

Advanced CSS Concepts

CSS variables:

CSS variables are the custom keywords used to store property values. These variables can be called at all the code areas where the desired value needs to be put. This is the most efficient code practice as value stored in the variable can be later on modified or deleted (as per the product requirement) very easily and quickly whereas locating all the places where the value has been written and then modifying/deleting it might be cumbersome and lead to unhandled code errors.

Syntax of declaring and using variables is as follows:

Variable Declaration: "--variable-name: value;" E.g. --grey-shade: grey;

Variable Consumption: "property: var(--variable-name);" E.g. color: var(--grey-shade);

CSS pseudo-selectors

These are the set of pre-defined classes in CSS used for special case styling on an element. Few of the highly used pseudo-selectors are as follows:

1. `:link` - This is a pseudo-selector used on anchor tags to define styles which enhances the pre-defined link behaviour of an anchor tag.
2. `:checked` - This is a pseudo-selector used on input type radio or checkbox, used to style them differently when they are in checked state.
3. `:disabled` - This is used on input types or elements to style them differently when they are disabled state.
4. `:invalid` - This is used on several input types to style them differently when they we input mismatched type of values in them. This pseudo-selector has met form validation needs efficiently.
5. `:hover` - This selector is used to add styles to an element to present it differently when user hovers on the element.
6. `:first-child` - This selector is used to add styles differently, to the first child of an element which acts as parent to it.
7. `:last-child` - This selector is used to add styles differently, to the last child of an element which acts as parent to it.
8. `:nth-child` - This selector is used to add styles differently, to the nth child of an element which acts as parent to it. This nth child could be any number like `nth-child(4)` the fourth child element, `nth-child(odd)` all the odd child elements, `nth-child(5n)` all the child elements appearing at a count of 5 will be styled differently and so on.

CSS pseudo-elements

These are the set of pre-defined styles in CSS used for special case styling on an element. Few of the highly used pseudo-elements are as follows:

1. `::first-line` - This selects the first line of typography tags and is used to style them differently.
2. `::first-letter` - This is used to add different styles to first letter in a typographic element or input elements.
3. `::after` - This is used to concatenate a string value towards the end of an element. The mandatory property-value pair needed to display output of this is content: "string/empty".
4. `::before` - This is used to concatenate a string value towards the beginning of an element. The mandatory property-value pair needed to display output of this is content: "string/empty".

Font-Awesome Icons Library

Font-awesome icons library is used widely to include icons on a web page or website. It is a repository of icons and provide commonly used icons for free to the web applications. There are other similar icon repositories available online. the Font-awesome repository may be used in the following ways:

1. CDN link: developers may include the cdn (content delivery network) link for font-awesome repository in the head section of root html file in this way: `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">`
2. NPM Package: while developing a website using a JS framework, a node package may be installed in the application and used directly. This method has an advantage over the cdn link that if the server where the icons are placed, runs down, it does not affect our website as we have the package locally installed in our app. The package may be installed running this command "npm i font-awesome".
3. Grammar to use the icons:

1. CSS content-code method: by using the pseudo-element after/before with the element to which icon needs to be applied, the content attribute value can be substituted with string content code of the icon and including the font-family attribute to be FontAwesome.

E.g.

```
p::after{
  content: '\f01e';
  font-family: FontAwesome;
}
```

2. HTML entity code method: much alike css content code method, the html entity code is written inside the element where icon is needed and element css must contain font-family as FontAwesome.

E.g.

```
<span> '&#www' home </span >
css: span { font-family: FontAwesome;}
```

3. HTML tag and class attribute method: An i tag with class="fa icon-class" is placed in the html code to display the icon.

E.g.

```
<i class="fa fa-adjust">
```