1. Difference between absolute and relative path.

# Absolute path:

* An absolute path is defined as complete path from start of actual file system from/directory.
* It is also referred to as full path or file path.
* It refers to the location of a file or directory relative to the root directory in Linux.
* Absolute URLs are used to link to other websites that are not located on the same domain.

# Relative path:

* Relative path is defined as the path related to the present working directly.
* It is also referred to as non-absolute path.
* It refers to the location of a file or directory relative to the current directory.
* Relative URLs are used to link to other websites that are located on same domain.

1. Importance of alt and target attributes.

# Alt attribute:

* The required alt attribute specifies an alternate txt for an image, if the image cannot be displayed. The alt attribute provides alternate information for an image if a user for some reasons cannot view it(because of slow internet connection, an error in the src attribute).

# Target attribute:

* With the target attribute, you can define where the linked document will be opened.
* The line below will open the document in a new browser window.

1. Inline-block element.

* Displays an element as an inline-level block container. You can set height and width values.

1. What are forms?

* HTML Form is a document which stores information of a user on a web server using interactive controls.
* An HTML form contains different kind of information such as username, password, contact number, email id etc.
* The elements used in an HTML form are check box, input box, radio buttons, submit buttons etc.

# Types of Forms

* Text Input control
* Checkbox
* Radio box control
* Select box control
* File Select boxes
* Hidden controls
* Clickable buttons
* Submit and Reset buttons

# Form Elements:

* <input>
* <label>
* <select>
* <textarea>
* <button>
* <fieldset>
* <legend>
* <datalist>
* <output>
* <optgroup>

# Form tag attributes:

* Accept-charset
* Action
* Autocomplete
* Enctype
* Method
* Name
* Novalidate
* Rel
* Target

# Form Element attributes

* Name
* Value
* Selected from dropdown
* Disabled
* Type

# Layout concepts (Flex):

The flex container properties are:(w3 schools)

* Flex-direction

The flex-direction property defines in which direction the container wants to stack the flex items.

Example:

1.The column value stacks the flex items vertically (from top to bottom)

Flex-container {

display: flex;

flex direction: column;

}

2.The column-reverse value stacks the flex items vertically (but from bottom to top)

3.The row value stacks the flex items horizontally (from left to right)

4. The row-reverse value stacks the flex items horizontally (but from right to left)

* Flex-wrap Property

The flex-wrap property specifies whether the flex items should wrap or not.

Example:

1.The wrap value specifies that the flex items will wrap if necessary:

. flex-container {

display: flex;

flex-wrap: wrap;

2. The nowrap value specifies that the flex items will not wrap (this is default).

3.The wrap-reverse value specifies that the flexible items will wrap if necessary, in reverse order.

* Flex-flow Property

The flex-flow property is a shorthand property for setting both the flex-direction and flex-wrap properties

Example:

. flex- container {

display: flex;

flex-flow: row wrap;

}

* Justify-content Property

The justify content property is used to align the flex items

Example:

1.center

2.flex start

3.flex end

4.space around.

* Align-items Property:

The align items property is used to align the flex items

1. center
2. Flex starts
3. Flex end
4. Stretch
5. Base line

* Align-content Property

The align content property is used to align flex lines

1. Space between
2. Space around
3. Stretch
4. Center

# HTML5 features

* Nav tag

The <nav> tag helps in defining a set of navigations links. It helps to create a section of website that contains navigation links(within the current document or to another document.)

* Header

The <header> element in HTML5 contains introductory content. It can include a set of introductory or navigational support on a website.

* Footer

The <footer> tag defines the footer of a document or a section. Typically, the footer contains information related to the author, copyright, contact information, sitemap, related documents, back to top links, etc.

* Figure and Fig caption

These elements help user to insert an image with its caption. The figcaption tag used to add a description for the image

* Mark

The<mark> element highlights a particular text that is of special interest to the user in a HTML document.

* Main tag

This tag defines the important content of the <body> of a document.

* Progress tag

The progress tag helps users check the progress of a task during the execution. Users will need to define the dynamically changing values of the progress bar with the scripts.

* Section tag

The <section> element defines specific sections and subsections in the document.

* Article

Used to specify a blog , a magazine or a newspaper article or any other independent piece of content in a document.

* Aside

Used to indicate that the specified article is somehow related to the rest of the document.

* Data

Used to facilitate a machine readable version of the data

* Details

Used to define any additional information on a topic or a summary

* Dialog

Used to specify a window or a dialog box

* Meter

Used to determine a scalar value within a given range.

* Summary

Used to define a visible heading for the HTML<details>element.

# HTML5 Form tags

* Data list

Used to facilitate an auto complete feature for textfiels

* Output

Used to specify the output of a calculation or an outcome of the user action.

# Graphics tags:

* Canvas tag

The canvas tag in HTML5 helps user draw graphics or images on the fly using JavaScript. We can use it for drawing paths, boxes, circles, adding images, etc.

* Svg

Used to display shapes.

# HTML5 Media Tags:

* Audio & video tags

Used to define an audio file in HTML

Used to specify a video file in HTML

* Embed

Used to specify a container for an external file, application or a media

* Source

Used to specify multiple media resources for a media element.

* Track

Used to specify the text tracks for an <audio> and <video> element.

# HTML5 new <input> types:

* Color

Used to define an input field to indicate a color selector

* Date

Used to define an input field to indicate a date selector.

* Datetime

Used to display date and time along with the time zone information.

* Email

Used to specify an input field with an email pattern validation property.

* Month

Used to specify an input field to enter month for the particular year.

* Number

Used to specify a field that accepts a numeric value only.

* Range

Used to create a numeric value selector for a range of 1 to 100.

* Search

Used to create a search field.

* Tel

Used to define a control to enter a telephone number.

* Time

Used to define a date or time.

* url

Used to define an input field to enter a URL.

* Week

Used to create a week value selector for a particular year.

* Required attribute

When present, it specifies that an input field must be filled out before submitting the form.

* Min length

The min length attribute specifies the minimum number of characters required in an input field

* Max length

The max length attribute specifies the maximum number of characters allowed in the <input> element

* Pattern

The pattern attribute specifies a regular expression that the <input> elements value is checked against on form submission.

* Auto focus

When present, it specifies that the element should automatically get focus when the page loads.

# Basics of CSS:

The three main components of CSS rule is:

1. Selector: HTML element name, id name, class name.
2. Property: Its like an attribute such as background color, font-size, position, text-align, border etc.
3. Values: which defines property or values allocate for properties.

There are three types of CSS which are given below:

* Inline CSS

Inline CSS contains the CSS property in the body section attached with elements is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

* Internal CSS or Embedded CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section.
* External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attribute(for example class, id, heading). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

Advantages of Inline CSS:

* It can easily and quickly insert CSS rules to an HTML page.
* That’s why this method is useful for testing or previewing the changes, and performing quick-fixes to your website.

Disadvantages of Inline CSS:

* These styles cannot be reused anywhere else.
* These styles are tough to be edited because there are not stored in a single place.

Advantages of Internal CSS

* It only effects the page you are working on.
* You can still use classes and ID’s.
* It doesn’t require multiple pages which is great when you can only upload one HTML file.

Disadvantage of Internal CSS

* Adding the code to the HTML document can increase the page’s size and loading time.
* If the other forms/pages also need same styling -then external styling is important.

Advantages of External styling:

* Provides consistency between pages
* “Single” point of maintenance
* Easier to share with other sites

Disadvantages of External styling:

* Slower page load time

Class selector:

* Class selector selects the element with a specific class attribute. To select elements with a specific class. Write a period (.) character. Followed by the name of the class.

ID selector:

* The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page. So the id selector is used to select one unique element! To select an element with a specific id, write a #character, followed by the id of the element.

Tag selector:

* The tag selector is used to redefine existing HTML tags. Select this option if you want to change the formatting options for an HTML tag, such as the <h1> tag or the <ul>

Group selector:

* The group selector selects all the HTML elements with the same style definitions.
* To group selectors, separate each selector with a comma.

Global selector

* The global selector (\*) selects all the HTML elements on the page.

Attribute selector:

* This type of attribute selector is used to select all the elements that have the specified attribute and applies the CSS property to the attribute. For example the selector [class] will select all the elements with the style attribute.

[attribute=”value”] selector:

* This selector is used to select all the elements whose attribute has the value exactly same as the specified value.

[attribute~=”value”] selector:

* This selector is used to select all the elements whose attribute value is a list of space-separated values, one of which is exactly equal to the specified value.

[attribute|=”value”] selector:

* This selector is used to select all the elements whose attribute has a hyphen-separated list of values beginning with the specified value. The value has to be a whole word either alone or followed by a hyphen.

[attribute^=”value”] selector:

* This selector is used to select all the elements whose attribute value begins with the specified value. The value doesn’t need to be a whole word.

[attribute$=”value”] selector:

* This selector is used to select all the elements whose attribute value ends with the specified value. The value doesn’t need to be a whole word.

[attribute\*=”value”] selector:

* This selector selects all the elements whose attribute value contains the specified value present anywhere. The value doesn’t need to be a whole word.

Relationship selectors:

* Descendant selector (space):

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

* Child selector (>):

The child selector selects all elements that are the children of a specified element.

* Adjacent Sibling selector (+):

The adjacent sibling selector is used to select an element that is directly after another specific element.

Sibling element must have same parent element, and “adjacent” means “immediately following”.

* General Sibling selector (~):

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

Pseudo selectors:

* CSS pseudo-classes are keywords for the selectors that set attributes when the selector is in an extra special state.

Basically, pseudo classes set the style of an element when the element is in a particular special state.

List of most important selectors:

:link - It adds style to unvisited link

:visited – It adds style to a visited link.

:hover – it adds style to element when we mouse over it.

:active – It adds style to the active link.

:focus – It adds style to element when it has focus.

:first-child – This class adds style to the first child of the element.

:last-child – This class adds style to the second child of the element.

:lang – It defines the language of the specified element.

Box model properties:

* 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.

Properties:

* Content: This property is used to display the text, images, etc, that can be sized using the width and height property.
* Padding: This property is used to create space around the element, inside any defined border.
* Border: This property is used to cover the content and any padding, and also allows to set the style, color, and width of the border.
* Margin: This property is used to create space around the element i.e., around the border area.

Background Properties:

* The CSS background properties are used to define the background effects for elements. There are lots of properties to design the background.CSS background properties are as follows:
* CSS background-color Property: The background-color property in CSS is used to specify the background color of an element.
* CSS Background -repeat property: The background-repeat property in CSS is used to repeat the background image both horizontally and vertically.
* CSS Background-attachment property: The property background-attachment property in CSS is used to specify the kind of attachment of the background image with respect to its container.
* CSS Background-position property: In CSS body-position property is mainly used to set an image at a certain position.
* CSS Background-origin property: The background-origin property defines which helps in adjusting the background image of the webpage.
* CSS Background-clip property: The background-clip property in CSS is used to define how to extend background (color or image) within an element.

Float:

* The float CSS property places an element on the left or right side of the container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the page, though still remaining a part of the flow.
* Types of floating properties are left, right, none.

Flex:

* CSS Flexible box layout is a module of CSS that defines a CSS box model optimized for user interface design, and the layout of items in one dimension.

Properties:

* Display
* Flex-direction
* Flex-wrap
* Flex-flow
* Justify-content
* Align-items
* Align-content
* Gap, row-gap, column-gap

Grid:

* The CSS Grid layout module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
* The grid property is a shorthand property for: grid-template-rows, grid-template-columns.

Positions in CSS:

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.

Static:

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.

Relative:

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.

Fixed:

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.

Absolute:

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.

Sticky:

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).

Media Queries:

In CSS, a media query is used to apply a set of styles based on the browser's characteristics including width, height, or screen resolution. For large screen sizes like desktops, we can see a search menu in the upper left hand corner.

Media queries in CSS3 extended the CSS2 media types idea: Instead of looking for a type of device, they look at the capability of the device.

Media queries can be used to check many things, such as:

* width and height of the viewport
* width and height of the device
* orientation (is the tablet/phone in landscape or portrait mode?)
* resolution

Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones).



Transform:

The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, translate, skew, etc., elements.

* Rotate:

The CSS rotate() function lets you rotate an element on a 2D axis.The rotate() function accepts one argument: the angle at which you want to rotate your web element. You can rotate an element clockwise or counter-clockwise.

* Scale:

The scale() [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) [function](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Functions) defines a transformation that resizes an element on the 2D plane. Because the amount of scaling is defined by a vector, it can resize the horizontal and vertical dimensions at different scales.

* Skew:

The skew() [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) [function](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Functions) defines a transformation that skews an element on the 2D plane.

* Translate:

The translate CSS property allows you to transfer an element from one place to another along the X (horizontal) axis, the Y (vertical) axis, and the Z (depth) axis.

Transition:

CSS transitions allows you to change property values smoothly, over a given duration.

The following properties:

* transition-delay:

The transition-delay property specifies when the transition effect will start.

The transition-delay value is defined in seconds (s) or milliseconds (ms).

* Transition-duration:

The transition-duration property specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.

* Transition-property:

The transition-property property specifies the name of the CSS property the transition effect is for (the transition effect will start when the specified CSS property changes).

**Tip:** A transition effect could typically occur when a user hover over an element.

**Note:** Always specify the [transition-duration](https://www.w3schools.com/cssref/css3_pr_transition-duration.asp) property, otherwise the duration is 0, and the transition will have no effect.

* Transition-timing function:

The transition-timing-function property specifies the speed curve of the transition effect.

This property allows a transition effect to change speed over its duration.

Animation:

The animation property is a shorthand property for:

* Animation-name:

The animation-name property specifies a name for the [@keyframes](https://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp) animation.

* Animation-delay:

The animation-delay property specifies a delay for the start of an animation.

The animation-delay value is defined in seconds (s) or milliseconds (ms).

* Animation-direction:

The animation-direction property defines whether an animation should be played forwards, backwards or in alternate cycles.

* Animation-iteration-count:

The animation-iteration-count property specifies the number of times an animation should be played.

* Animation-duration:

The animation-duration property defines how long an animation should take to complete one cycle.

* Animation-fill-mode:

The animation-fill-mode property specifies a style for the element when the animation is not playing (before it starts, after it ends, or both).

CSS animations do not affect the element before the first keyframe is played or after the last keyframe is played. The animation-fill-mode property can override this behaviour.