\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. What is CSS?

CSS is a language that describes the style of an HTML document.

CSS describes how HTML elements should be displayed.

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to **describe the content** of a web page, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

1. Can you add numbers to id attribute or class attribute?

No. We cant add number attribute to id or class.

Id=”2” or class=”2”

#2{

Color: red // this wont work

}

1. I have class attached to two separate elements like h1 and p. Now I want to change p only without affecting h1. How do you do that?

Ex:

<h1 class=”center”>hey</h1>

<p class=”center”>Hello</p>

We can achieve it by

p.center{

color: red;

}

1. I have two classes on single element. Both these classes has same property. So which property will element take?

<p class=”one two”>Hey</p>

.one {

Color: red;

}

.two {

Color: yellow

}

The element will always take latest and override its previous value. So here the element will take color yellow.

1. Suppose there are three elements which have same color. How do you write css for these.

We can achieve it by grouping.

H1, p, span{

Color: red;

}

1. Is background image repeated by default?

Yes. It is. We can override this by specifying no-repeat.

By default they repeat in both x and y axis. We can repeat them horizontally or vertically by specifying.

Background-repeat: repeat-x / repeat-y;

To make background fixed all the time even when you scroll, use

Background-attachment: fixed;

To change position , use

Background-position: top left;

Background shorthand

Background: color image repeat attachment position;

1. How will you specify background color should extend only content box not till border?

To do this we have something called background-clip

If content-box -> till content box;

If padding-box -> till padding box;

Border-box -> till border;

1. What values will it take in background-size?

We can specify it as background-size: 30px 20px; (height, width)

If only one value is provided, then first value will be width and height becomes auto.

1. How to specify background to start from specific place?

We have something called background-origin padding-box/content-box/margin-box

1. What is margin collapse?

Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.

This does not happen on left and right margins! Only top and bottom margins!

h1 {  
  margin: 0 0 50px 0;  
}  
  
h2 {  
  margin: 20px 0 0 0;  
}

In the example above, the <h1> element has a bottom margin of 50px and the <h2> element has a top margin set to 20px.

Common sense would seem to suggest that the vertical margin between the <h1> and the <h2> would be a total of 70px (50px + 20px). But due to margin collapse, the actual margin ends up being 50px.

1. Explain Margin and padding?

Padding is used to generate space between elements content.

Margin is used to generate space between elements.

If the padding property has three values:

* **padding: 25px 50px 75px;**
  + top padding is 25px
  + right and left paddings are 50px
  + bottom padding is 75px

1. What is box-sizing?

The CSS width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element ([the box model](https://www.w3schools.com/css/css_boxmodel.asp)).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

In the following example, the <div> element is given a width of 300px. However, the actual rendered width of the <div> element will be 350px (300px + 25px of left padding + 25px of right padding):

div {  
  width: 300px;  
  padding: 25px;

box-sizing: border-box;  
}

To keep the width at 300px, no matter the amount of padding, you can use the box-sizing property. This causes the element to maintain its width; if you increase the padding, the available content space will decrease.

The CSS box-sizing property allows us to include the padding and border in an element's total width and height.

By default, the width and height of an element is calculated like this:

width + padding + border = actual width of an element  
height + padding + border = actual height of an element

This means: When you set the width/height of an element, the element often appears bigger than you have set (because the element's border and padding are added to the element's specified width/height).

1. How can you override width property?

We can override width property by specifying max-width. Max width tells element that this is my max width and you cant go beyond that.

1. Tell me something about box-model ?

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent
* In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.
* **Important:** When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the full size of an element, you must also add padding, borders and margins
* div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;   
  }
* Here is the calculation:
* 320px (width)  
  + 20px (left + right padding)  
  + 10px (left + right border)  
  + 0px (left + right margin)  
  **= 350px**
* The total width of an element should be calculated like this:
* Total element width = width + left padding + right padding + left border + right border + left margin + right margin
* The total height of an element should be calculated like this:
* Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

1. What is outline in CSS?

Outline is a line drawn OUTSIDE border to make the element standout.

Usage:

Outline: 1px solid red;

1. What is outline-offset ?

The outline-offset property adds space between an outline and the edge/border of an element. The space between an element and its outline is transparent.

Usage:

outline-offset: 15px;

1. Tell me some text properties in css ?

We have

Color

Text-align: right/center/left

The text-decoration property is used to set or remove decorations from text.

The value text-decoration: none; is often used to remove underlines from links:

text-decoration: overline;

text-decoration: underline;

text-decoration: line-through;

text-decoration: none;

1. How do you write such that every line has equal width we see in papers?

When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):

1. How do you give some small gap at start of a paragraph?

We have something called text-indent: value to provide gap at start of paragraph.

1. How do you increase or decrease spacing between letters ?

We can use letter-spacing property which takes both positive and negative values.

The line-height property is used to specify the space between lines:

If line-height is 0 all the lines will merge

The direction property is used to change the text direction of an element:

The word-spacing property is used to specify the space between the words in a text.

The text-shadow property adds shadow to text.

The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red):

The white-space property specifies how white-space inside an element is handled.

It takes 3 values

Nowrap: -> single line

Normal -> normal scenario.

Pre -> line by line (if you write content in line by line)

The vertical-align property sets the vertical alignment of an element,

Like suppose I have an image between text. Now I can position this image according my text. Ie top of text , bottom of text etc

The vertical-align property sets the vertical alignment of an element.

The text-overflow property specifies how overflowed content that is not displayed should be signaled to the user. It can be clipped, display an ellipsis (...), or display a custom string

In order to use this property we need to set white-space: nowrap & overflow:hidden

Clip: shows content as per width. Rest content cut down. Default behavior.

Ellipsis: shows rest content in … (3 dots)

1. Tell me something about CSS Fonts?

The CSS font properties define the font family, boldness, size, and the style of a text.

In CSS, there are two types of font family names:

* **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
* **font family** - a specific font family (like "Times New Roman" or "Arial")

**Note:** On computer screens, sans-serif fonts are considered easier to read than serif fonts.

The font family of a text is set with the font-family property.

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

**Note**: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

font-family: "Times New Roman", Times, serif;

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

1. What is the order of CSS links

When setting the style for several link states, there are some order rules:

* a:hover MUST come after a:link and a:visited
* a:active MUST come after a:hover

1. How to position list style type markers?

The list-style-position property specifies the position of the list-item markers (bullet points).

"list-style-position: outside;" means that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically. This is default:

"list-style-position: inside;" means that the bullet points will be inside the list item. As it is part of the list item, it will be part of the text and push the text at the start:

1. Tell me something about position property?

The position property specifies the type of positioning method used for an element.

There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling

A "positioned" element is one whose position is anything except static.

An element with position: sticky; is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

1. Tell me something about combinators?

A combinator is something that explains the relationship between the selectors.

There are four different combinators in CSS:

* descendant selector (space)
* child selector (>)
* adjacent sibling selector (+)
* general sibling selector (~)

Usage:

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

<span><p>Paragraph 3 in the div.</p></span>

<p>Paragraph 4 in the div.</p>

</div>

<p>Paragraph 5. Not in a div.</p>

<p>Paragraph 6. Not in a div.</p>

The descendant selector matches all elements that are descendants of a specified element.

Here it selects all the p’s inside div no matter where there are. But they must be inside div.

Above example we have p1,p2,p3,p4 all have bg as yellow.

div p {  
  background-color: yellow;  
}

The child selector selects all elements that are the immediate children of a specified element. This selector selects those are immediate to it. Here p1,p2 ,p4 gets bg as yellow. If there is any wrapper around them the main doesn’t treat it as child.

div > p {  
  background-color: yellow;  
}

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element. Sibling elements must have the same parent element, and "adjacent" means "immediately following".

div + p {  
  background-color: yellow;  
}

if there is any span or any element between them, then it wont apply. The rule is immediate first guy.

The general sibling selector selects all elements that are siblings of a specified element.

div ~ p {  
  background-color: yellow;  
}

here theres no catch it selects all the siblings no matter what.

1. Tell me something about pseudo classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user mouses over it
* Style visited and unvisited links differently
* Style an element when it gets focus

selector:pseudo-class {  
  property:value;  
}

FirstChild:

p:first-child {

color: blue;

}

<p>Text1.</p>

<p>Text2</p>

Here p with text1 will get blue color. If you want to style a element which is first child we can take use of this.

1. So generally quote <q> tag gives “ ” by default. Now instead of “ ” I want to have ~ this around my quotes. How do you achieve this?

<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>

q:lang(no) {

quotes: "~" "~";

}

1. What are pseudo elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

Style the first letter, or line, of an element

Insert content before, or after, the content of an element

selector::pseudo-element {  
  property:value;  
}

::first-line,::first-letter,::before,::after

1. As a user when I select any text on my webpage, that particular text should have different color. How do you achieve this?

We have something called ::selection. Using this we can change color.

::selection{

Color: red;

}

1. What is CSS attribute selector?

The [attribute] selector is used to select elements with a specified attribute.

Usage:

a[target="\_blank"] {   
  background-color: yellow;  
}

The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.

The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":

[title~="flower"] {  
  border: 5px solid yellow;  
}

The [attribute|="value"] selector is used to select elements with the specified attribute starting with the specified value.

The following example selects all elements with a class attribute value that begins with "top":

**Note:** The value has to be a whole word, either alone, like class="top", or followed by a hyphen( - ), like class="top-text"!

[class|="top"] {  
  background: yellow;  
}

The [attribute^="value"] selector is used to select elements whose attribute value begins with a specified value.

The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.

The [attribute\*="value"] selector is used to select elements whose attribute value contains a specified value.

1. What are CSS counters?

CSS counters are like "variables". The variable values can be incremented by CSS rules (which will track how many times they are used).

body {

counter-reset: section;

}

h2::before {

counter-increment: section;

content: "Section " counter(section) ": ";

}

<h1>Using CSS Counters:</h1>

<h2>HTML Tutorial</h2>

<h2>CSS Tutorial</h2>

This will generate

Section 1 HTML Tutorial

Section 2 CSS Tutorial

1. What is CSS specifity?

If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.

## **How to Calculate Specificity?**

Memorize how to calculate specificity!

Start at 0, add 1000 for style attribute, add 100 for each ID, add 10 for each attribute, class or pseudo-class, add 1 for each element name or pseudo-element.

Consider these three code fragments:

A: h1  
B: #content h1  
C: <div id="content"><h1 style="color: #ffffff">Heading</h1></div>

The specificity of A is 1 (one element)  
The specificity of B is 101 (one ID reference and one element)  
The specificity of C is 1000 (inline styling)

Since 1 < 101 < 1000, the third rule (C) has a greater level of specificity, and therefore will be applied.

1. How do you get rounded corners ?

We use border-radius for that.

1. How do put image as border?

With the CSS border-image property, you can set an image to be used as the border around an element.

1. Tell me something about gradient?

CSS gradients let you display smooth transitions between two or more specified colors. To create a linear gradient you must also define at least two color stops.

CSS defines two types of gradients:

* **Linear Gradients (goes down/up/left/right/diagonally)**
* **Radial Gradients (defined by their center)**

background-image: linear-gradient(direction, color-stop1, color-stop2, ...);

default direction is top to bottom.

We can also use angles in place of direction.

repeating-linear-gradient(red, yellow 10%, green 20%);

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

background-image: radial-gradient(red 5%, yellow 15%, green 60%);

starts from center to spread out

1. What is css shadow effect?

With CSS you can add shadow to text and to elements.

In this chapter you will learn about the following properties:

* text-shadow
* box-shadow

text-shadow: 2px 2px 5px red; hor,vertical,blur,color

Multiple text shadows

text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;

box-shadow: 10px 10px 5px grey; -> hor,vrt,blur,color -> for elements

1. Tell me something about css text effects?

* text-overflow
* word-wrap
* word-break
* writing-mode

The CSS text-overflow property specifies how overflowed content that is not displayed should be signaled to the user.

* It can be clipped:  text-overflow: clip;
* or it can be rendered as an ellipsis (...):text-overflow: ellipsis;

The CSS word-wrap property allows long words to be able to be broken and wrap onto the next line.

To break word into next line word-wrap: break-word;

Else use word-wrap: no-wrap;

The CSS word-break property specifies line breaking rules.

  word-break: keep-all; -> words break with ‘-‘ hey-there

 word-break: break-all;

he

y there

The CSS writing-mode property specifies whether lines of text are laid out horizontally or vertically.

writing-mode: horizontal-tb; -> default

writing-mode: vertical-rl; -> words will be in vertical direction

1. How do you use your own web font?

Web fonts allow Web designers to use fonts that are not installed on the user's computer.

When you have found/bought the font you wish to use, just include the font file on your web server, and it will be automatically downloaded to the user when needed.

Your "own" fonts are defined within the CSS @font-face rule.

## Different Font Formats

**TrueType Fonts (TTF)**

**OpenType Fonts (OTF)**

**The Web Open Font Format (WOFF)**

WOFF is a font format for use in web pages. It was developed in 2009, and is now a W3C Recommendation. WOFF is essentially OpenType or TrueType with compression and additional metadata. The goal is to support font distribution from a server to a client over a network with bandwidth constraints.

**The Web Open Font Format (WOFF 2.0)**

TrueType/OpenType font that provides better compression than WOFF 1.0.

**SVG Fonts/Shapes**

**Embedded OpenType Fonts (EOT)**

**Usage:**

@font-face {  
  font-family: myFirstFont;  
  src: url(sansation\_light.woff);  
}  
  
div {  
  font-family: myFirstFont;  
}

1. Tell me something about CSS 2D Transforms?

CSS transforms allow you to translate, rotate, scale, and skew elements.

A transformation is an effect that lets an element change shape, size and position.

CSS supports 2D and 3D transformations.

Transform: ‘any of below value’

* translate()

The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

* rotate()

The rotate() method rotates an element clockwise or counter-clockwise according to a given degree. Using negative values will rotate the element counter-clockwise.

* scale()

The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).

* skewX()

The skewX() method skews an element along the X-axis by the given angle.

* skewY()

The skewY() method skews an element along the Y-axis by the given angle.

* matrix()

The matrix() method combines all the 2D transform methods into one.

The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements.

The parameters are as follow: matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

1. What about 3D transforms?

* rotateX()
* rotateY()
* rotateZ()

transform-style: preserve-3d;

The transform-style property specifies how nested elements are rendered in 3D space.

transform-origin: 20% 40%;

The transform-origin property allows you to change the position of transformed elements.

  perspective: 100px;

The perspective property is used to give a 3D-positioned element some perspective.

The perspective property defines how far the object is away from the user. So, a lower value will result in a more intensive 3D effect than a higher value.

When defining the perspective property for an element, it is the CHILD elements that get the perspective view, NOT the element itself.

perspective-origin: left;

The perspective-origin property defines at from which position the user is looking at the 3D-positioned element.

When defining the perspective-origin property for an element, it is the CHILD elements that will get the effect, NOT the element itself

backface-visibility: visible/ hidden;

The backface-visibility property defines whether or not the back face of an element should be visible when facing the user.

The back face of an element is a mirror image of the front face being displayed.

This property is useful when an element is rotated. It lets you choose if the user should see the back face or not.

This div element has "backface-visibility: visible", and the back face of the div element shows a mirror image of the front face

1. Tell me something about CSS transition ?

CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.

To create a transition effect, you must specify two things:

* the CSS property you want to add an effect to
* the duration of the effect

transition: width 2s;

div:hover {  
  width: 300px;  
}

Multiple transitions

transition: width 2s, height 4s;

transition-property: width;

transition-duration: 2s;

transition-timing-function: linear;

transition-delay: 1s;

or as short hand

 transition: width 2s linear 1s;

1. Tell me something about CSS Animations ?

CSS animations allows animation of most HTML elements without using JavaScript or Flash!

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}  
  
/\* The element to apply the animation to \*/  
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
}

The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

@keyframes example {  
  0%   {background-color: red;}  
  25%  {background-color: yellow;}  
  50%  {background-color: blue;}  
  100% {background-color: green;}

animation-name: example;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;

short hand property:

 animation: example 5s linear 2s infinite alternate;

1. What is object fit property?

The CSS object-fit property is used to specify how an <img> or <video> should be resized to fit its container.

This property tells the content to fill the container in a variety of ways; such as "preserve that aspect ratio" or "stretch up and take up as much space as possible".

By specifying this value, it don’t destroy aspect ratio of image.

1. How to create disable look on buttons ?

By using cursor: not-allowed.

1. How to create multiple columns ?

The column-count property specifies the number of columns an element should be divided into.

The following example will divide the text in the <div> element into 3 columns:

div {  
  column-count: 3;  
}

1. How do you achieve resize in css?

resize: none / horizontal/vertical;

imp: you need to set overflow: auto to have resize effect.

The outline-offset property adds space between an outline and the edge or border of an element.

Outlines differ from borders in three ways:

* An outline is a line drawn around elements, outside the border edge
* An outline does not take up space

  outline: 5px dashed blue;  
  outline-offset: 5px

1. What are CSS variables and how to use them ?

:root {  
  --main-bg-color: coral;   
}  
  
#div1 {  
  background-color: var(--main-bg-color);   
}  
  
#div2 {  
  background-color: var(--main-bg-color);  
}