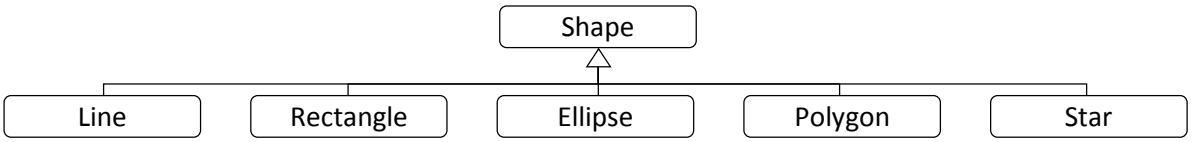
Description

Create a JavaScript file named `shapes.js` that implements several Shape object constructor functions. The constructor functions should set up the following object hierarchy.



Shape initializes object properties that hold graphic styles for the specific shape object and sets them to appropriate default values. Shape also implements a `draw(…)` method that uses these properties to implement the common tasks performed when drawing a shape. The `draw(…)` method should be inherited by all Shape subclasses.

Shape subclasses perform the tasks unique to a particular shape to be drawn, namely to create the path corresponding to the shape given its particular parameters. All Shape subclasses should implement a method called `trace(…)` that creates the path outlining the Shape subclass. The `trace(…)` method will be invoked as the first step of the Shape `draw(…)` method before setting styles and filling and/or stroking the shape.

The following table details all object properties to be implemented.

Object	Constructor/Property	Description
`Shape`	`new Shape()`	The Shape constructor function has no parameters. It initializes the `fillStyle`, `strokeStyle` and `lineWidth` properties of the object to any suitable default values.
	`fillStyle`	Saves the `fillStyle` value to be used when drawing this shape. If set to `undefined`, the shape should not be filled.
	`strokeStyle`	Saves the `strokeStyle` to be used when drawing this shape. If set to `undefined`, the shape should not be stroked.
	`lineWidth`	Saves the `lineWidth` value to be used when drawing this shape. If set to `undefined`, the shape should not be stroked.
	`draw(ctx)`	Draws the shape on the 2D drawing context. Starts be invoking the subclass's `trace(ctx)` method to create a path. Then sets styles using saved properties, and fills and/or strokes the shape.
`Line`	`new Line(x1, y1, x2, y2)`	The Line constructor function initializes properties that define the first point (`x1`, `y1`) and last point (`x2`, `x2`) of the line.
	`trace(ctx)`	Creates the line path on the 2D drawing context (`ctx`)
`Rectangle`	`new Rectangle(x, y, w, h)`	The Rectangle constructor function initializes the upper left corner (`x`, `y`), width (`w`) and height (`h`) of the rectangle.
	`trace(ctx)`	Creates the rectangle path on the 2D drawing context (`ctx`)

`Ellipse`	`new Ellipse(cx, cy, rx, ry)`	The Ellipse constructor function initializes the center point (`cx`, `cy`), x-radius (`rx`), and y-radius (`ry`) of the ellipse.
	`trace(ctx)`	Creates the ellipse path on the 2D drawing context (`ctx`) using an arc.
`Polygon`	`function(cx, cy, radius, npoints)`	The Polygon constructor function initializes the center point (`cx`, `cy`), `radius` and number of points (`npoints`) that make up the polygon.
	`trace(ctx)`	Creates the polygon-shaped path on the 2D drawing context (`ctx`) made up of a closed path of straight lines.
`Star`	`function(cx, cy, radius, npoints)`	The Star constructor function initializes the center point (`cx`, `cy`), `radius` and number of points (`npoints`) that make up the polygon.
	`trace(ctx)`	Creates a star-shaped path on the 2D drawing context (`ctx`) made up of a closed path of straight lines.

Requirements

1. Implement all constructor functions as detailed in the table above.
2. Implement all object properties and additional functions as detailed in the table above.
3. Implement the inheritance hierarchy as depicted in the diagram.
4. Note that when `fillStyle` is set to `undefined`, the shape should not be filled. When either `strokeStyle` or `lineWidth` is set to `undefined`, the shape should not be stroked.
5. Write a test program that programmatically draws a scene or abstract drawing. Your drawing must be made up of at least one instance each of Ellipse, Rectangle, and Line, and at least one instance of Polygon or Star.
6. You MUST enter header comments into your JavaScript file including (1) File name, (2) Your name, (3) Description and or purpose of the assignment.
7. You MUST comment your code, explaining what you did in each section.
8. Submit your HTML and/or JavaScript files using Canvas under the appropriate assignment.