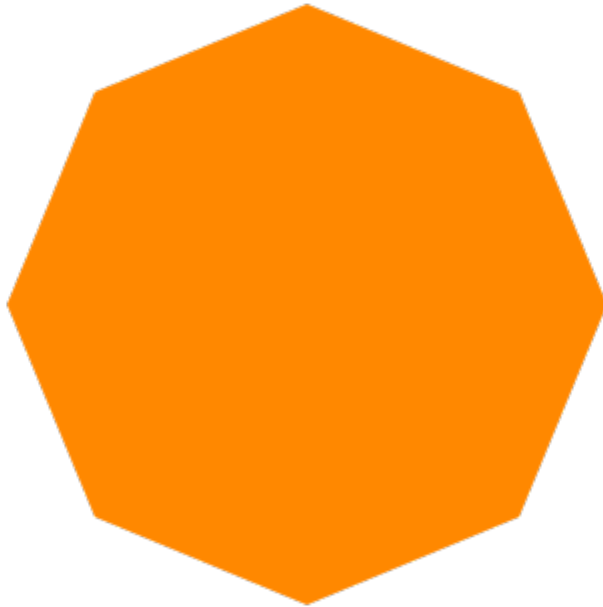


# ADDING COLORS IN THE HTML5

## *Polygons and the artistsColour wheel*

### *HTML5 canvas element - polygons and the artists colour wheel*

When drawing a circle, the computer program actually draws a polygon, with so many sides it looks like a circle. By actually specifying the number of sides the program can be made to draw regular polygons. The example below shows an octagon. Only some minor changes have been made to the original circle program.



```
function polygon1()
{
    var sides = document.getElementById('poly1').value;
    drawCircle(sides);
}

function drawCircle(s)
{
    // Create a circle i.e. polygon with very many sides
    var canvas = document.getElementById("fig1");
    var context = canvas.getContext("2d");
    if (context)
    {
        var A = 150;          //radius
        //prevent number of sides being specified that are less than 3
```

```

var sides = parseInt(s); (sides < 3 ) ? sides = 3 : sides;
var originx = 240; var originy = 180; //set origin
context.clearRect(0,0,480,360); //clear previous drawing
context.beginPath(); //start new path
context.moveTo(originx + A,originy); //move to starting point
for(var angle=0;angle<2*Math.PI;angle+=2*Math.PI/sides)
{
    var x = A * Math.cos(angle) + originx
    var y = A * Math.sin(angle) + originy;
    context.lineTo(x,y); //draw line to new point
}
context.fillStyle = "#f80"; //set pen colour
context.fill();
}
else
{
    alert("Browser does not support HTML5 canvas");
}
}

```

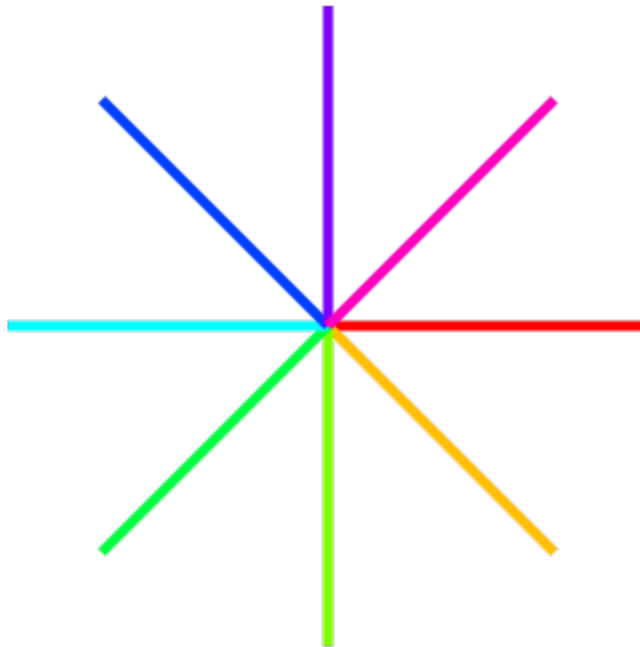
## Colour wheel - How RGB colours change with angle

Web colours are specified using RGB (Red, Green,Blue) values, however this is not the only way colours can be specified. An alternative, used by artists is HSV (Hue, Saturation Value). Here we will only concern ourselves with the hue. From the diagram below you can see that a circle has been divided into segments and for each segment, only one of the three (RGB) colour components changes.

**Note:** within javascript, colours may be specified as "#rgb", "#rrggbb", rgb(rr,gg,bb) or rgba(rr,gg,bb,opacity). The rgba() function is similar to the rgb() function but it has one extra parameter which sets the transparency value. The valid range is from 0.0 (fully transparent) to 1.0 (fully opaque).

Hence any colour (hue) can be represented by its position in a **colour wheel**, as the next diagram shows. Here just 8 colours have been shown.

Colours opposite one another on the colour wheel are complementary, i.e. the compliments of Red, Green and Blue are Cyan,Magenta and yellow



By increasing the number of colours we get the full colour wheel.



```
function polygon2()
{
    var sides = document.getElementById('poly2').value;
    drawSegment(sides);
}

function drawSegment(s)
{
    // Create a circle i.e. polygon with very many sides
    var canvas = document.getElementById("fig4");
    var context = canvas.getContext("2d");
    context.save();
    context.clearRect(0,0,480,360);           //clear previous drawing

    context.translate(240,180);              //move origin to centre of canvas
    var A = 150;                             //radius
    var sides = parseInt(s); (sides < 3 ) ? sides = 3 : sides;    //number of
sides
    var x0 = A; var y0 = 0;                  //set initial point
    var rd = 0;

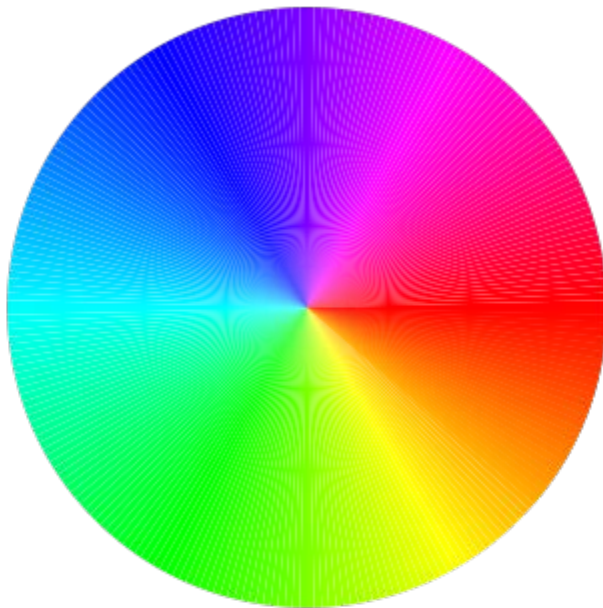
    while (rd<=Math.ceil(2*Math.PI))
```

```

{
  var angle = rd*360/(2*Math.PI);
  var colour = hue(angle).colour;
  var x = A * Math.cos(rd);           //calculate x,y values
  var y = A * Math.sin(rd);

  context.beginPath();
  context.fillStyle = colour;
  context.moveTo(0,0);                //move to starting point
  context.lineTo(x,y);                //draw line to new point
  context.lineTo(x0,y0);              //draw line back to previous point
  context.lineTo(0,0);
  //context.stroke();
  context.fill();
  x0 = x; y0 = y;
  rd += 2*Math.PI/sides;
}
context.restore();                    //put origin back
}

```



**Note:** When calling the fill method, any open shapes will be closed automatically and it isn't necessary to use the closePath method.

Source : <http://www.soslug.org/node/1735>