JavaScript Grp1

班級：五專資工三甲

組別：實習課第6組

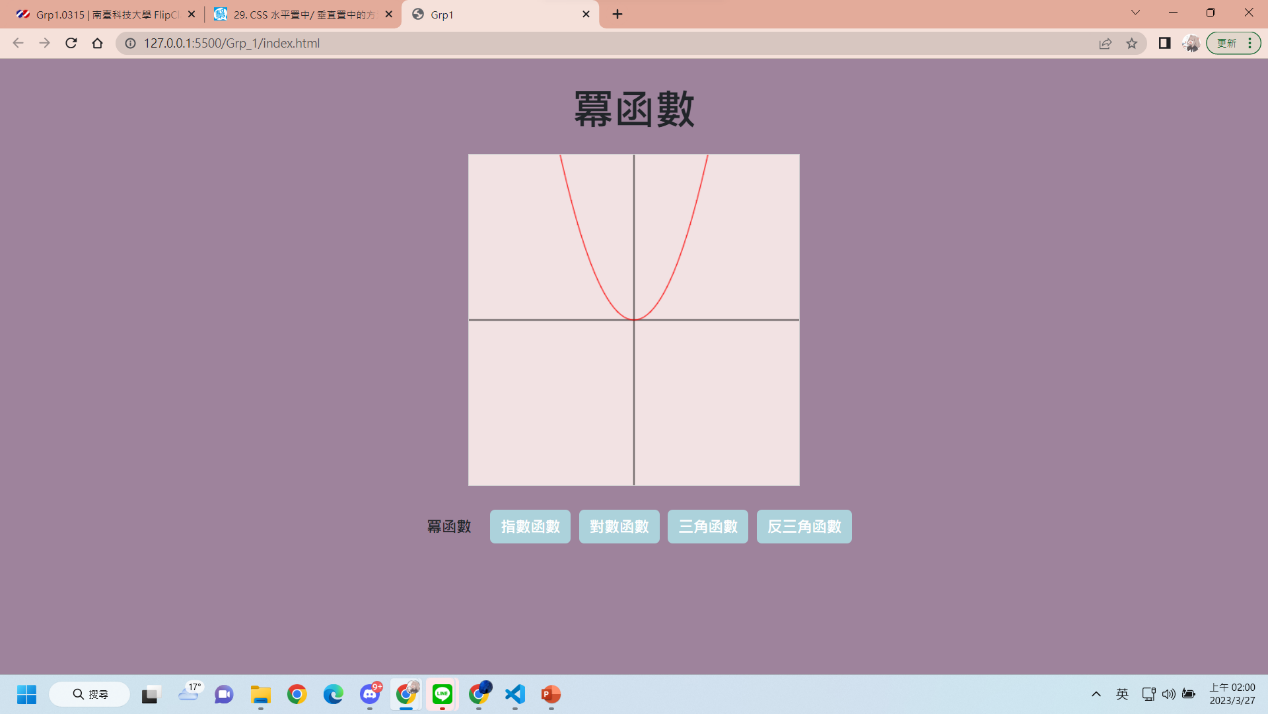
組長：陳紋誼

組員：陳品瑋、鄧光易、蔡世偉

1. 題目



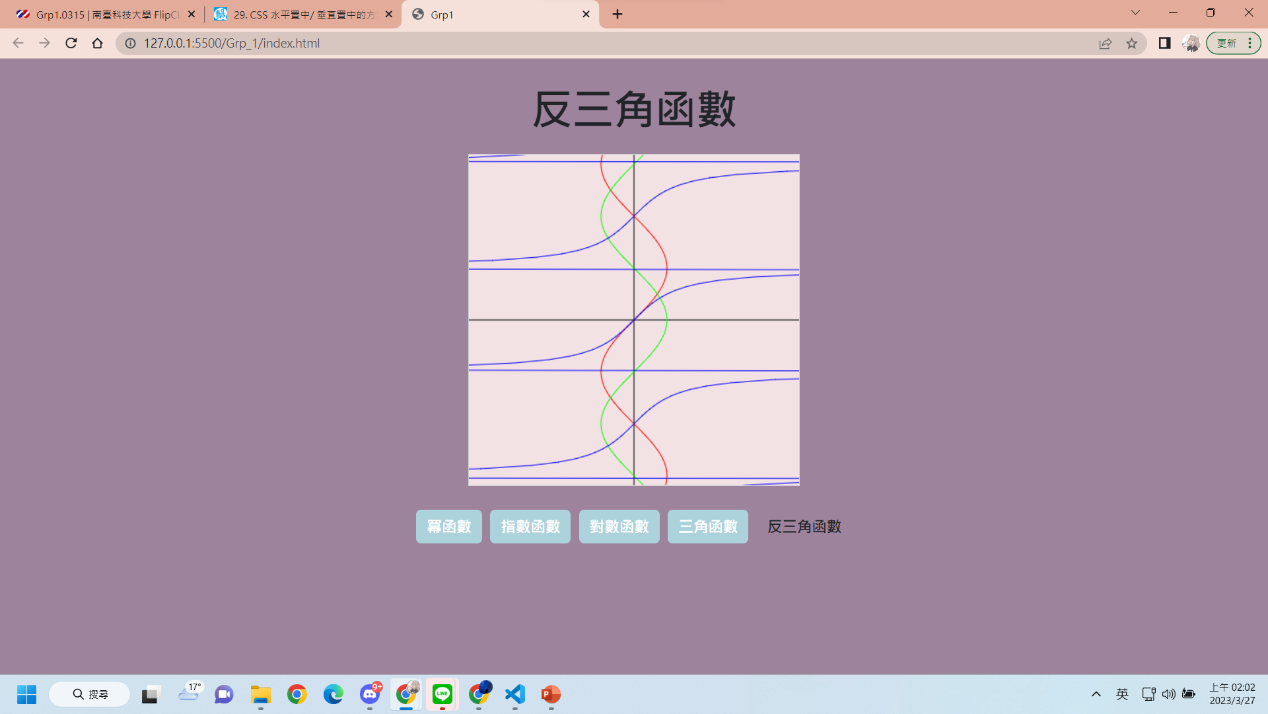
1. 截圖











1. 說明

按鈕使用bootstrap的樣式。

使用者按下任意函數的按鈕，就會呼叫該函數的function，使canvas畫布畫出該函數的圖形。

X和Y的正負值都介於-10~10之間，數字一格的大小是40px，平均分配在400\*400大小的畫布裡。

每個函數的function都會呼叫計算該函數的Math方法再一格一格繪出圖形，冪函數使用Math.pow()計算y的值，指數函數使用Math.exp()計算y的值，對數函數使用Math.log()計算y的值，三角函數分別使用Math.sin()、Math.cos()、Math.tan計算三個三角函數的y值，反三角函數分別使用Math.sin()、Math.cos()、Math.tan計算三個三角函數的x值，使其與三角函數的結果是反過來的。

透過在每個函數function的開頭執行canvas.width = canvas.width，使canvas畫布被重置寬度，達到清空畫布的效果，防止切換按鈕後圖形會重疊或是不再基準點上被執行。

1. .html程式

<!DOCTYPE html>

<html>

<head>

    <title>Grp1</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-qKXV1j0HvMUeCBQ+QVp7JcfGl760yU08IQ+GpUo5hlbpg51QRiuqHAJz8+BrxE/N"

        crossorigin="anonymous"></script>

</head>

<style>

    body {

        background-color: rgb(158, 131, 156);

    }

    .btn {

        font-size: 18px;

        font-weight: bolder;

        background-color: rgb(172, 210, 219);

        color: rgb(255, 255, 255);

        transition: 0.5s;

        margin: 5px;

    }

</style>

<body>

    <br>

    <p id="js" align="center" style="font-size: 50px;font-weight: bolder"></p>

    <canvas id="myCanvas" width="400" height="400"

        style="border:1px solid #d3d3d3; display: block; margin: 0 auto; background-color: rgb(242, 226, 227);"></canvas>

    <br>

    <div id="Button" style="display: block; margin: 0 auto; width:auto; height:auto;">

        <div style="display: flex; justify-content: center;">

            <button class="btn" onclick="PowerFunction()">冪函數</button>

            <button class="btn" onclick="ExponentialFunction()">指數函數</button>

            <button class="btn" onclick="LogarithmFunction()">對數函數</button>

            <button class="btn" onclick="TrigonometricFunctions()">三角函數</button>

            <button class="btn" onclick="InverseTrigonometricFunctions()">反三角函數</button>

        </div>

    </div>

    <script>

        var canvas = document.getElementById("myCanvas");

        var ctx = canvas.getContext("2d");

        var xMin = -10; // x 的最小值

        var xMax = 10; // x 的最大值

        var yMin = -10; // y 的最小值

        var yMax = 10; // y 的最大值

        var scale = 40; // 每個單位格的大小

        var step = 0.1; // x 的增量

        function PowerFunction() {

            canvas.width = canvas.width;

            document.getElementById("js").innerHTML = "冪函數";

            ctx.translate(canvas.width / 2, canvas.height / 2); // 將坐標系移動到畫布中心

            ctx.beginPath();

            ctx.moveTo(xMin \* scale, 0);

            ctx.lineTo(xMax \* scale, 0);

            ctx.moveTo(0, yMin \* scale);

            ctx.lineTo(0, yMax \* scale);

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#ff0000"; // 設置線條顏色

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.pow(x, 2); // 計算 y 的值

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

        }

        function ExponentialFunction() {

            canvas.width = canvas.width;

            document.getElementById("js").innerHTML = "指數函數";

            ctx.translate(canvas.width / 2, canvas.height / 2);

            ctx.beginPath();

            ctx.moveTo(xMin \* scale, 0);

            ctx.lineTo(xMax \* scale, 0);

            ctx.moveTo(0, yMin \* scale);

            ctx.lineTo(0, yMax \* scale);

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#ff0000";

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.exp(x);

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

        }

        function LogarithmFunction() {

            canvas.width = canvas.width;

            document.getElementById("js").innerHTML = "對數函數";

            ctx.translate(canvas.width / 2, canvas.height / 2);

            ctx.beginPath();

            ctx.moveTo(xMin \* scale, 0);

            ctx.lineTo(xMax \* scale, 0);

            ctx.moveTo(0, yMin \* scale);

            ctx.lineTo(0, yMax \* scale);

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#ff0000";

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.log(x);

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

        }

        function TrigonometricFunctions() {

            canvas.width = canvas.width;

            document.getElementById("js").innerHTML = "三角函數";

            ctx.translate(canvas.width / 2, canvas.height / 2);

            ctx.beginPath();

            ctx.moveTo(xMin \* scale, 0);

            ctx.lineTo(xMax \* scale, 0);

            ctx.moveTo(0, yMin \* scale);

            ctx.lineTo(0, yMax \* scale);

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#ff0000";

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.sin(x);

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#00ff00";

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.cos(x);

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#0000ff";

            for (var x = xMin; x < xMax; x += step) {

                var y = Math.tan(x);

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

        }

        function InverseTrigonometricFunctions() {

            canvas.width = canvas.width;

            document.getElementById("js").innerHTML = "反三角函數";

            ctx.translate(canvas.width / 2, canvas.height / 2);

            ctx.beginPath();

            ctx.moveTo(xMin \* scale, 0);

            ctx.lineTo(xMax \* scale, 0);

            ctx.moveTo(0, yMin \* scale);

            ctx.lineTo(0, yMax \* scale);

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#ff0000";

            for (var y = yMin; y < yMax; y += step) {

                var x = Math.sin(y); // 計算 x 的值

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#00ff00";

            for (var y = yMin; y < yMax; y += step) {

                var x = Math.cos(y); // 計算 x 的值

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

            ctx.beginPath();

            ctx.strokeStyle = "#0000ff";

            for (var y = yMin; y < yMax; y += step) {

                var x = Math.tan(y); // 計算 x 的值

                ctx.lineTo(x \* scale, -y \* scale);

            }

            ctx.stroke();

        }

    </script>

</body>

</html>