*CSS cheat sheet*

Cascading style sheets

Used for designing websites.

Syntax: selector {property : value;} p {color : blue;}

Three ways to add CSS to markup(HTML)

1. Inline CSS: CSS is added directly to elements using style attribute.

<p style=”color:pink”></>

1. Internal CSS: CSS is added inside head tag in <style> tag

<head>

<style

p{color:pink}

</style>

</head>

1. External CSS: a separate file (.css file) is created for writing css and include in html file.

Link css file:

<link ref=”file” rel=”stylesheet”></link>

Inline css takes priority.

Internal or external, whichever is written later in head tag takes precedence.

CSS Selectors

Element selector – using tag names.

Id selector- #id ->id is unique.

Class selector- .class ->multiple elements can have one class.

Group selector- comma separated values of classes, to write same css

We can inspect elements and change and verify styles temporarily.

Fonts in CSS:

Font-family: for the fonts of the text. 2 types: Web-safe fonts- which come installed with OS & web fonts- which we have to include from web. (tip: use google fonts for web fonts).

Font-size: size of text

Line-height: height of a line

Font-weight: sets the thickness/thinness of the characters

Color: sets color of the text

Background-color: sets background-color of the text.

Height/Width: to set height and width of an element.

Border(short-hand): border-width border-style border-color.

Background-image: Sets an image as background.



By-default image repeats itself on x and y axis.

To repeat an image horizontally, use

Background-repeat: x-axis

To repeat an image vertically, use

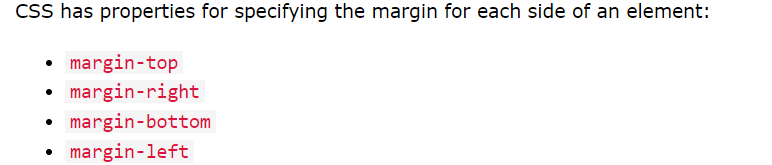
Background-repeat: y-axis

To show image only once, use

Background-repeat: no-repeat

Margin-

Used to create space outside of elements.



Padding-

Used to create space inside the border of an element.

Padding/Margin shortcut:

Padding/Margin: top right bottom letft (tera bottle)

Padding/Margin: top/bottom left/right

Box-Model-

A black text on a white background

Description automatically generated

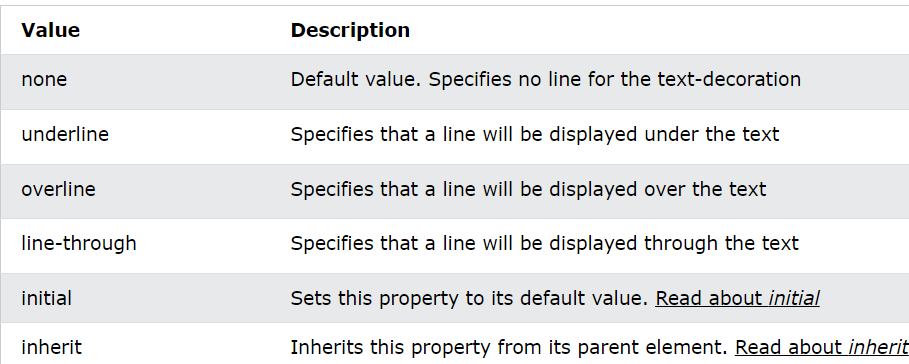
If you set box-sizing as border-box then the padding and border is included in the width & height of the element

STYLING LINKS AND BUTTONS - PSEUDO SELECTORS:

Cursor property: When set to pointer it displays mouse coursor when pointing to an element.

Text-decoration: For adding decoration to the text.

Underline, overline, line-through, none.



Pseudo classes:

To write properties for special state of an element.

When we want to have something when we hover or mouse over an element then use,

selector:hover{

property: value;

}

For visited link , use

a:visited{}

For active link

a:active{}

DISPLAY PROPERTY:

Defines the display of the element

Display: inline

Width and height will have no effect. Like span

The next element will be in same line if there is space

Display:block

Starts with new line. And takes up whole width.

When display is block and width is provided, margin auto will bring the element in center.

Display: inline-block

Aligns the elements inline (in one line)

You can apply width and height of an element like block elements

Display: none

POSITION PROPERTY:

Used for positioning the element.

Position: fixed

Positioned relative to the viewport and remain in same position even if we scroll the page.

We can set the position using top, right, bottom and left

Position: static

It is by default (normal flow), the properties top, right, bottom and left will not be effective.

Position: relative

Positioned relative to its original position, and can be adjusted using (top, right…..) properties

Position: absolute

Positioned relative to its nearest non-static ancestor.

Visibility and z-index

Determines whether the element is visible or hidden

Default value is Visible.

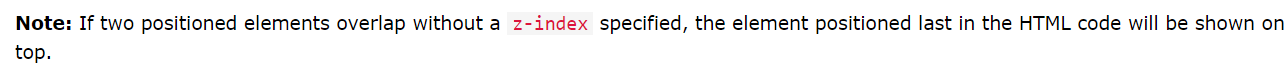
Visibility: hidden , hides the element but shows the empty space.

Use Display: none, to both hide and remove the space.

z-index- specifies stacking order of an element.

The element with higher z-index(stacking order)will always come in front of the the lower z-index.

z-index only works for positioned element (fixed, sticky, relative,absolute).



FLEX-BOX:

Flexible-box

So, there is 2 things, flex-container and flex-items inside it.

To use flex model, we have to define container’s display as flex.

Display: flex (to initialize container as a flex)

Properties of flex-container:

Flex-direction : Specifies in which direction you want to show the items

Flex-direction: row (stacks the items horizontally) default

Flex-direction: column (stacks the items vertically)

Flex-direction: row-reverse /column-reverse

Flex-wrap: specifies whether items should wrap or not.

Flex-wrap: wrap (if needed, items will wrap) when screen-size is smaller

Default is no-wrap

Justify-content: center/space-between/space-evenly/space-around

Used to align items horizontally.

Try different values to align items horizontal direction in different ways

Align-items: center/flex-end/flex-start(default)

To align items vertically.

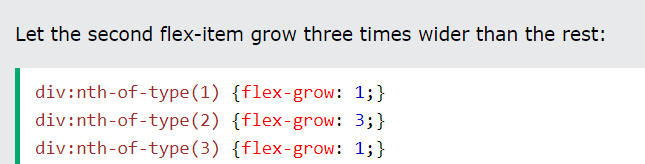
Try different values to align items vertically in different ways.

Flex-properties of the items:

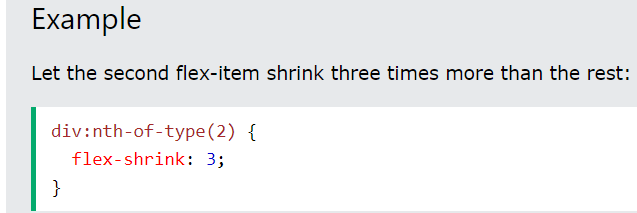
If the container is not flex, then no properties will take effect

Order: default is 0, higher the order of a flex-item, later it shows in the container.

Flex-grow: specifies how much element will grow relative to other elements.

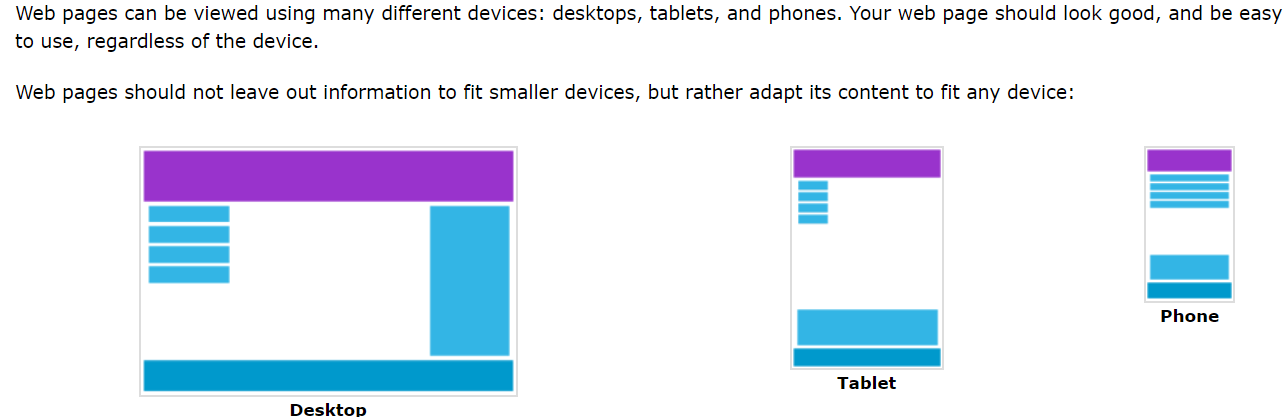


Flex-shrink: how much element will shrink relative to other elements



RESPONSIVE DESIGN

It means that your website should look good on any device.



Size Units:

Rem: relative to font-size of root element.

Vh: relative to height of viewport, 100vh (entire height of the viewport)

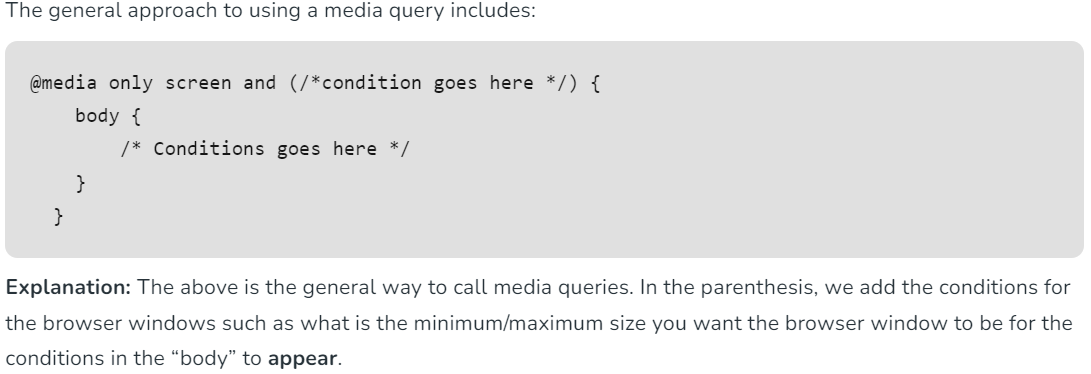
Vw: relative to width of viewport, 100vw (entire width of viewport)

MEDIA-QUERIES:

Used for responsive web design.

It means view of page differs from screen to screen.

It includes a condition, and the design will show only when conditions are met.



For ex:

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

Body {background-color: black;}

}

Background-color will only be applied if screen size is 0 to 500px.

Similarly for screen size 300px to 500px

@media only screen and (min-width:300px ) and (max-width:500px) {

*// CSS*

}

CSS Selectors:

We have seen simple selectors, selecting elements through id, class, and tag.

We have pseudo class selectors, selecting on the specific state of an element.

We can use few these combinations selectors,

1. Descendant Selector (space)

Div p {

// CSS

}

This will select all descendant <p> (no matter how much depth) inside <div> element.

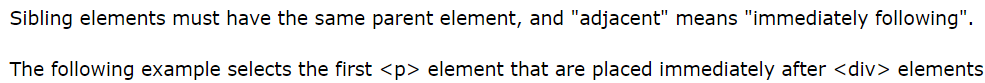
1. Child selector (>)

Div > p {// CSS}

This will CSS to all <p> which are direct child of div.

1. Adjacent Sibling selector (+)

Div + p {//CSS} // p is right after div i.e. next sibling of div



1. Attribute selector

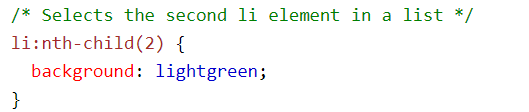
It is used to select elements with specific attribute.

Element[attribute=”value”]

a[target=’\_blank’] selects all element with target equal to \_blank

This setting attribute target to ‘\_blank’ opens link in new tab

Nth child selector



Matches every element which is nth child of its parent

n can be number or keywords like odd or even.

BEFORE AND AFTER PSEUDO SELECTORS