Solution

PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

UE17MC405

DECEMBER 2017: IN SEMESTER ASSESSMENT (ISA) MCA. I SEMESTER

TEST – 2

UE17MC405-Introdution to Web Technology

Time: 1½ Hrs | Answer All Questions | Max Marks: 40

1. a) **Draw a half circle using HTML Graphics element. Fill the half circle with one color and text message with other color.** **5**

- **Proper use of SVG /Canvas tag: 1 Mark,**
- **Use of clip-path tag /arc: 2 Marks**
- **Circle tag / fill style:1 Mark**
- **Drawing text: 1 Mark**

SVG

```
<!DOCTYPE html>
<html lang="en">
        <head>
        <meta charset="UTF-8"/>
        <title>Circle</title>
        </head>
<body>
<svg height="200px" width="200px" >
    <clipPath id="cut-off-bottom">
       <rect x="0" y="0" width="200"
height="100"/>
    </clipPath>

    <circle cx="100" cy="100" r="100"
clip-path="url(#cut-off-bottom)"
fill="yellow" />
        <text x="80" y="50" font-
family="sans-serif" font-size="20px"
fill="red">PESU</text>
</svg>
</body>
</html>
```

Canvas

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200"
height="200" style="border:1px solid
#d3d3d3;">
Your browser does not support the HTML5
canvas tag.</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.arc(95,50,40,0,-1 * Math.PI, 0.75 *
Math.PI);
ctx.fillStyle = "yellow";
ctx.fill();
ctx.fillStyle = "red";
ctx.font = "15px Arial";
ctx.fillText("PESU",70,40);
//context.arc(x,y,r,sAngle,eAngle,clockwise);
</script>
</body>
</html>
```

Solution

	b)	**How to provide multiple background images for the text element?**	**5**

Each correct property: 1 Mark, Explanation: 1 Mark

CSS3 background properties have greater control on the background element. Different background images are separated by commas, and the images are stacked on top of each other, where the first image isclosest to the viewer.

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
/*two background images, the first image is a flower (aligned to the bottom and right)
and the second image is a paper background (aligned to the top-left corner)*/
background-image: url(flower.gif), url(paper.gif);
background-position: right bottom, left top;
background-repeat: no-repeat, repeat;
background-size: 100px 80px;
//other possible values for background-size are contain and cover
padding: 15px;
}
//background shorthand property
/*#example1 { background: url(flower.gif) right bottom no-repeat,
url(paper.gif) left top repeat; } */
</style>
</head>
<body>
<div id="example1">
<h1>Lorem Ipsum Dolor</h1>
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
<p>Ut wisi enim ad minim veniam, quis nostrud exerci tationullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
</div>
</body>
</html>
```

2.	**a)**	**Explain with example types of CSS3 colors pattern available to display element background.**	**5**

Each Color explanation with example: 1 Mark, Syntax: 1 Mark

CSS3 introduces:

- **RGBA colors :** values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color. alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).
- **HSL colors:** HSL stands for Hue, Saturation and Lightness. HSL stands for Hue, Saturation and Lightness.
- **HSLA colors:** HSLA color values are an extension of HSL color values with an alpha channel - specifies the opacity for a color.
- **Opacity:** The CSS3 opacity property sets the opacity for a specified RGB value.
- The opacity property value must be a number between 0.0 (fully transparent) and 1.0

Solution

(fully opaque)

```
rgba(255, 0, 0, 0.2);
rgba(255, 0, 0, 0.4);
rgba(255, 0, 0, 0.6);
rgba(255, 0, 0, 0.8);
```

```
hsl(0, 100%, 30%);
hsl(0, 100%, 50%);
hsl(0, 100%, 70%);
hsl(0, 100%, 90%);
```

```
hsla(0, 100%, 30%, 0.3);
hsla(0, 100%, 50%, 0.3);
hsla(0, 100%, 70%, 0.3);
hsla(0, 100%, 90%, 0.3);
```

```
rgb(255, 0, 0);opacity:0.2;
rgb(255, 0, 0);opacity:0.4;
rgb(255, 0, 0);opacity:0.6;
rgb(255, 0, 0);opacity:0.8;
```

	b)	**Write a CSS3 program to animate a red square to become orange circle and back to yellow square again.**	**5**

```
<!DOCTYPE html>
<html> <head> <style>
#element { height: 250px; width: 250px;
margin: 0 auto; background-color: red;
animation-name: stretch; animation-duration: 1.5s;
animation-timing-function: ease-out;  animation-delay: 0;
animation-direction: alternate; animation-iteration-count: infinite;
animation-fill-mode: none; animation-play-state: running; }
@keyframes stretch {
    0% {transform: scale(.3);background-color: red; border-radius: 100%;}
    50% {background-color: orange;}
    100% {transform: scale(1.5);background-color: yellow;} }
body, html {height: 100%;}
body {display: flex; align-items: center; justify-content: center;}
</style></head><body><p id="element"></p></body></html>
```

1
1
1
1
1

3.	**a)**	**Explain the functions available in 2D transformation with example.**	**5**

Function with explanation: 1 Mark, Syntax/ Example: 1 Mark

CSS3 transforms allow you to translate, rotate, scale, ,skew, matrix elements. A 2D transformation is an effect that lets an element change shape, size and position.

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

Solution

transform: translate(50px,100px);

The **rotate()** method rotates an element clockwise or counter-clockwise according to a given degree.

transform: rotate(20deg);

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

transform: scale(0.5,0.5);

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

transform: skewX(20deg);

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

transform: skewY(20deg);

The **skew() Method**

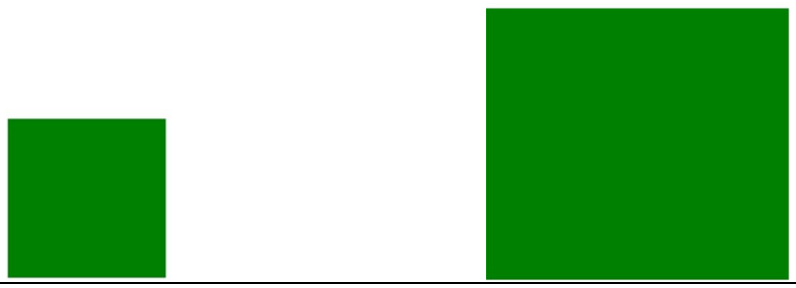
The skew() method skews an element along the X and Y-axis by the given angles. skews the element 20 degrees along the X-axis, and 10 degrees along the Y-axis.

transform: skew(20deg,10deg);

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.

transform: matrix(1, 0, 0.5, 1, 150, 0);

b)	**Write a program to display the use of CSS transition and using pseudo class change html element for both x and y coordinate to certain size as shown below.** 	**5**

- **Transition: 2 Marks, Hover: 2 Marks. Div: 1 Mark**

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
   width: 100px;
   height: 100px;
   background: green;
   -webkit-transition: width 2s, height 4s; /* For Safari 3.1 to 6.0 */
   transition: width 2s, height 4s;
}
div:hover {
   width: 300px;
   height: 300px;
}
</style>
</head>
<body>
```

Solution

4. a) Describe various string manipulation functions in Java script. 5

Each function: 1Mark

String methods can be used through String primitive values, as if the values were objects. The String object includes one property ***length*** and a large collection of methods.

The number of characters in a string is stored in the ***length*** property as follows:

```
var str=”George”;
var len=str.length;
```

Here len contains the value 6.

The other string manipulation methods are listed below:

Method	Parameters	Result
CharAt	A number	Returns the character in the String object that is at the specified position.
indexOf	One character string	Returns the position in the String object of the parameter.
Substring	Two numbers	Returns the substring of the String object from the first parameter position to the second.
toLowerCase	None	Converts any uppercase letters in the string to lowercase.
toUpperCase	None	Converts any lowercase letters in the string to uppercase.

For the string methods, character positions start at ‘zero’.

Example:

```
var str=”George”;
```

Then,

str.charAt(2) is ‘o’

str.indexOf(‘r’) is 3

str.substring(2,4) is ‘org’

str.toLowerCase() is ‘george’

b) Develop a JavaScript function for the following problems **5**

Input: A sequence of number

Output: The sequence with its digits in the reverse order

Function: 2 Mark, Logic:3 Mark

```
<html>
<head></head>
<body>
Enter a number :
<input type=text name=number>
<input type="button" value="Click me!" onclick="disp(number)" >
        <script type="text/javascript">
                function disp(num)
                {
                var rev=0, rem=0, n= Number(num.value);
                        while(n!=0)
```

Solution

```
                        {
                        rem = n%10;
                        n = Math.floor(n/10);
                        rev = rev*10 + rem;
                        }
                alert("The " + num.value + " in reverse is " + rev);
                }
        </script>
</body>
</html>
```

OR (without function)

```
<html >
<head><title> Exercise3b </title></head>
<body>
                <script type="text/javascript">
                        var r=0,d,n=65656;
                        document.write("The given digit is " +n+ "<br />");
                        do{
                                d=n%10;
                                n=parseInt(n/10);
                                r=(r*10)+d;
                          }
                        while (n>0);
                        document.write("The reverse of the given digit is ", +r);
                </script>
</body>
</html>
```