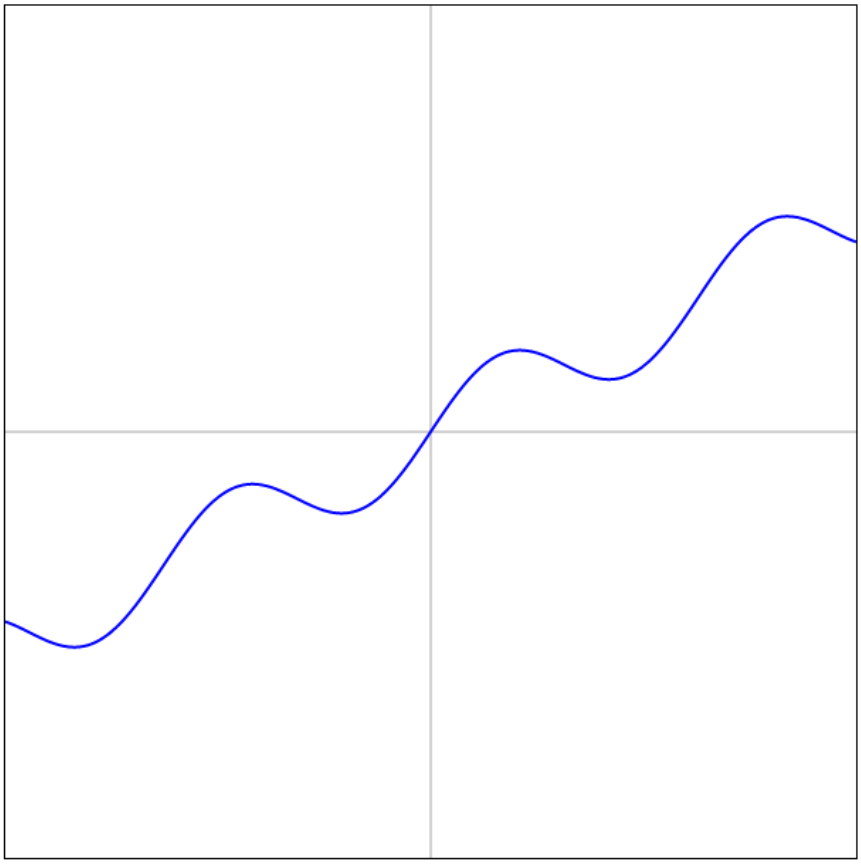
Łukasz Zawodziński gr. 3
nr Indexu: 136699

Interaktywna grafika i prezentacja danych

Laboratorium 7

Podstawy obsługi Canvas

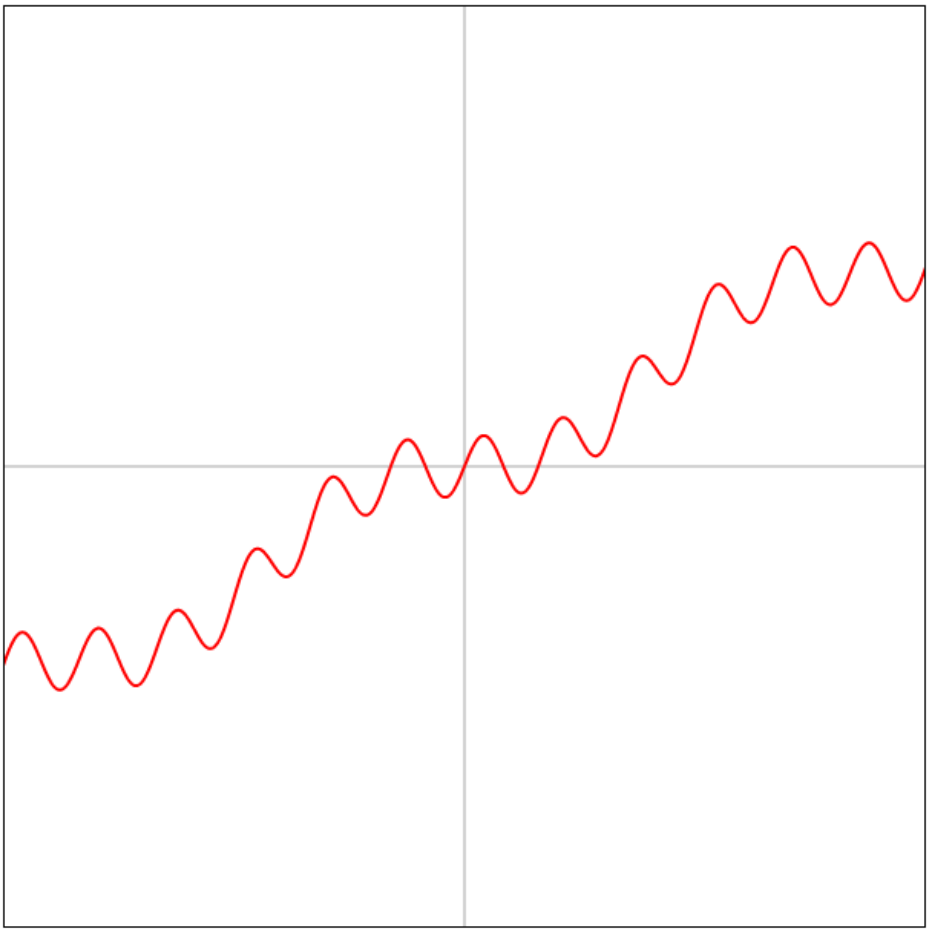
Zadanie Wykres 1



```
 <canvas id="myCanvas0" width="600" height="600" style="border:1px solid
#000000;"></canvas>

    <script>
```

```
const canvas = document.getElementById("myCanvas0");
const ctx = canvas.getContext("2d");

var scaleX = 30;
var scaleY = 30;
const c = 0;

const originX = canvas.width / 2;
const originY = canvas.height / 2;

ctx.translate(originX, originY);
ctx.scale(1, -1);

ctx.save();
ctx.strokeStyle = "#aaa";
ctx.lineWidth = 1;

ctx.beginPath();
ctx.moveTo(-originX, 0);
ctx.lineTo(originX, 0);
ctx.stroke();

ctx.beginPath();
ctx.moveTo(0, -originY);
ctx.lineTo(0, originY);
ctx.stroke();
ctx.restore();


ctx.beginPath();
ctx.lineWidth = 2;
ctx.strokeStyle = "blue";

const step = 0.01;
var firstPoint = true;

for (let x = -10; x <= 10; x += step) {
    const y = Math.sin(x) + 0.5 * x - c;

    const px = x * scaleX;
    const py = y * scaleY;

    if (firstPoint) {
    ctx.moveTo(px, py);
    firstPoint = false;
    } else {
    ctx.lineTo(px, py);
    }
}

ctx.stroke();

</script>
```

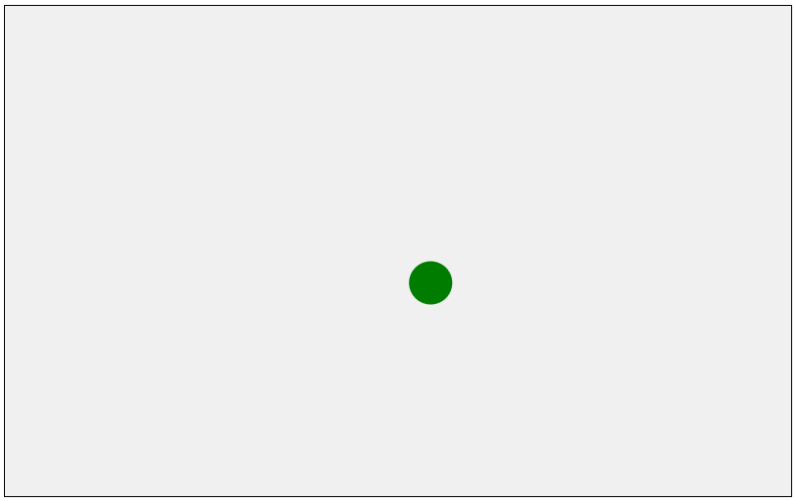
Zadanie Wykres 2



```
<canvas id="myCanvas1" width="600" height="600" style="border:1px solid
#000000;"></canvas>

    <script>
        const canvas1 = document.getElementById("myCanvas1");
        const ctx1 = canvas1.getContext("2d");

        var scaleX = 40;
        var scaleY = 40;

        ctx1.translate(originX, originY);
        ctx.scale(1, -1);

        ctx1.save();
        ctx1.strokeStyle = "#aaa";
        ctx1.lineWidth = 1;

        // Oś X
        ctx1.beginPath();
        ctx1.moveTo(-originX, 0);
        ctx1.lineTo(originX, 0);
        ctx1.stroke();
```

```
        // Oś Y
        ctx1.beginPath();
        ctx1.moveTo(0, -originY);
        ctx1.lineTo(0, originY);
        ctx1.stroke();
        ctx1.restore();


        ctx1.beginPath();
        ctx1.lineWidth = 2;
        ctx1.strokeStyle = "red";

        var firstPoint = true;

        for (let x = -10; x <= 10; x += step) {
            const y = Math.sin(2 * x) * Math.cos(3 * x) + 0.5 * x - c;

            const px = x * scaleX;
            const py = y * scaleY;

            if (firstPoint) {
            ctx1.moveTo(px, -py);
            firstPoint = false;
            } else {
            ctx1.lineTo(px, -py);
            }
        }
        ctx1.stroke();
    </script>
```

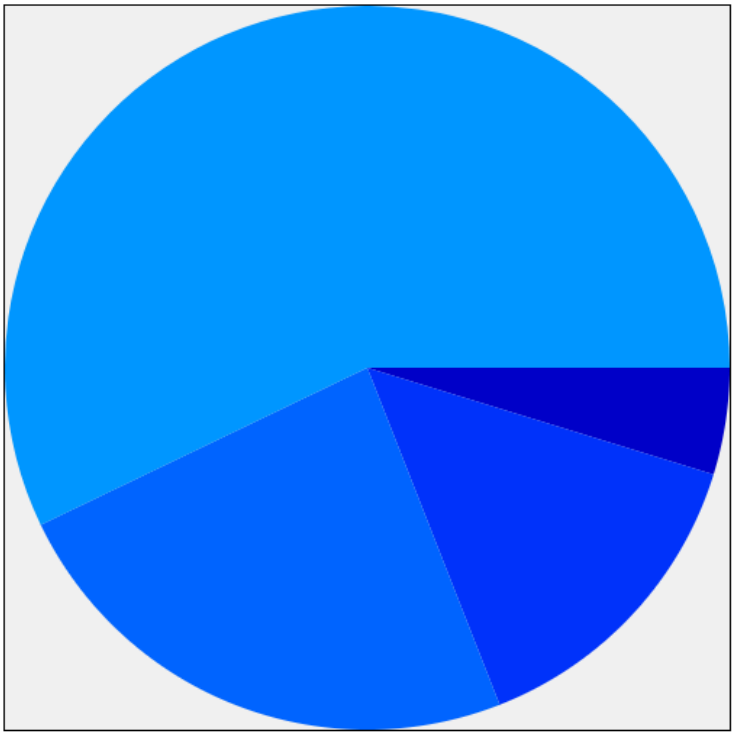
Animacja Piłki



```
<canvas id="AnimationCanvas" width="800" height="500"></canvas>

  <script>
    window.onload = function () {
      var canvas = document.getElementById("AnimationCanvas");
      var context = canvas.getContext("2d");

      var last_time, stop, linear_speed;
      var ballRadius = 22;
      var ballX = 100, ballY = 200;
      var speedX = 400, speedY = 300;

      function InitAnimation() {
        stop = false;
        var date = new Date();
        last_time = date.getTime();
        linear_speed = 200;

        window.requestAnimationFrame(drawAnimation);
      }


      function drawAnimation() {
        context.clearRect(0, 0, canvas.width, canvas.height);

        var date = new Date();
        var time_interval = date.getTime() - last_time;

        var distanceX = speedX * time_interval / 1000;
        var distanceY = speedY * time_interval / 1000;

        ballX += distanceX;
        ballY += distanceY;

        if (ballX + ballRadius >= canvas.width || ballX - ballRadius < 0) {
            speedX = -speedX;
            context.fillStyle = "blue";

        }
        if (ballY + ballRadius >= canvas.height || ballY - ballRadius < 0) {
          speedY = -speedY;
          context.fillStyle = "green";

        }

        context.save();
        context.beginPath();
        context.arc(ballX, ballY, ballRadius, 0, 2 * Math.PI);
        <!-- context.fillStyle = "red"; -->
        context.fill();
        context.closePath();
        context.restore();


        last_time = date.getTime();
        if (!stop) window.requestAnimationFrame(drawAnimation);
      }
```

```
    InitAnimation();
  }
</script>
```

Wykres kołowy

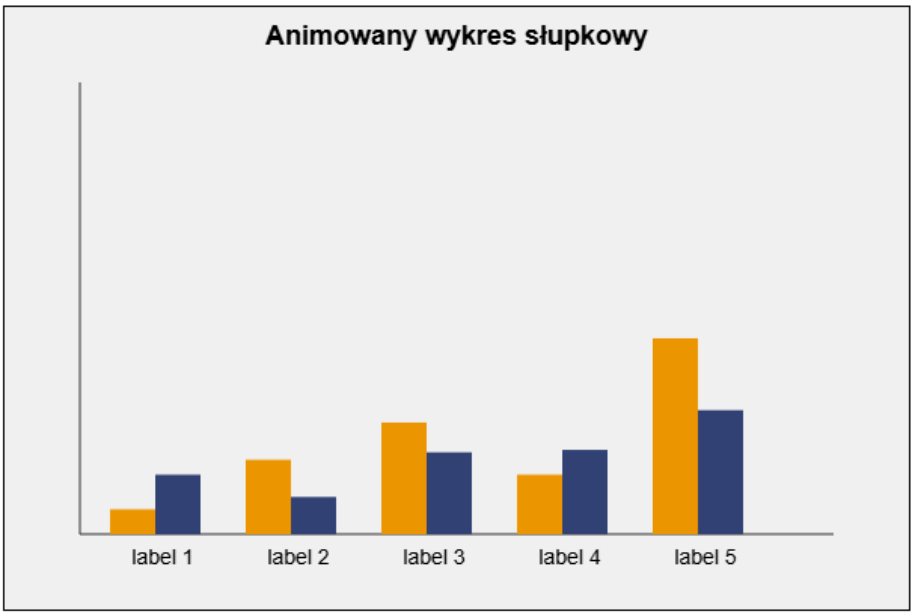


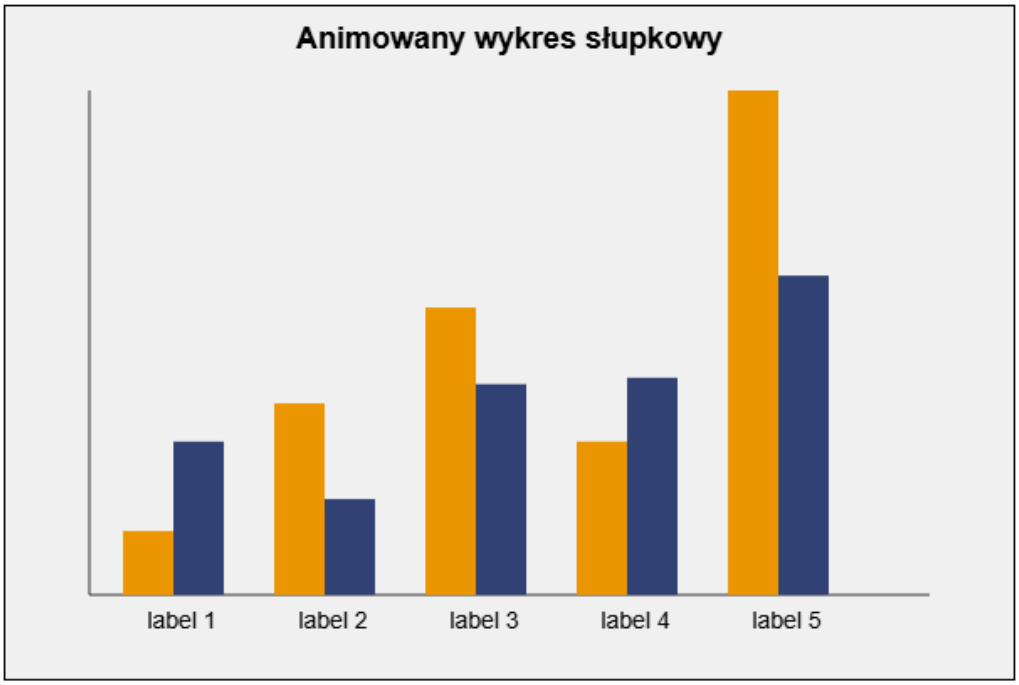
```
<canvas id="myCanvas2" width="500" height="500"></canvas>

  <script>
```

```
    const canvas2 = document.getElementById("myCanvas2");
    const ctx2 = canvas2.getContext("2d");

    var tab = [10, 30, 50, 120];
    var total = tab.reduce(function (sum, value) {
      return sum + value;
    }, 0);

    var radius = 250;
    var centerX = canvas2.width / 2;
    var centerY = canvas2.height / 2;
    var startAngle = 0;
    var currentAngle = startAngle;
    var animationDuration = 2000;
    var startTime = null;


    function animatePieChart(timestamp) {
      if (!startTime) startTime = timestamp;

      var progress = (timestamp - startTime) / animationDuration;


      if (progress > 1) {
        progress = 1;
      }

      ctx2.clearRect(0, 0, canvas2.width, canvas2.height);

      currentAngle = startAngle;

      for (var i = 0; i < tab.length; i++) {
        var value = tab[i];
        var sliceAngle = (value / total) * (2 * Math.PI);

        var targetAngle = sliceAngle * progress;

        ctx2.fillStyle = 'rgba(0 , ' + (0 + i * 50) + ', ' + (200 + i * 50) +
', 1)';

        ctx2.beginPath();
        ctx2.lineTo(centerX, centerY);
        ctx2.arc(centerX, centerY, radius, currentAngle, currentAngle +
targetAngle);
        ctx2.closePath();
        ctx2.fill();

        currentAngle += sliceAngle;
      }

      if (progress < 1) {
        window.requestAnimationFrame(animatePieChart);
      }
    }

    window.requestAnimationFrame(animatePieChart);
  </script>
```

Wykres słupkowy

animacja:






```
<canvas id="barChartCanvas" width="600" height="400"></canvas>
    <script>
      var dane_wykresu = {
        serie: [
          [10, 30, 45, 24, 79],
```

```
      [24, 15, 33, 34, 50],
    ],
    etykiety: ["label 1", "label 2", "label 3", "label 4", "label 5"],
    tytul: "Animowany wykres słupkowy",
    type: "wykres_slupkowy",
  };

  const canvas = document.getElementById("barChartCanvas");
  const ctx = canvas.getContext("2d");
  const width = canvas.width;
  const height = canvas.height;
  const margin = 50;
  const barWidth = 30;
  const barGap = 20;
  const seriesGap = 10;
  const numSeries = dane_wykresu.serie.length;
  const numBars = dane_wykresu.etykiety.length;

  let maxVal = 0;
  dane_wykresu.serie.forEach((serie) => {
    serie.forEach((val) => {
      if (val > maxVal) maxVal = val;
    });
  });

  // Animation
  const animationDuration = 2000;
  let startTime = null;

  function drawAxes() {
    ctx.save();
    ctx.strokeStyle = "#333";
    ctx.beginPath();
    ctx.moveTo(margin, margin);
    ctx.lineTo(margin, height - margin);
    ctx.lineTo(width - margin, height - margin);
    ctx.stroke();
    ctx.restore();
  }

  function drawLabels() {
    ctx.save();
    ctx.font = "14px Arial";
    ctx.textAlign = "center";
    for (let i = 0; i < numBars; i++) {
      let x =
        margin +
        barGap +
        i * (numSeries * barWidth + seriesGap + barGap) +
        (numSeries * barWidth + seriesGap) / 2;
      ctx.fillText(dane_wykresu.etykiety[i], x, height - margin + 20);
    }

    ctx.font = "bold 18px Arial";
    ctx.textAlign = "center";
    ctx.fillText(dane_wykresu.tytul, width / 2, margin / 2);
    ctx.restore();
```

```
    }

    function animateBars(timestamp) {
      if (!startTime) startTime = timestamp;
      let progress = (timestamp - startTime) / animationDuration;
      if (progress > 1) progress = 1;

      ctx.clearRect(0, 0, width, height);
      drawAxes();
      drawLabels();

      for (let i = 0; i < numBars; i++) {
        for (let j = 0; j < numSeries; j++) {
          let val = dane_wykresu.serie[j][i];
          let scaledHeight =
            (val / maxVal) * (height - 2 * margin) * progress;
          let x =
            margin +
            barGap +
            i * (numSeries * barWidth + seriesGap + barGap) +
            j * barWidth;
          let y = height - margin - scaledHeight;
          ctx.save();
          ctx.fillStyle = j === 0 ? "#ee9900" : "#334577";
          ctx.fillRect(x, y, barWidth, scaledHeight);
          ctx.restore();
        }
      }

      if (progress < 1) {
        window.requestAnimationFrame(animateBars);
      }
    }

    window.requestAnimationFrame(animateBars);
  </script>
```