

## Rotating\_Word\_on\_an\_Arc

//Rotating Word Along an Arc

//commands include: translate(), pushMatrix() and popMatrix()

//Mr. H.

//declares a "String" for the phrase displayed

**String** phrase = "Hello World";

//declares a font

**PFont** displayedFont;

//arc radius of phrase

**float** phraseRadius = 150;

//creates variables for the length of the phrase

//needed to place each letter along arc

**float** phraseArcLength = 0;

**void setup()** {

**size**(600, 600);

  displayedFont = **createFont**("Times", 40, **true**);

**textFont**(displayedFont);

**textAlign**(**CENTER**,**CENTER**);

}

**void draw()** {

**background**(255);

  //Draws a grey circle centred in window

**translate**(**width** / 2, **height** / 2);

**noFill**();

**stroke**(230);

**ellipse**(0, 0, phraseRadius\*2, phraseRadius\*2);

  //the "millis()" command needed to move text

  phraseArcLength = **millis**()/10;

  //places each letter in order

**for** (**int** i = 0; i < phrase.**length** (); i = i + 1)

  {

    //adds each letter to get the total phrase

**char** phraseChar = phrase.**charAt**(i);

**float** phraseWidth = **textWidth**(phraseChar);

    //centre each letter and move it over half its width

    phraseArcLength = phraseArcLength + phraseWidth/2;

    //calculates angle in radians by dividing arclength by radius

**float** angle = phraseArcLength / phraseRadius;

```
pushMatrix();
//using polar coordinates of radius and angle to place letters
//along arcs but need to be converted back to x and y coordinates
translate(phraseRadius*cos(angle), phraseRadius*sin(angle));

//rotates the letters so bottom is lined up with curve
rotate(angle+PI/2);

//displays each letter of the phrase
fill(50, 50, 200);
text(phraseChar, 0, 0);
popMatrix();

//moves halfway again so next letter lines up correctly
phraseArcLength = phraseArcLength + phraseWidth/2;
}
}
```