Shape5.js

Shape5.js is a wrapper for P5.js. You can try out the Shape5.js library here. The idea behind is project is to reduce the learning curve for making static art with P5.

Shapes

The following shapes are available in Shape5.js. Please pay attention to the spelling and capitalization of each shape. See the Attributes section for a more detailed look at each shape.

- Circle();
- Square();
- Triangle();
- Polygon();
- Oval();
- Rectangle();
- Semicircle();
- Rhombus();
- RightTriangle();
- IsoscelesTriangle();
- Stripe();

Syntax

The pattern for creating a shape is to declare a variable, instantiate a shape with the variable, modify the shape, and finally draw the shape.

Vocabulary Selection

To avoid overloading P5 keywords, words like `spin`, `show`, and `stripe` are used instead of `rotate`, `draw`, and `line`. This was a conscious decision so that users will not think that the workings of Shapes.js are in alignment with P5.js. For example, the `rotate` command in P5.js rotates the contents of the `draw` function around the origin point. Shape5.js, however does not want to have to introduce users to `push`, `translate`, and `pop` in order to rotate a shape. In addition, users do not need to understand radians, they can express all rotations in terms of degrees.

Shape Variable

The shape variable can be global or local to the `draw` function. Global variables may help to reinforce more traditional coding practices. Using the global variable method is not necessary. It may prove to be more confusing for first-time coders since the shape must be instantiated in the `setup` function. Shape variables can also be declared and instantiated in a single line of code directly in the `draw` function.

Instantiating the Shape

If you are using the global variable method, then the shapes must be instantiated in the `setup` function. P5 will not allow you to instantiate the shape before the `setup` function. If you are doing all of the coding in the `draw` function, then creating the variable and instantiating the shape can happen in one line of code.

Modifying the Shape

How a shape can be modified is dependent upon the attributes associated with each shape. All shapes share a limited set of attributes. See the Attributes section below for a complete list of attributes for each shape. By default, Shape5.js uses CSS colors. Users can also use the traditional `color` command as outlined by the P5 documentation. Modifications for shapes should come before it is drawn to the canvas. Below is the pattern for changing an attribute. See the example section for more detail.

```
variable_name.attribute = value;
```

Drawing the Shape

Each shape has a `.show()` method. Use this to draw the shape to the canvas.

Example Shape - Global Variable

```
let box;

function setup() {
  createCanvas(400, 400);
  box = new Square();
}

function draw() {
  background('black');

  box.x = 150;
  box.y = 300;
  box.size = 75;
  box.color = 'papayawhip';
  box.spin = 45;
  box.show();
}
```

Example Shape - Local Variable

```
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background('black');

  let box = new Square();
  box.x = 150;
  box.y = 300;
  box.size = 75;
  box.color = 'papayawhip';
  box.spin = 45;
  box.show();
}
```

Attributes

- **Circle();**
    - *this.x* - x-position of the shape

- *this.y* - y-position of the shape
- *this.r* - radius of the circle
- *this.color* - color of the shape, default is `'darkgray'`
- *this.spin* - rotation of the shape, default is 0

- **Square();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.size* - length of each side
  - This shape's position is located at the center of the square

- **Triangle();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.size* - size of each side of the triangle (equilateral)
  - This shape's position is located in the center of the triangle

- **Polygon();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.size* - size of the polygon, not related to the length of each side
  - *this.sides* - number of sides, default is 6
  - This shape's position is located at the center of the polygon

- **Oval();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.width* - width of the oval
  - *this.height* - height of the oval

- This shape's position is located at the center of the oval

- **Rectangle();**
  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.width* - width of the rectangle
  - *this.height* - height of the rectangle
  - This shape's position is located at the center of the rectangle

- **Semicircle();**
  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.size* - radius of the semicircle
  - *this.degrees* - angle of the semicircle; based on the `arc` command in P5; positive numbers draw the semicircle to the right, negative numbers to the left; semicircles begin from the 12 o'clock position
  - This shape's position is located at the point of the semicircle

- **Rhombus();**
  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.size* - length of each side of the rhombus (equilateral polygon)
  - *this.angle* - size of two of the interior angles; the other two are calculated from this attribute
  - The rhombus defaults to a diamond shape
  - This shape's position is located at the center of the rhombus

- **RightTriangle();**
  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.width* - width of the triangle

- *this.height* - height of the triangle
- This shape's position is located at 90° angle

- **IsoscelesTriangle();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.length* - length of the two equal sides of the triangle
  - *this.width* - length of the third side of the triangle
  - This shape's position is located at the midpoint of the unequal side of the triangle

- **Stripe();**

  - *this.x* - x-position of the shape
  - *this.y* - y-position of the shape
  - *this.color* - color of the shape, default is `'darkgray'`
  - *this.spin* - rotation of the shape, default is 0
  - *this.length* - length of the line
  - *this.width* - stroke weight of the line