# Homework: Prototype Chain and Inheritance in JavaScript

This document defines the homework assignments from the ["Advanced JavaScript" Course @ Software University](https://softuni.bg/trainings/1099/Advanced-JavaScript-March-2015). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

## 2D Geometry Structure – Pseudo-Classical Model

Define a function constructor **Shape**. Define the following types of shapes:

* **Circle** – holds a center point **O** with coordinates **x** and **y** (written for simplicity as **O(x, y)**). Also has a radius **r** (number) and a **color** (in hexadecimal format)
* **Rectangle** – holds the coordinates of its top left corner **A(x, y)**, **width** (number), **height** (number) and **color** (in hexadecimal format)
* **Triangle** – consists of three points **A(x, y)**, **B(x, y)** and **C(x, y)**. Also has a **color** (in hexadecimal format)
* **Line** – holds two points **A(x, y)** and **B(x, y)**, has a **color** (in hexadecimal format)
* **Segment** – consists of two endpoints **A(x, y)** and **B(x, y)**, also has a **color** (in hexadecimal format)

Override the **toString()** method to print information about each object. Add all shapes **in a module**.

Debug your code and see what the prototype is and what the \_\_proto\_\_ is in each object and function constructor. Use the browser developer tools and watch the locals.

You should submit a JavaScript file (or files) as a part of your homework.

## 2D Geometry Structure – Prototypal Model

Your task is to define a prototype for all types from the previous task using prototypal model.

Override the **toString()** method to print information about each object. Add all shapes **in a module**.

Debug your code and see what the \_\_proto\_\_ is in each object. Use the browser developer tools and watch the locals.

You should submit a JavaScript file (or files) as a part of your homework.

## \* 2D Geometry - Canvas

Your task is to use one of the previous structures (by your choice) and to add a **draw()** method for each shape to draw itself on Canvas.

Design your object-oriented hierarchy to avoid code repetition and increase abstraction. Encapsulate all data and validate the input (for example, you cannot create a triangle with only two points).

You can use Canvas API tutorial from <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial>

Basics steps:

1. You should create canvas element in your HTML page with width and height.
2. In your JavaScript file you should get a canvas element and assign it to a variable.
   * Example: var canvas = document.getElementById("canvas");
3. You should get context from this variable
   * Example: var context = canvas.getContext("2d");
4. You can use context methods for drawing shapes and drawing path method for the other shapes:
   * context.fillRect(), context.strokeRect(), context.clearRect()
   * context.beginPath(), context.closePath(), context.stroke(), context.fill()
   * <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Drawing_shapes>

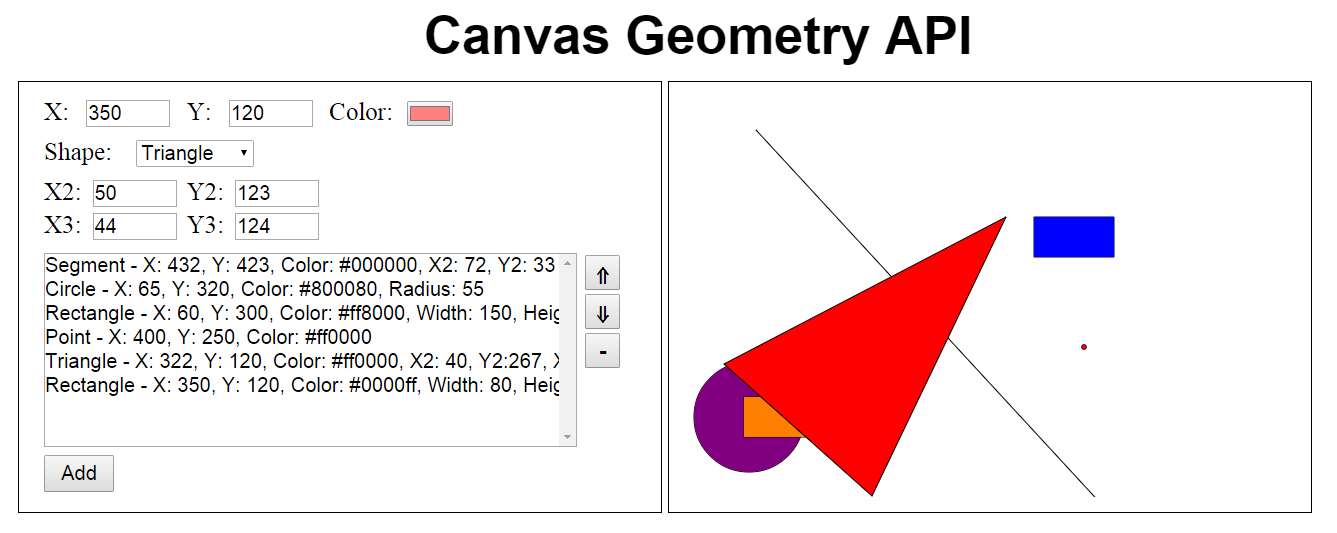
You should submit a JavaScript file (or files) as a part of your homework.

## \* Canvas Geometry API

Use your previous geometry module to create another module that draws shapes on Canvas. The API should support the following:

* **Creating** lines, segments, circles, triangles and rectangles
* **Drawing** the created shapes on Canvas
* **Receiving** input data from a form
  + The type of the chosen shape affects the form input fields depending on what the shape needs to have
* **Displaying information** about all drawn shapes in a user-friendly way
* **Removing** shapes from the canvas (using the "-" button)
* **\* Changing the z-index** of the shape (i. e. setting a shape over another and vice versa) (using the arrow buttons)

The image below is only a sample. Styling the page is optional.



You should submit a JavaScript file (or files) with HTML and CSS as a part of your homework.