CSS

* Cascading style sheets

CSS Syntax



Css selectors:

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

https://youtu.be/m\_upHrxjR8s?si=NH1gjBcvM6NchakU)

There are three ways of inserting a style sheet:

* External CSS
* Internal CSS
* Inline CSS

Cascading Order

* Inline style (inside an HTML element)
* External and internal style sheets (in the head section)
* Browser default

Properties:

* Comments
* Colors (hex, rgb and hsl)

Background

* Background-color
* Background-image
* Background-repeat (x and y-axis, no repeat)
* Background-attachment (specifies whether the background image should scroll or be fixed)
* Background-position (top right, center etc)
* Background (shorthand property) background: #ffffff url("img\_tree.png") no-repeat right top;
* Background-origin (specifies the location of background)
* Opacity

Css Border-style

The following values are allowed:

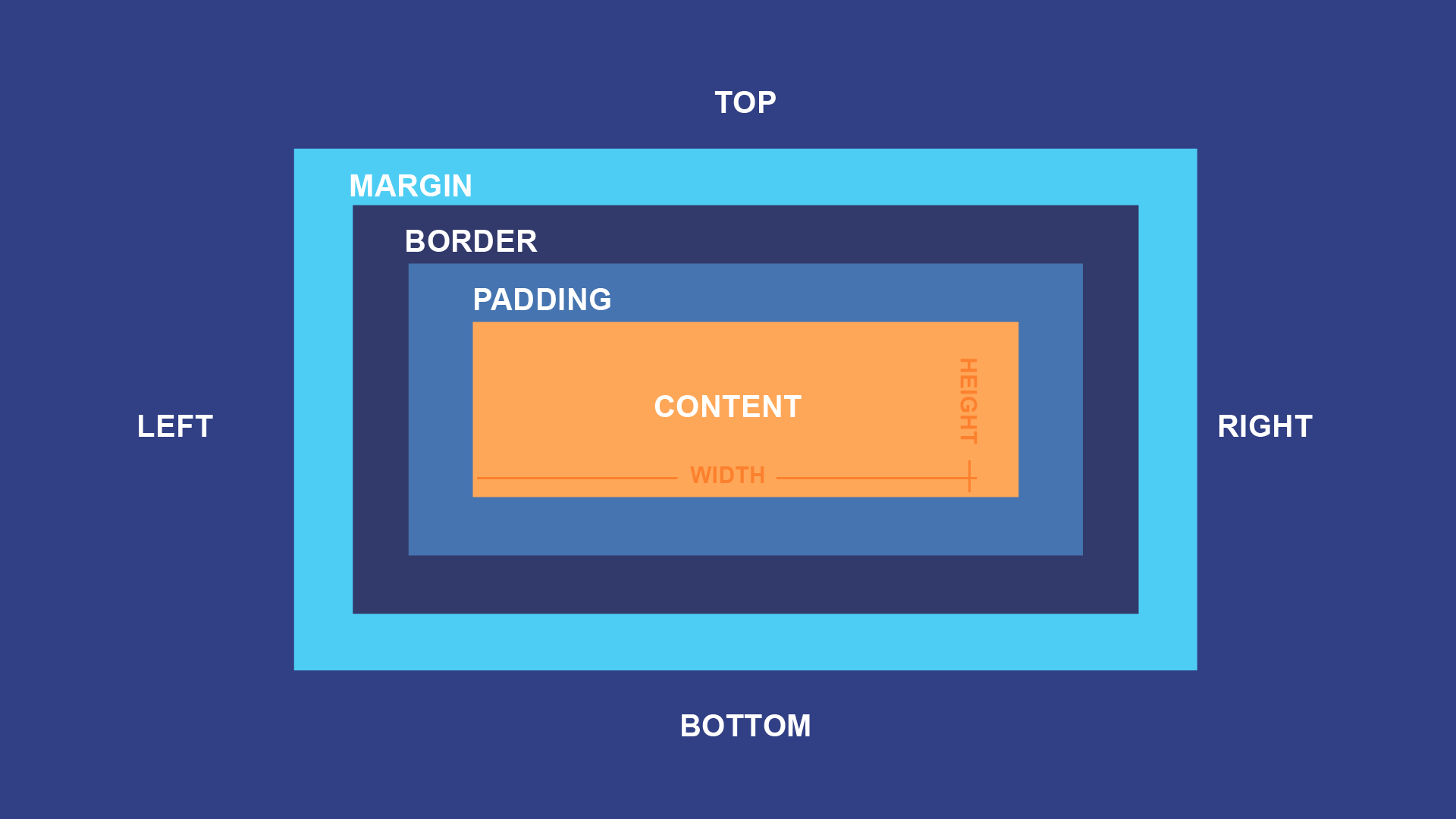
* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border
* border-width
* border-color
* border-shorthand property(border: 5px solid red;)
* border-radius(can be applied to every corner alone)
* border-image (is has same properties as background image)
* CSS margin
* CSS padding

Height and Width

The height and width properties may have the following values:

* auto - This is default. The browser calculates the height and width
* length - Defines the height/width in px, cm, etc.
* % - Defines the height/width in percent of the containing block
* initial - Sets the height/width to its default value
* inherit - The height/width will be inherited from its parent value
* **Max-width and Min-width**

Box Model



* Outline- outline in the outer layer of margin and it has similar properties to border style .

Css Text

* **Color**
* **Alignment**
* **Decoration**
* **Text-transform**
* **Letter spacing and word spacing**
* Text shadow
* Font size
* Font family
* Font style {italic ,bold and oblique(leaning text)}
* **Font weight**
* **Font variant ( it includes properties like small caps,not capitalized etc.)**
* **Using Google Fonts (you can use multiple google fonts)**
* **Font-shorthand (**  font: italic small-caps bold 12px/30px Georgia, serif;**)**
* Add Icons using font awesome

Links

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked
* Cursor Property= <https://www.w3schools.com/css/tryit.asp?filename=trycss_cursor>
* Display value(inline block,none etc)
* Visibillity=hidden
* Min and Max width properties

Position:

* Position-absolute
* Position-relative
* Position-sticky
* Position-fixed
* Using position property you can also align items left and right.
* Z-index
* Overflow (scroll, hidden, visible and x,y-axis)
* Float(Left, right)
* Clear( Left ,right and both) If floated element is taller than containing element that the container will overflow to fix this there is a hack named as clear fix and it shown in this link=(<https://www.w3schools.com/css/css_float_clear.asp>)
* Dropdowns can be made using select property
* Attributes can be give style as well like this:

a[target] {

background-color: yellow;

* !Important rule- It cannot be overided in any way.
* CSS Gradient:
* Linear Gradients (goes down/up/left/right/diagonally)
* Radial Gradients (defined by their center)
* Conic Gradients (rotated around a center point)
* Box shadow(just like text shadow)

CSS Text Effect

|  |  |
| --- | --- |
| * [text-justify](https://www.w3schools.com/cssref/css3_pr_text-justify.asp) | * Specifies how justified text should be aligned and spaced |
| * [text-overflow](https://www.w3schools.com/cssref/css3_pr_text-overflow.asp) | * Specifies how overflowed content that is not displayed should be signaled to the user |
| * [word-break](https://www.w3schools.com/cssref/css3_pr_word-break.asp) | * Specifies line breaking rules for non-CJK scripts |
| * [word-wrap](https://www.w3schools.com/cssref/css3_pr_word-wrap.asp) | * Allows long words to be able to be broken and wrap onto the next line |
| * [writing-mode](https://www.w3schools.com/cssref/css3_pr_writing-mode.asp) | * Specifies whether lines of text are laid out horizontally or vertically |

2D and 3d transformations

* <https://www.w3schools.com/css/css3_2dtransforms.asp>
* <https://www.w3schools.com/css/css3_3dtransforms.asp>

Flexbox:

* Flex-Direction(column, row,column reverse etc)
* Justify Content(used to align items on main axis)
* Align-items(used to align items on the cross axis)
* Flex-wrap (does not squeeze content in one but allows it to aquire another line, it also unlocks another property =align-content it only works when wrap is enabled and items are wrapping)
* Align-items(same properties as justify content)
* Gap(adds gap between items)
* Flex-grow(it is applied to items and the item takes the remaining available space)
* Flex-shrink(the lower the value give to the object the more it prevents to shring the size of screen is reduced)
* Flex-basis(it the size of each item, its default value is auto.)
* Flex shorthand( Flex: flex-grow flex-shrink flex basis) second and third values are optional.
* align self ( only aligns the item that this property is given to)
* order( orders the items according to the given value)

Grid

https://youtu.be/EiNiSFIPIQE?si=t-nwWDvSzuZczKDW

* Grid templete rows and colums(determines how many and of which length the rows and columns are going to be)
* Grid row/column start/end (determines where each grid item will start and where it will end, it determines both rows and colums)
* Grid shorthand= Grid-row(star value/end value) Grid-column(start value/end value). Example = Grid-row:1/2; or Grid-column:3/4 etc.
* Instead of values you can use span(span means where ever you are increse yourself twice=Grid-row: span 2;
* Grid-Area is the shorthand of Grid shorthand (Grid-area= row-start/column-start/row-end/column-end)
* Grid-auto rows= if you gave no space to a row and added it in css then it will automatically take the value of pixels given to this property
* Grid-autoflow-column=if you add a new item in grid it will display in colums same goes for Grid-autoflow-row.
* Repeat-In Grid row and grid colums a value can be give called repeat, it means that (repeat: 4 , 100px ; ) =this tells us that the first value is the amount of times repeated and second value is the size of repeated things.
* MinMax- this property is also given to grid row templete and column templete, it is written as MinMax: 100px-200px; this is telling that the maximum size can be 200 px and when shrinked it should be atlease 100px before disappearing.
* Grid gap- it is written as gap:1rem 2rem; (1 rem is gap between rows and 2 rem is between columns) or gap:1rem;(1 rem gap between both column and rows)
* Justify Items and justify self
* Align Items and align self

Css Combinators

* (space) =Desendent-meaning all of the childer would get the listed property of the container. E.g ( .container p { background-color: yellow;). In this example all the p in the container would get bc yellow
* >= child- Meaning that all the children of the container will get the property listed. The grandchild will not get this property
* +=general sibling- All the sibling of container will get this property. E.g- In body there are 2 div and if we give this property to the first div the listed properties would apply to the second div because there father is body and the 2 divs are siblings
* +=Adjacent sibling- The first sibling that comes after the div it is applied on will get the property. E.g There are 3 div in body if we give this to the 1 div the properties would be applied to the 2 div but not the third one.