TASK1: Explain the purpose of using the var() function in css. Also you have created two buttons with id named primaryBtn and secondaryBtn which should be given background colors using the var() function. The color code for primaryColor is #00b7ff and secondaryColor is #6c757d.

SOLUTION:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

:root {

--primaryColor: #00b7ff;

--secondaryColor: #6c757d;

}

.firstbutton{

background-color: var(--primaryColor);

}

.seconbutton{

background-color: var(--secondaryColor);

}

</style>

</head>

<body>

<div>

<button class="firstbutton">primaryBtn</button>

<button class="secondbutton">secondaryBtn</button>

</div>

</body>

</html>

The var() function is used to insert the value of a CSS variable.

CSS variables have access to the DOM, which means that you can create variables with local or global scope, change the variables with JavaScript, and change the variables based on media queries.

TASK2: Create a 3D cube using the transform property of CSS.

SOLUTION:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="box">

<div class="cube">

<div class="face front">front</div>

<div class="face back">back</div>

<div class="face top">top</div>

<div class="face bottom">bottom</div>

<div class="face left">left</div>

<div class="face right">right</div>

</div>

</div>

</body>

</html>

CSS

.box{

/\* width: 100vw; \*/

min-height: 100vh;

background-color: #cccc;

display: flex;

justify-content: center;

align-items: center;

/\* position: absolute; \*/

}

.cube{

background-color: chocolate;

position: relative;

transform: rotateX(20deg) rotateY(30deg);

transform-style: preserve-3d;

}

.face{

width: 200px;

height: 200px;

position: absolute;

display: flex;

justify-content: center;

align-items: center;

background-color: cadetblue;

opacity: 0.7;

border: 2px solid black;

font-size: 20px;

font-family: sans-serif;

}

.front {

transform: translateZ(100px);

}

.back {

transform: rotateY(180deg) translateZ(100px);

}

.right {

transform: rotateY(90deg) translateZ(100px);

}

.left {

transform: rotateY(-90deg) translateZ(100px);

}

.top {

transform: rotateX(90deg) translateZ(100px);

}

.bottom {

transform: rotateX(-90deg) translateZ(100px);

}

TASK3:Create a simple circular loader which will rotate continuously to look like a loading screen on a website.

SOLUTION:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

body{

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

margin: 0;

background-color: #f0f0f0;

}

.loader{

width: 50px;

height: 50px;

border-radius: 50%;

border: 7px solid rgba(0, 0, 0, 0.5);

border-top-color: #044c99;

animation: loading 0.75s ease infinite;

}

@keyframes loading {

from { transform: rotate(0turn);}

to { transform: rotate(1turn);}

}

</style>

</head>

<body>

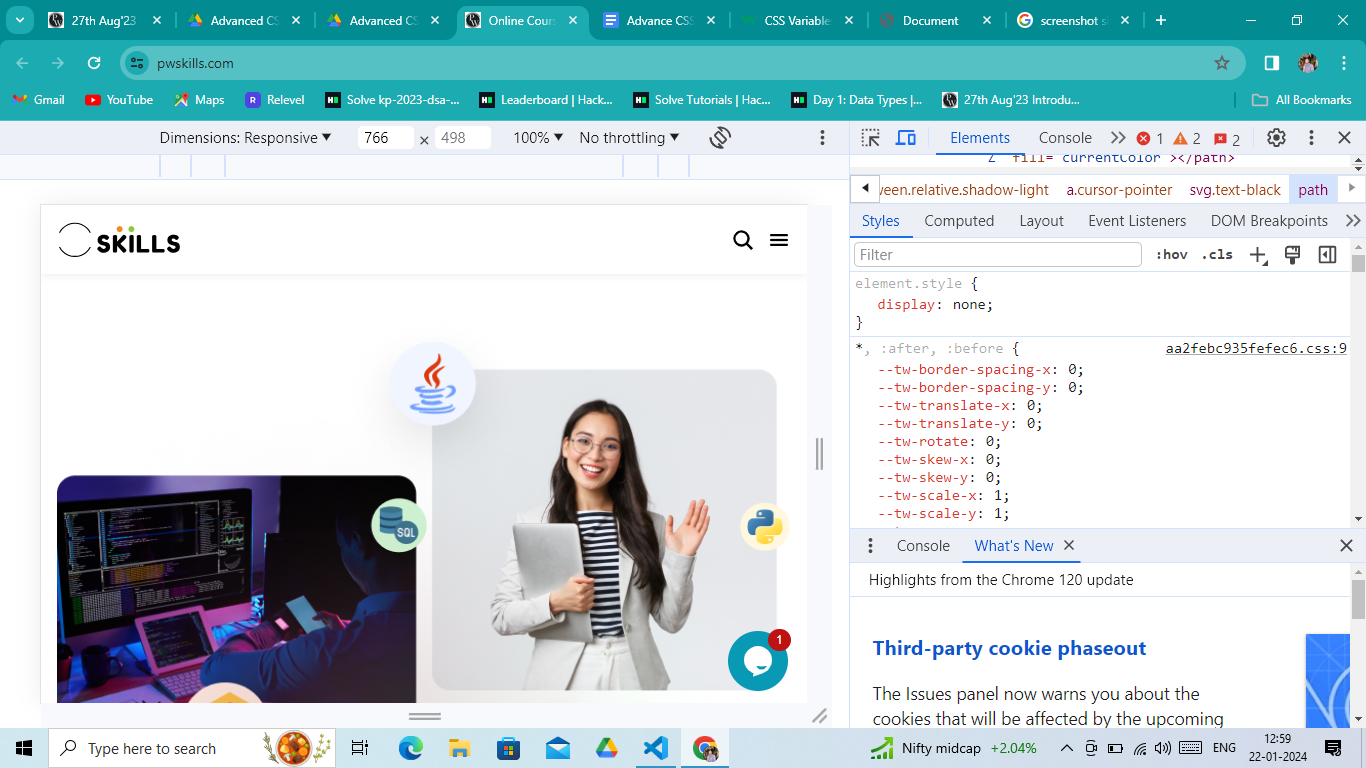
<div class="loader"></div>

</body>

</html>

TASK4: You have to visit the PW Skills website (https://pwskills.com/) and have to hide the logo by using the developer tool. This should be done using the css and developer tool only, use of javascript is prohibited.

SOLUTION:



TASK5: You have to visit the PW Skills website (https://pwskills.com/) and have to change the content of the ‘Login / Register’ button to ‘Connect with us’ using the developer tool. This should be done using the elements of developer tools only, use of javascript is prohibited.

SOLUTION:

