Sofia|Trirong|Audiowide"> | |
| Import Font Effects in HTML File | <link *rel*="stylesheet"  *href*="https://fonts.googleapis.com/css?family=Sofia&effect=fire"> | |
| Use Imported Font Effects | <p *class*="font-effect-fire">Sofia on fire</p> |  |
| Import Google Font in the CSS File with @import (for JudgeW) | @import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');  /\* Google Fonts > Search "Roboto" > Select this style > @import \*/ | |
| Import Google Font in the CSS File with @font-face | @font-face {  font-family: "Gret";  src: url("./Gret/Gret-bold.woff") format("woff"); /\* .ttf > format("truetype") \*/  (font-weight: bold;)  } | |

### FONT AWSOME ICONS

|  |  |  |  |
| --- | --- | --- | --- |
| Import Font Awsome in the HTML File | <head>  *Font Awsome > Sign In > Profile Pic > Font Awsome CDN > copy <link>*  </head> | | |
| Import Font Awsome in the CSS File | @import url("*Font Awsome > Sign In > Profile Pic > Font Awsome CDN > get href value from <link>*"); | | |
| Use Font Awsome Icons | <i *class*="fas fa-star">  </i> | color: gold;  font-size: 20px; |  |
| <i *class*="fab fa-facebook-square">  </i> | color: #2E74B5;  font-size: 26px; |  |

## TEXT LAYOUT

### TEXT ALIGNMENT

|  |  |  |
| --- | --- | --- |
| Left | (text-align: left | start;)  /\* default \*/ |  |
| Right | text-align: right | end; |  |
| Center | text-align: center; |  |
| Justify | text-align: justify; |  |
| Vertical Alignment of an Inline Element in a Text or a Text in a Table | .first img { vertical-align: baseline; } /\* default \*/  .second img { vertical-align: text-top; }  .third img { vertical-align: text-bottom; }  .fourth img { vertical-align: super; }  .fifth img { vertical-align: sub; } |  |

### LINE LAYOUT & WORD BREAKING

|  |  |  |
| --- | --- | --- |
| Text with Words Longer than Container | (word-wrap: normal;) /\* default \*/ |  |
| Break Long Words | word-wrap: break-word; |  |
| Break All Words | word-break: break-all; /\* default: normal \*/ |  |
| Force Text to Stay on the Same Line | white-space: nowrap; /\* default: normal \*/ |  |
| Force Text to Stay on the Same Line & Hide Overflow | white-space: nowrap;  overflow: hidden; /\* default: visible \*/ |  |
| Force Text to Stay on the Same Line & Indicate Overflow with Ellipsis | white-space: nowrap;  overflow: hidden;  text-overflow: ellipsis; /\* default: clip \*/ |  |
| Set Indentation for the First Line | text-indent: 20px;  /\* default: 0 \*/ |  |
| Add a Scrollbar | overflow: scroll; /\* adds a scrollbar \*/ |  |
| overflow: auto; /\* adds a scrollbar wher needed \*/  overflow-y: scroll; /\* adds a vertical scrollbar \*/ |
| Set Space between Words | word-spacing: 10px;  /\* default: 0 \*/ |  |

### TEXT DIRECTION

|  |  |  |
| --- | --- | --- |
| Vertical with Rows Right to Left | writing-mode: vertical-rl; /\* default: horizontal-tb \*/ |  |
| Vertical with Rows Left to Right | writing-mode: vertical-lr; /\* default: horizontal-tb \*/ |  |
| Horizontal Right to Left | direction: rtl;  unicode-bidi: bidi-override; |  |

# CSS IMAGES

## GENERAL IMAGE POSITIONING

|  |  |  |
| --- | --- | --- |
| Image to Fill the Whole Area | width: 150px;  height: 100px;  (object-fit: fill;) /\* default \*/  (object-position: center;) /\* default; 50% 50% \*/ |  |
| background-size: 100% 100%; |
| Image to Cover the Whole Area (Cropped if Necessary) | object-fit: cover; |  |
| background-size: cover;  background-position: center; |
| Image to Appear Whole in the Area (Parts May Remain Uncovered) | object-fit: contain; |  |
| background-size: contain;  background-repeat: no-repeat;  background-position: center; |
| Image to Cover Area Starting from the Center | object-fit: none; |  |
| background-position: center; |
| Vertically Centered Image, 50px to the Right | object-fit: none;  object-position: 50px 50%; |  |
| background-repeat: no-repeat;  background-position: 50px 50%; |
| The Smallest of None and Contain | object-fit: scale-down;  /\* the smallest version of none and contain \*/ |  |
| Responsive Images | max-width: 100%;  height: auto | 100%; |  |

## IMAGE FILTERS AND MASKS

|  |  |  |
| --- | --- | --- |
| No Filter | filter: none; |  |
| Blurred Image | filter: blur(4px); |  |
| Set Image Brightness | filter: brightness(150%); |  |
| Set Image Contrast | filter: contrast(150%); |  |
| Make Image Black & White | filter: grayscale(100%); |  |
| Rotate Image Hue | filter: hue-rotate(180deg); |  |
| Invert Image Colors | filter: invert(100%); |  |
| Set Image Opacity | filter: opacity(50%); |  |
| Set Image Saturation | filter: saturate(7); |  |
| Apply Sepia Effect | filter: sepia(70%); |  |
| Add Shadow | filter: drop-shadow(8px 8px 5px gray); |  |
| Apply Masking Layer | -webkit-mask-image: radial-gradient(circle at center,  black 40%, rgba(0, 0, 0, .5) 0);  mask-image: radial-gradient(circle at center,  black 40%, rgba(0, 0, 0, .5) 0)); |  |

## IMAGE REFLECTION

|  |  |  |
| --- | --- | --- |
| Reflection to the Right | -webkit-box-reflex: right (0px); /\* above | below | left \*/ |  |
| .second-img { transform: rotateY(180deg); } |
| Reflection Fading out under Image | -webkit-box-reflex: below 0 linear-gradient(to bottom,  transparent, rgba(0, 0, 0, .3)); |  |
| .second-img {  transform: rotateY(180deg);  -webkit-mask-image: linear-gradient(to bottom,  transparent, rgba(0, 0, 0, .3));  mask-image: linear-gradient(to bottom,  transparent, rgba(0, 0, 0, .3));  } |

## IMAGE SPRITES

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

# 2D TRANSFORMATIONS

|  |  |  |
| --- | --- | --- |
| Move an Element | transform: translate(20px, 40px);  /\* horizontal, vertical \*/ |  |
| Rotate an Element | transform: rotate(20deg);  /\* -20deg: counterclockwise \*/ |  |
| Make an Element Bigger/Smaller | transform: scale(2, 0.5);  /\* twice as wide, half as high \*/ |  |
| Make an Element More/Less Wide | transform: scaleX(2); /\* twice as wide \*/ |  |
| transform: scale(2, 1); /\* twice as wide \*/ |
| Make an Element More/Less Tall | transform: scaleY(0.5); /\* half as high \*/ |  |
|  | transform: scale(1, 0.5); /\* half as high \*/ |
| Flip an Element Horizontally | transform: scale(-1, 0.5);  /\* flipped, half as high \*/ |  |
| Skew an Element Horizontally | transform: skewX(20deg); |  |
| transform: skew(20deg, (0)); |
| Skew an Element Vertically | transform: skewY(20deg); |  |
| transform: skew(0, 20deg); |
| Apply All 2D Transformations at Once | transform: matrix(1.5, -0.2, 0.1, 1, 0, 20);  /\* scaleX, skewX, skewY, scaleY, translateX, translateY \*/ |  |

# 3D TRANSFORMATIONS

|  |  |  |
| --- | --- | --- |
| Rotate an Element around the X-Axis | transform: rotateX(130deg); |  |
| Rotate an Element around the Y-Axis | transform: rotateY(130deg); |  |
| Rotate an Element around the Z-Axis | transform: rotateZ(130deg); |  |
| transform: rotate(130deg); |

# TRANSITIONS

|  |  |  |
| --- | --- | --- |
| Change Element's Width | div { width: 100px; height: 100px; }  transition-property: width;  transition-duration: 2s; /\* default: 0s \*/  (transition-timing-function: ease;) /\* default \*/  (transition-delay: 0;) /\* default \*/  }  div:hover { width: 200px; } |  |
| Rotate Element while Changing Its Opacity | div {  transition: transform 2s linear 1s, opacity 4s;  /\* property, duration, timing function, delay \*/  }  div:hover {  transform: rotate(90deg);  opacity: 0.5;  } |  |
| Square to Circle | div {  transition: borded-radius 1s ease-in;  }  div:hover { border-radius: 50%; } |  |
| Flip Effect | div {  transition: background-color 0s linear 1s,  transform 2s linear;  }  div:hover {  background-color: gray;  transform: rotateY(180deg);  } |  |
| Move an Element while Changing Its Dimensions | div { transition: transform 1s ease-in-out; }  div:hover {  transform: translate(50px -50px) scale(1.5);  } |  |
| Add Tooltip | .tooltip .tooltiptext {  opacity: 0;  transition: opacity 1s;  }  .tooltip:hover .tooltiptext {  opacity: 1;  } |  |
| Make an Overlay Layer Appear in Front of Image | .tree { position: relative; }  .overlay {  position: absolute;  top: 0; right: 0; bottom: 0; left: 0;  opacity: 0;  transition: opacity 1s;  }  .overlay:hover { opacity: 1; } |  |
| Make an Overlay Layer Drop down in Front of Image | .tree { position: relative; }  .overlay {  position: absolute;  top: 0; right: 0; bottom: 100%; left: 0;  overflow: hidden;  transition: bottom .5s;  }  .overlay:hover { bottom: 0; } |  |
| Make a Close Button Hoverable | .btn {  position: absolute;  top: 5px; right: 10px;  transition: .3s;  }  .btn:hover {  opacity: .5;  cursor: pointer;  } |  |

# ANIMATIONS

|  |  |  |  |
| --- | --- | --- | --- |
| Change an Element's Color | div {  background-color: #2E74B5;  animation-name: color-change;  animation-duration: 2s; /\* default: 0s \*/  (animation-timing-function: ease;) /\* default \*/  (animation-delay: 0;) /\* default; takes negative values \*/  (animation-iteration-count: 1;) /\* default \*/  (animation-direction: normal;) /\* default \*/  (animation-fill-mode: none;) /\* default \*/  }  @keyframes color-change {  from { background-color: #2E74B5; } /\* 0%, start \*/  to { background-color: gray; } /\* 100%, complete \*/  } | |  |
| Move an Element while Changing Its Opacity | div {  background-color: #2E74B5;  animation: move 6s ease-out 1s infinite;  /\* name, duration, timing function, delay, iteration \*/  }  @keyframes move {  0% { background: rgb(46, 116, 181); left: 0; top: 0; }  25% { background: rgba(46, 116, 181, 0.7);  left: 0; top: 50px; }  50% { background: rgba(46, 116, 181, 0.5);  left: 50px; top: 50px; }  75% { background: rgba(46, 116, 181, 0.7);  left: 0; top: 50px; }  100% { background: rgb(46, 116, 181); left: 0; top: 0; }  } | |  |
| Move an Element in Alternate Directions | animation: move 6s 2 alternate;  /\* name, duration, iteration, direction \*/ | |  |
| Zoom In | div { animation: zoom .6s; }  @keyframes zoom {  from { transform: scale(0.1); }  to { transform: scale(1); }  } | |  |
| Animation Timing Function Demo | animation: move 5s 1 forwards;  /\* name, duration, iteration, fill mode \*/  .a { animation-timing-function: linear; }  .b { animation-timing-function: ease; }  .c { animation-timing-function: ease-in; }  .d { animation-timing-function: ease-out; }  .e { animation-timing-function: ease-in-out; } |  | |

# CSS COUNTERS

|  |  |  |  |
| --- | --- | --- | --- |
| About CSS Counters | CSS counters are like variables maintained by CSS whose values can be incremented by CSS rules (to track how many times they are used). Counters let you adjust the appearance of content based on its placement in the document. | | |
| Create a CSS Counter | counter-reset: section 3; /\* counter named "section" initially set to 3 \*/ | | |
| Increment Counter | counter-increment: section; | | |
| Use Counter to Number Headings | body { counter-reset: part 3; }  h2::before {  counter-increment: part;  content: "Part " counter(part) ": ";  } | <h2>One</h2>  <h2>Two</h2>  <h2>Three</h2> | Part 4: One  Part 5: Two  Part 6: Three |
| Use Nesting Counters | body { counter-reset: title; }  h1 { counter-reset: subtitle; }  h1::before {  counter-increment: title;  content: "Part " counter(title) ": ";  }  h2::before {  counter-increment: subtitle;  content: "Part " counter(title) "."  counter(subtitle) ": ";  } | <h1>One</h1>  <h2>Two</h2>  <h2>Three</h2>  <h1>One</h1>  <h2>Two</h2>  <h2>Three</h2> | Part 1: One  Part 1.1: Two  Part 1.2: Three  Part 2: One  Part 2.1: Two  Part 2.2: Three |
| Nesting One Counter | ul { counter-reset: title; }  li::before {  counter-increment: title;  content: counters(title, ".") ". ";  } | <ul>  <li>item</li>  <li>item</li>  <ul>  <li>item</li>  <li>item</li>  </ul>  </ul> | 1. item  2. item  2.1. item  2.2. item |

# CSS WEBSITE LAYOUT

## BOX MODEL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| About CSS Box Model | It is a box that wraps around every HTML element and consists of (from outside to inside) margins, borders, padding and the actual content. |  | | |
| Inline Elements | Don't start on a new line, margin can be added just on the right and left sides, the top and bottom padding is ignored when positioning the element, width ant height are not applied: a, label, map, span, strong, em, i, img, textarea, input, button, select, etc. Inline elements cannot contain block elements. | | | |
| .parent { width: 350px; height: 200px; }  .child {  width: 80px; /\* not applied \*/  margin: 5px;  padding: 15px; /\* ignored on the top \*/  } | |  | |
| Inline-Block Elements | Don't start on a new line, but can have padding and margin added on all four sides. Declaration in CSS code: display: inline-block;. | | | |
| .child {  display: inline-block;  width: 80px; /\* applied \*/  margin: 5px;  padding: 15px; /\* considered \*/  } | |  | |
| Block Elements | Start on a new line and fill up the full width available: main, header, article, section, fieldset, nav, ul, ol, li, form, h1-h6, p, div, etc. | | | |
| .child {  display: block;  width: 80px; /\* applied \*/  margin: 5px; /\* ignored on the top \*/  padding: 15px; /\* considered \*/  } | |  | |
| Hide an Element | .second-child {  display: none;  } | |  | |
| .second-child {  visibility: hidden;  /\* the element still takes the space \*/  } | |  | |
| Width and Height Dimensions (NOT Applied to Inline Elements) | width: auto; /\* adjusts to display the content correctly \*/  width: 240px | 1(r)em;  width: 50%; /\* 50% of the container's width \*/  width: calc(100% - 20px); /\* 100% of the container's width minus 20px \*/  width: 50vw; /\* 50% of the current screen size \*/  width: fit-content;  max-width: none | 200px | 50%; /\* the maximum width of the element \*/  min-width: none | 200px | 50%; /\* the minimum width of the element \*/  height: auto | 240px | 1(r)em | 50% | 50vh;  max-height: none | 200px | 50%; /\* the maximum width of the element \*/  min-height: none | 200px | 50%; /\* the minimum width of the element \*/ | | | |
| Overflow (Only for Block Elements with Height) | overflow: visible; /\* default \*/ | | |  |
| overflow: hidden; | | |  |
| overflow: scroll; /\* adds a scrollbar \*/  overflow: auto; /\* adds a scrollbar wher needed \*/  overflow-y: scroll; /\* adds a vertical scrollbar \*/ | | |  |
| overflow: hidden;  white-space: nowrap; /\* text stays on the same line \*/  (text-overflow: clip;) /\* default \*/ | | |  |
| overflow: hidden;  white-space: nowrap; /\* text stays on the same line \*/  text-overflow: ellipsis; | | |  |
|  |  | | | |
| Margins (the Space Wrapping the Border) | margin-top: 10px | 20% | auto; /\* auto centers the element horizontally \*/  margin-right: 20px;  margin-bottom: 30px; /\* collapses with the top margin of the element below \*/  margin-left: 40px;  margin: 10px 20px 30px 40px; /\* shorthand property \*/  margin: 10px 20px 30px; /\* top: 10px, left & right: 20px, bottom: 30px \*/  margin: 10px 20px; /\* top & bottom: 10px, left & right: 20px \*/ | | | |
| Paddings (the Space between the Content and the Border) | padding-top: 10px;  padding-right: 20px;  padding-bottom: 30px;  padding-left: 40px;  padding: 10px 20px 30px 40px; /\* shorthand property \*/  padding: 10px 20px; /\* top & bottom: 10px, left & right: 20px \*/ | | | |
| Border |  | | | |
|  |  | | | |
| Outline (outside Border, NOT Calculated in Dimensions) | outline-width: 4px | thin | medium | thick; /\* the width of the element's outline \*/  outline-style: dotted| dashed | solid | ...; /\* the style of the element's outline \*/  outline-color: 4px; /\* the color of the element's outline \*/  outline: 4px dotted red; /\* width, style, color \*/  outline-offset: 15px; /\* the space between the outline and the border \*/ | | | |
| Set the Width/Height Calculation | box-sizing: content-box; /\* default value, width and height only include content \*/  box-sizing: border-box; /\* size includes content, border, paddings and margins \*/ | | | |
| Set the Sizing for the Whole Document | html { box-sizing: border-box; }  \*, \*::before, \*::after { box-sizing: inherit; } | | | |

## POSITION & FLOAT

|  |  |  |  |
| --- | --- | --- | --- |
| Make the Rest of a Container Wrap around a Given Child | float: right; /\* floats the element to the right of its container \*/  float: left; /\* floats the element to the left of its container \*/  float: none; /\* default; restricts the element to float \*/  float: initial; /\* the element remains in its default position \*/  float: inherit; /\* the element inherits the property from its parent \*/ | | |
| Specify on Which Side of an Element Floating Elements are NOT Allowed to Float | clear: none; /\* other elements are allowed to float to the left and to the right \*/  clear: left; /\* other elements are NOT allowed to the left \*/  clear: right; /\* other elements are NOT allowed to the right \*/  clear: both; /\* other elements are NOT allowed to float around \*/ | | |
| The Clearfix Hack | .parent { overflow: auto; } /\* when a floated element is taller than its parent \*/  .parent::after { content: ''; clear: both; display: table; } /\* same as above \*/ | | |
| Specify the Type of Positioning Method Used for an Element | position: static; /\* default; element is in its normal position in the layout flow \*/  position: relative; /\* looks like static (in natural flow) but is relative to parent \*/  position: absolute; /\* relative to closest positioned ancestor/root (NOT in flow) \*/  position: fixed; /\* positioned according to the viewport \*/  position: sticky; /\* relative until a given position is met, then fixed \*/ | | |
| Positioning Properties (Don't Work with Static Position!) | top: 20px; /\* relative: 20 px down from original position;  absolute: 20px down from the parent's (closest positioned element's) top \*/  left: -100px; /\* relative: 100 px left from original position;  absolute: 100px left from the parent's left border \*/  top: 50%; left: 50%; transform: translate(-50%, -50%); /\* center (absolute) \*/  z-index: 1; /\* behind the elements with z-index 2, 3, etc.; position: absolute! \*/ | | |
| Tooltip | <div>  Hover over me  <span>Tooltip text</span>  </div> | div { position: relative; }  span {  position: absolute;  bottom: 130%;  left: 0;  } |  |
| Tooltip Arrow | span::after {  position: absolute;  top: 100%;  left: calc(50% - 5px);  border: 5px solid;  border-color: gray transparent transparent transparent;  } | |  |

## FLEFBOX

### ABOUT FLEXBOX

|  |  |
| --- | --- |
| About Flexbox | It is a method for laying out rows and columns that offers space distribution between items in an interface and powerful alignment capabilities. Flexbox expands items to fill available free space or shrinks them to prevent overflow. |
| Advantages of Flexbox | With flexbox, we can vertically center an element inside its parent, make all the children of a container take up an equal amount of the available width/height, make all the columns in a multiple-column layout adopt the same height even if they contain a different amout of content, etc. |

### PROPERTIES FOR THE PARENT

|  |  |
| --- | --- |
| Create a Flexbox Container | display: flex; /\* the child elements are turned into flexbox items \*/  display: inline-flex; /\* container is displayed inline, its children ‒ flexbox items \*/ |
| Specify Flex Direction | flex-direction: row; /\* default; flexbox items are ordered in the text direction \*/  flex-direction: row-reverse; /\* the opposite way of the text direction \*/  flex-direction: row; /\* the flexbox items are ordered like the text direction \*/  flex-direction: column; /\* the same way as the text direction (cross axis) \*/  flex-direction: column-reverse; /\* the text direction (cross axis) reversed \*/ |
| Specify the Number of Lines the Fexbox Items Will Appear on | flex-wrap: nowrap; /\* default, the flexbox items will remain on a single line \*/  flex-wrap: wrap; /\* will be distributed among multiple lines if needed \*/  flex-wrap: wrap-reverse; /\* multiple lines if needed, next line before previous \*/ |
| Specify Flex Direction and Number of Lines | flex-flow: row wrap; /\* default; shorthand for flex-direction and flex-wrap \*/ |
| Horizontal Alignment of Items within Flexbox Container | justify-content: flex-start; /\* pushed towards the start of the container \*/  justify-content: flex-end; /\* pushed towards the end of the container \*/  justify-content: center; /\* centered along the container's main axis \*/  justify-content: space-between; /\* remaining space distributed between items \*/  justify-content: space-around; /\* remaining space distributed around items \*/ |
| Vertical Alignment of Items within Flexbox Container | align-items: stretch; /\* default; will stretch across the whole cross axis \*/  align-items: flex-start; /\* aligned at the start of the cross axis \*/  align-items: flex-end; /\* aligned at the end of the cross axis \*/  align-items: center; /\* aligned at the center of the cross axis \*/  align-items: baseline; /\* according to the baseline of the content \*/ |
| Vertical Alignment of Lines (when Multiple Lines) | align-content: stretch; /\* each line will stretch to fill the remaining space \*/  align-content: flex-start; /\* only needed space from the start of container \*/  align-content: flex-end; /\* only needed space to the end of container (down) \*/  align-content: center; /\* only needed space at the center of container \*/  align-content: space-between; /\* remaining space appears between the lines \*/  align-content: space-around; /\* remaining space distributed around the lines \*/ |

### PROPERTIES FOR THE CHILDREN

|  |  |
| --- | --- |
| Specify the Order of a Flexbox Item | order: 0; /\* will appear after item with order: -7 and before item with order: 9 \*/ |
| Specify Item Growth When Space Available | flex-grow: 0; /\* item will NOT grow, it will only fill the space it needs \*/  flex-grow: 1; /\* will fill up remaining space if the only item with flex-grow \*/ |
| Specify How Much an Item Should Shrink if not Enough Space | flex-shrink: 0; /\* will NOT shink and will NOT wrap its content \*/  flex-shrink: 1; /\* default; will shink by a factor of 1 and will wrap its content \*/ |
| Speify the Initial Size of a Flexbox Item (Main Axis) | flex-basis: auto; /\* automatically sized based on its content or width/height \*/  flex-basis: 80px; /\* will wrap its content to avoid overflow \*/  flex-basis: 100%; /\* = width: 100% when flex-direction: row \*/ |
| Specify Initial Size and Size Changes | flex: 0 1 auto; /\* default; flex-grow: 0, flex-shrink: 1, flex-basis: auto \*/ |
| Align a Single Item | align-self: auto; /\* will use the value of align-items \*/  align-self: flex-start; /\* aligns the item at the start of the container's cross axis \*/  align-self: flex-end; /\* aligns the item at the end of the container's cross axis \*/  align-self: center; /\* aligns the item at the center of the container's cross axis \*/  align-self: stretch; /\* will stretch across the container's whole cross axis \*/ |

## CSS LAYOUT EXAMPLES

### HORIZONTAL ALIGNMENT

|  |  |  |
| --- | --- | --- |
| Center One or More Inline or Inline-Block Element Horizontally inside Parent | .parent {  text-align: center;  } |  |
| Center One Block Element Horizontally inside Parent | margin: 0 auto; |  |
| Center Several Flex Items Horizontally inside Parent | .parent {  display: flex;  justify-content: center;  } |  |
| Stretch an Inline-Block or Block Element to Take the Whole Width of the Parent | width: 100%; |  |
| Stretch a Flex Item to Take the Whole Width of the Parent | width | flex-basis: 100%;  or  flex-grow: 1;  /\* + align-items: flex-start; to prevent stretching \*/ |  |
| Distribute Space between Inline-Block or Flex Items 50/50 | <div>1</div><div>2</div>  <!-- inline-block: no whitespace/new line between the HTML elements! -->  .first, .second { width: 50%; } |  |
| Distribute Space between Flex Items 50/50 | .parent {  display: flex; }  .first, .second {  flex-grow: 1; } |  |
| Distribute Space Evenly between 3 Inline-Block or Flex Items | <div>1</div><div>2</div><div>...  <!-- inline-block: no whitespace/new line between the HTML elements! -->  .first, .second, .third {  width: 33.333%; } |  |
| Distribute Space Evenly between 3 Block or Flex Items | <div>1</div><div>2</div><div>...  <!-- block: no whitespace/new line between the HTML elements! -->  .first, .second, .third {  width: 33.333%; float: left; } |  |
| Distribute Space Evenly between 3 or More Flex Items | .parent {  display: flex; }  .first, .second, .third {  flex-grow: 1; } |  |
| Distribute Space between Inline-Block or Flex Items 70/30 | <div>1</div><div>2</div>  <!-- inline-block: no whitespace/new line between the HTML elements! -->  .first { width: 70%; }  .second { width: 30%; } |  |
| Distribute Space between Flex Items 70/30 when Enough Space | .parent { display: flex; }  .first { flex-grow: 3; }  .second { flex-grow: 1; } |  |
| Distribute Space between Flex Items 70/30 when NOT Enough Space | .parent { display: flex; }  .first { flex-shrink: 0; }  .second { flex-shrink: 3; } |  |
| Position Inline or Inline-Block Elements with All Available Empty Space in between | .parent { position: relative; }  .first { position: absolute; left: 0; }  .first { position: absolute; right: 0; } |  |
| Position Block Elements with All Available Empty Space in between | .parent {  column-count: 2;  column-gap: calc(100% - 160px);  }  .child { width: 80px; } |  |
| Position Flex Items with All Available Empty Space in between | justify-content: space-between;  /\* + align-items: flex-start; to prevent stretching \*/ |  |
| Position Inline or Inline-Block Elements with All Available Empty Space Equally Distributed around Them | .parent { position: relative; }  .first { position: absolute; left: 25%;  transform: translate(-50%, 0) }  .second { position: absolute;  right: 25%;  transform: translate(50%, 0) } |  |
| Position Block Elements with All Available Empty Space Equally Distributed around Them | .parent { column-count: 2; }  .child {  width: 80px;  margin: 0 auto;  } |  |
| Position Flex Items with All Available Empty Space Distributed around Them | justify-content: space-around;  /\* + align-items: flex-start; to prevent stretching \*/ |  |

### VERTICAL ALIGNMENT

|  |  |  |
| --- | --- | --- |
| Position Inline-Block Elements Vertically next to Each Other | .first { vertical-align: bottom; }  .second, .third {  vertical-align: middle; }  .fourth { vertical-align: top; } |  |
| Position Flex Items Vertically next to Each Other | .parent { display: flex; }  .first { align-self: flex-end; }  .second { align-self: stretch; }  .third { align-self: center; }  .fourth { align-self: flex-start; } |  |
| Center Any Element Vertically inside Parent | .parent { position: relative; }  .child { position: absolute; top: 50%;  transform: translate(0%, -50%);  margin-top: 0; } |  |
| Center a Flex Item Vertically inside Parent with a Parent Property | .parent {  display: flex;  align-items: center;  } /\* parent must have height! \*/ |  |
| Center a Flex Item Vertically inside Parent with a Child Property | .parent {  display: flex; }  .child {  align-self: center; } |  |
| Stretch Inline-Block and Block Elements Vertically | .first, .second {  height: 100%;  } |  |
| Stretch Flex Elements Vertically | .parent {  display: flex;  /\* align-items: stretch; (default) \*/  } |  |
| Display Rows of Flex Items at the Top of the Parent | .parent {  align-content: flex-start;  } |  |
| Display Rows of Flex Items at the Bottom of the Parent | .parent {  align-content: flex-end;  } |  |
| Display Rows of Flex Items in the Middle of the Parent | .parent {  align-content: center;  } |  |
| Display All Available Vertical Space between the Rows of Flex Items | .parent {  align-content: space-between;  } |  |
| Distribute Evenly the Available Vertical Space between the Rows of Flex Items | .parent {  align-content: space-around;  } |  |
| Stretch the Rows of Flex Items to Fill the Whole Vertical Space | .parent {  align-content: stretch;  } |  |

### HORIZONTAL AND VERTICAL ALIGNMENT

|  |  |  |
| --- | --- | --- |
| Center Any Element Horizontally and Vertically inside Parent | .parent { position: relative; }  .child {  position: absolute;  top: 50%;  left: 50%;  transform: translate(-50%, -50%);  margin: 0;  } |  |
| Center a Flex Item Horizontally and Vertically inside Parent | .parent {  display: flex;  justify-content: center;  align-items: center;  } /\* parent must have height! \*/ |  |

### DISPLAYING IMAGES

|  |  |  |
| --- | --- | --- |
| Make a Large Image Scalable | <article><img></article>  img { width: 100%; height: 100%; } |  |
|  |  |  |

### DROPDOWN MENUS

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

# RESPONSIVE WEB DESIGN

|  |  |
| --- | --- |
| About Responsive Web Design | Responsive web design is using HTML and CSS to automatically resize, hide, shrink or enlarge a website to make it look good on all devices. |
| Set the Viewport in HTML | <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">  <!-- gives the browser instructions on how to control the page's dimensions and scaling --> |

## MEDIA QUERIES

|  |  |
| --- | --- |
| About Media Queries | A CSS technique which uses the @media rule to include a block of CSS properties only if a certain condition is true. Good practice is to initially style the page for mobile phones and then add media queries for larger screens (AFTER the original CSS declaration). |
| Use Media Queries | @media only screen and (min-width: 600px) {  p { color: red;} /\* only applies when screen width > 600 px \*/  }  @media print { p { color: red;} } /\* only when printed \*/ |
| Media Types | all | print | screen | speech |
| Media Feature Rules | (min/max-)width | height | orientation | (min/max-)aspect-ratio |
| Logical Operators | and /\* combines multiple media features \*/  not /\* negates a media query \*/  only /\* applies a style only if an entire query matches \*/  , (comma) /\* combines multiple media queries into a single rule \*/ |
| Typical Device Breakpoints | @media screen and (max-width: 600px) { ... } /\* extra small devices \*/  @media screen and (min-width: 600px) { ... } /\* small devices \*/  @media screen and (min-width: 768px) { ... } /\* medium devices \*/  @media screen and (min-width: 992px) { ... } /\* large devices \*/  @media screen and (min-width: 1200px) { ... } /\* extra large devices \*/ |
| Media-Dependent @import Rules | @import url("finerprint.css") print;  @import url("landscape.css") screen and (orientation: landscape); |

# CSS ARCHITECTURE

## ITCSS LAYERS

|  |  |
| --- | --- |
| About ITCSS | Inverted Triangle CSS is a methodology that helps developers organize their project CSS files in layers (subfolders) in order to better deal with CSS specifics like global namespace, cascade and selectors specifity. |
| Layer Architecture in the Project Folder | Create a folder "styles" in the project folder. It should contain the following subfolders: 01-settings, 02-tools, 03-generic, 04-elements, 05-objects, 06-components, 07-utilities. |
| Import Stylesheets into HTML | When using pure CSS, import all CSS files (in the given order) into the HTML page, each in its own <link> tag, or import all CSS files into a styles/style.css file using @import url('./...') and then import only this file in the HTML. |
| Layers Settings & Tools | Used with preprocessors, no actual CSS code generated. |
| Generic Layer | 03-generic/generic-normalize.css: /\* tag selectors \*/  github.com/sindresorhus/modern-normalize/modern-normalize.css  /\* resets browser settings for HTML elements (styles, box-sizing definition, etc.) \*/  03-generic/generic-fonts.css:  @import url('https://fonts.googleapis.com/css2?family=Lato&display=swap');  03-generic/generic-scrollbar.css:  ::-webkit-scrollbar { width: 8px; }  ::-webkit-scrollbar-track { box-shadow: inset 0 0 5px red; border-radius: 8px; }  ::-webkit-scrollbar-thumb { background-color: red; border-radius: 8px; }  03-generic/generic-selection.css:  ::selection { color: white; background-color: tomato; } |
| Elements Layer | 04-elements/elements-body.css: /\* tag selectors \*/  body { min-height: 100vh; display: flex; flex-flow: column wrap; }  04-elements/elements-heading.css:  h1, h2, h3, h4, h5 { font-family: Georgia, serif; }  h1 { font-size: 36px; line-height: 43px; }  04-elements/elements-links.css:  a { color: black; text-decoration: none; }  a:hover { text-decoration: underline; } |
| Objects Layer | 05-objects/objects-containers.css: /\* class selectors for abstract objects \*/  .o-centered-container { display: flex; flex-flow: column wrap; }  .o-row-container { display: flex; flex-flow: row wrap; } |
| Components Layer | 06-components/components-site-header.css: /\* class (id) selectors \*/  .c-site-header { border: 2px solid gray; border-radius: 8px; padding: 10px; }  06-components/components-card.css:  .c-card { border: 2px solid gray; border-radius: 8px; padding: 10px; }  .c-card p { margin: 0; }  .c-card h1 { margin-top: 0; }  06-components/components-thumbnail.css:  .c-thumbnail { width: 40px; height: 40px; }  .c-thumbnail--circle { border-radius: 50%; }  .c-thumbnail--big { width: 100px; height: 100px; } |
| Utilities Layer | 07-utilities/utilities-hidden.css: /\* class (id) selectors \*/  .u-visually-hidden { /\* hidden, but read by reading software \*/  position: absolute;  padding: 0 **!important**;  border: 0 **!important**;  width: 1px **!important**;  height: 1px **!important**;  overflow: hidden;  } |

## BEM NAMING CONVENTION

|  |  |
| --- | --- |
| About BEM | Block, Element, Modifier is a popular naming convention for classes in HTML and CSS. It helps developers experience less problems from cascading and also be able to transfer blocks (components) from finished projects to new ones.  It can be perfectly combined with the ITCSS architecture, playing a key role in the sixth layer, components. |
| Parent Element | .btn |
| Child Element | .btn\_\_price |
| Modifier | .btn--big |
| Example | <a *class*="btn btn--big btn--orange" *href*="#"></a> |

## OBJECT ORIENTED CSS

|  |  |  |
| --- | --- | --- |
| Purpose and Principles | The purpose of OOCSS is to encourage code reuse and, ultimately, faster and more efficient stylesheets that are easier to add and to maintain. It is based on two main principles: separation of structure from skin and separation of containers and content. | |
| Separation of Structure from Skin | It is the abstraction of repeated visual features ("skins") into class-based modules, so that they become reusable. | |
| CSS | OOCSS |
| #button {  width: 200px;  height: 50px;  border: solid 1px #ccc;  background-color: lightgray;  }  #box {  width: 400px;  overflow: hidden;  border: solid 1px #ccc;  background-color: lightgray;  } | .button {  width: 200px;  height: 50px;  }  .box {  width: 400px;  overflow: hidden;  }  .skin {  border: solid 1px #ccc;  background-color: lightgray;  } |
| Separation of Containers and Content | It is the code writing that ensures the styles are not dependent on any containing element and can be reused anywhere: avoid selectors like *.sidebar h3* or *h1.title*). | |

## CSS MODULES

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

## CSS STYLED COMPONENTS

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