**ЛАБОРАТОРНАЯ РАБОТА 3 HTML 5. РАБОТА С ГРАФИКОЙ. CANVAS. АНИМАЦИЯ**

**Цель работы:** ознакомление с элементом <canvas> и создание простейших JavaScript-функций для генерации изображений в рамках элемента <canvas>, формирование базовых навыков рисования простых фигур на холсте.

**Задания**

Создайте пункт при регистрации в виде подтверждения о необходимости нарисовать фигуру

**Контрольные вопросы**

1. При помощи какого тега осуществляется размещение холста на странице?

<canvas></canvas>

1. С какой целью в HTML-документах проводится стилизация холста?

Элемент <canvas> имеет только два атрибута - **ширину** и **высоту**. Оба они не обязательны и могут быть выставлены с использованием свойств [DOM](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model). Если атрибуты высоты и ширины не установлены, canvas будет по умолчанию шириной **300 пикселей** и в высоту **150 пикселей**.

1. Охарактеризуйте процесс рисования на холсте. Назовите функции, которые для этого используются.

[<canvas>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/canvas) поддерживает только одну примитивную фигуру: прямоугольник. Все другие фигуры должны быть созданы комбинацией одного или большего количества контуров (paths), набором точек, соединённых в линии.

* [fillRect(x, y, width, height)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/fillRect) - Рисование заполненного прямоугольника.
* [strokeRect(x, y, width, height)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/strokeRect)- Рисование прямоугольного контура.
* [clearRect(x, y, width, height)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/clearRect)- Очистка  прямоугольной области, делая содержимое совершенно прозрачным.

1. Приведите алгоритм рисования простейших фигур на холсте.

* Создание контура
* Рисование контура
* Закрытие контура
* Обвести или залить контур

1. Каким образом осуществляется рисование дуг?

Для рисования дуг и окружностей, используем методы arc() и arcTo().

* [arc(x, y, radius, startAngle, endAngle, anticlockwise)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/arc) - Рисуем дугу с центром в точке (x,y) радиусом radius, начиная с угла startAngle и заканчивая в endAngle в направлении против часовой стрелки anticlockwise (по умолчанию по ходу движения часовой стрелки).
* [arcTo(x1, y1, x2, y2, radius)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/arcTo) - Рисуем дугу с заданными контрольными точками и радиусом, соединяя эти точки прямой линией.

1. Алгоритм рисования «кривой Безье».

* [quadraticCurveTo(cp1x, cp1y, x, y)](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/quadraticCurveTo) - Рисуется квадратичная кривая Безье с текущей позиции пера в конечную точку с координатами x и y, используя контрольную точку с координатами cp1x и cp1y.
* [bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)](https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/bezierCurveTo) - Рисуется кубическая кривая Безье с текущей позиции пера в конечную точку с координатами x и y, используя две контрольные точки с координатами (cp1x, cp1y) и (cp2x, cp2y).

**Вывод**: в ходе работы я ознакомилась с элементом <canvas> и создание простейших JavaScript-функций для генерации изображений в рамках элемента <canvas>, формирование базовых навыков рисования простых фигур на холсте.