

ADVANCED CSS - ANSWERS

1. 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.

Var() function:

The var() function in CSS is a fundamental tool for utilizing CSS custom properties (often called CSS variables). Its primary purpose is to retrieve the value of a custom property that has been defined elsewhere in the stylesheet. This enables more dynamic, maintainable, and readable CSS.

Purpose:

- Reusability and DRY principle (Don't Repeat Yourself).
- Centralized management and easier maintenance.
- Theming and dynamic styling.
- Fallback values for robustness.

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Button</title>
    <link href="style.css" rel="stylesheet">
<body>
    <div class="container">
        <button id="primary">Primary Button</button>
        <button id="secondary">Secondary Button</button>
    </div>
</body>
</html>
```

CSS:

```
:root{  
    --primary-color:#00b7ff;  
    --secondary-color:#6c757d;  
    --text-color:#f3f3f3;  
}  
  
#primary{  
    border:0px solid;  
    padding: 5px;  
    background-color: var(--primary-color);  
    color:var(--text-color);  
    width:20%;  
}  
  
#secondary{  
    border:0px solid;  
    padding: 5px;  
    background-color: var(--secondary-color);  
    color:var(--text-color);  
    width:24%;  
}
```

OUTPUT:



-
2. Create a 3D cube using the transform property of CSS.

HTML:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
    <title>Button</title>  
    <link href="style.css" rel="stylesheet">  
<body>
```

```
<div class="container">
  <div class="face front">FRONT</div>
  <div class="face back">BACK</div>
  <div class="face right">RIGHT</div>
  <div class="face left">LEFT</div>
  <div class="face top">TOP</div>
  <div class="face bottom">BOTTOM</div>
</div>
</body>
</html>
```

CSS:

```
:root{
  --front-color:blue;
  --back-color:red;
  --right-color:green;
  --left-color:orange;
  --top-color:indigo;
  --bottom-color:purple;

  --cube-size:150px;
}

body{
  display: flex;
  justify-content: center;
  align-items: center;
  min-height:100vh;
  background-color: #f2f5f0;
  font-family: sans-serif;
  overflow: hidden;
}

.container{
  height:var(--cube-size);
  width: var(--cube-size);
  position: relative;
  transform-style:preserve-3d;
  perspective: 1000px;
  animation: rotateCube 10s infinite linear;
}

@keyframes rotateCube{
  from {
    transform: rotateX(0deg) rotateY(0deg);
  }
}
```

```
        }
      to {
        transform: rotateX(360deg) rotateY(360deg);
      }
    }
  .face{
    position:absolute;
    width: var(--cube-size);
    height: var(--cube-size);
    display: flex;
    justify-content: center;
    align-items: center;
    border:1px solid rgba(0, 0, 0, 0.1);
    font-size:1.5rem;
    font-weight: bold;
    color:white;
    opacity: 0.9;
    border-radius: 8px;
    box-shadow:0px 4px 6px rgba(0, 0, 0, 0.1);
  }
  .front{
    background-color: var(--front-color);
    transform: translateZ(calc(var(--cube-size)/2)) ;
  }
  .back{
    background-color: var(--back-color);
    transform: rotateY(180deg)
    translateZ(calc(var(--cube-size)/2)) ;
  }
  .right{
    background-color: var(--right-color);
    transform: rotateY(90deg)
    translateZ(calc(var(--cube-size)/2)) ;
  }
  .left{
    background-color: var(--left-color);
    transform: rotateY(-90deg)
    translateZ(calc(var(--cube-size)/2)) ;
  }
  .top{
    background-color: var(--top-color);
    transform: rotateX(90deg)
    translateZ(calc(var(--cube-size)/2)) ;
  }
```

```
}

.bottom{
    background-color: var(--bottom-color);
    transform: rotateX(-90deg)
translateZ(calc(var(--cube-size)/2)) ;
}
```

OUTPUT:



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

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Button</title>
    <link href="style.css" rel="stylesheet">
<body>
    <div class="container">
    </div>
</body>
</html>
```

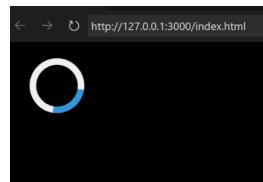
CSS:

```
body{
    background-color: black;
}

.container{
    width:50px;
    height: 50px;
    margin: 20px;
    border:8px solid #f3f3f3;
    border-radius: 50%;
    border-top: 8px solid #3498db;
    animation: spin 2s linear infinite;
}

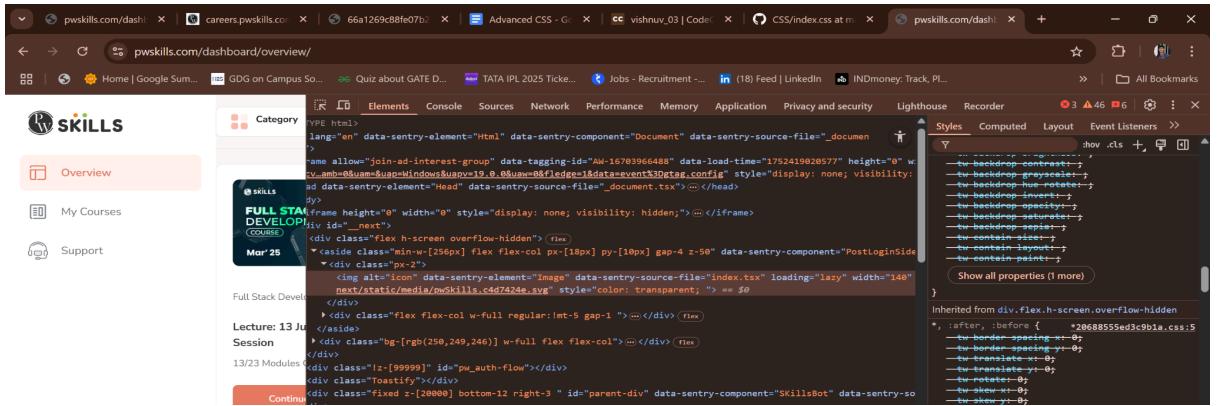
@keyframes spin{
    0% {
        transform:rotate(0deg);
    }
    100% {
        transform:rotate(360deg);
    }
}
```

OUTPUT:

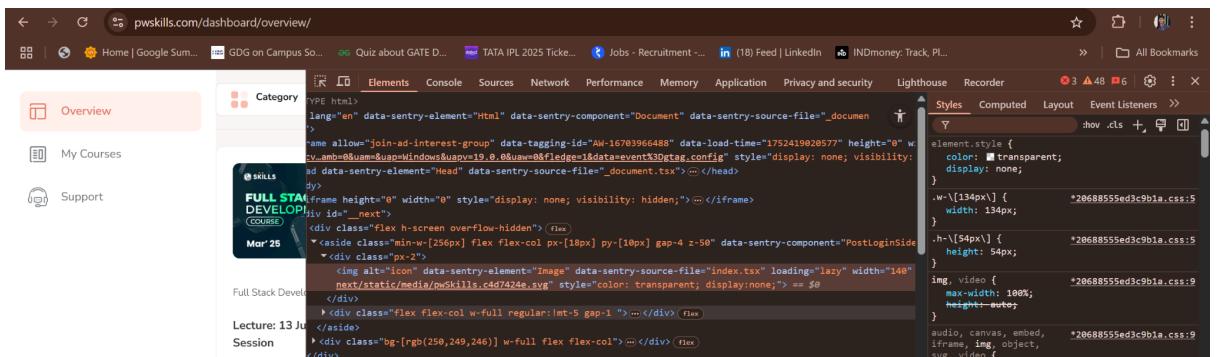


-
4. You have to visit the PW Skills website (<https://pwskills.com/>) and have to hide the logo by using the developer tool. This should be done using the css and developer tool only, use of javascript is prohibited.

Default Output:

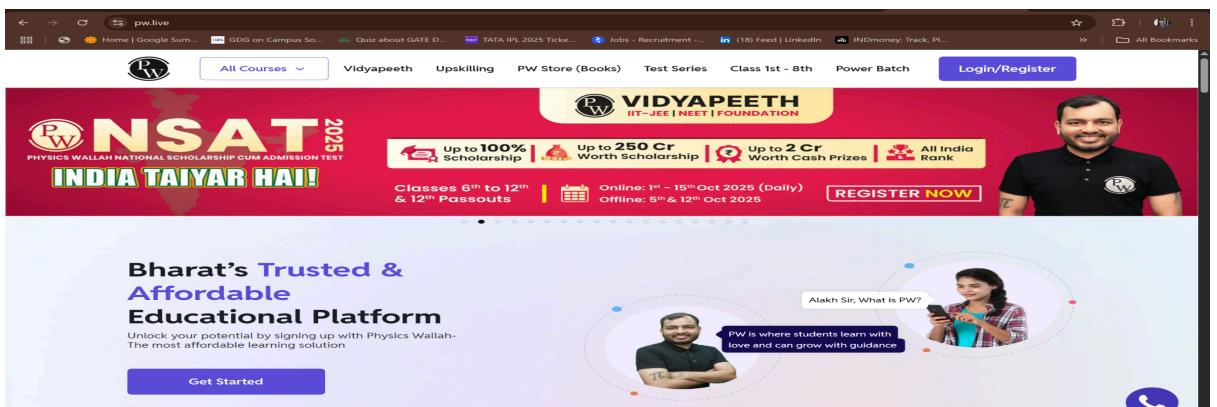


Final Output:



5. 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.

Default Output:



Final Output:

