

# Javascript examples

Nasser M. Abbasi

March 6, 2021

Compiled on February 7, 2024 at 11:56pm

## Contents

1	XY plot using flot library	1
2	Click on button, box slides, using requestAnimationFrame	2
3	Direct plot of $\sin(x)$ using canvas	4
4	Direct plot of $\sin(x)$ using canvas, using slider to read x range	5
5	Direct plot using canvas, menu to select function, using slider to read x range	8
6	Direct plot using canvas, menu to select function, using slider to read x range, slider to change sampling	10
7	Showing how to start and stop an animation and change speed	13

## 1 XY plot using flot library

This shows how to make a plot of a  $\sin(x)$  function. The library used is <http://www.flotcharts.org/>

Clicking on ex1.htm runs the code. The source code is

```
<html lang="en">
<head>
<title>
  Simple Javascript example 1
</title>
<meta charset="UTF-8">
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Flot Examples: Basic Options</title>
<script language="javascript" type="text/javascript" src="/scripts/flot/jquery.js"></script>
<script language="javascript" type="text/javascript" src="/scripts/flot/jquery.flot"></script>
<script type="text/javascript">

$(function() {

var options = {
    series: {
        lines: { show: true, fill: false, lineWidth:1},
        points: { show: false },
        color:"rgb(0, 0, 0)"
    }
};

    var data = [];
    for (var i = -2*Math.PI; i < 2*Math.PI; i += Math.PI/10) {
        data.push([i, Math.sin(i)]);
    }

    $.plot("#plot", [data], options);

    // Add the Flot version string to the footer

    $("#footer").prepend("Flot " + $.plot.version + " &ndash; ");
});

</script>
</head>
<body>

Example showing how to plot a sin function using flot library.

<div id="plot" style="width:400px;height:300px"></div>

</body>
</html>

```

## 2 Click on button, box slides, using requestAnimationFrame

Clicking on HTML runs the code. The source code is

```
<!DOCTYPE html>
<html>
<!-- simple animation of box moving. Nasser M. Abbasi, January 15, 2020 -->
<body>

<p>
<button onclick="process()">Click Me</button>
</p>

<div style="width: 400px; height: 50px; position: relative; background-color:lightblue;">
  <div id ="myAnimation", style="width: 50px; height: 50px; position: absolute; background
</div>

<p>current position: <span id="position_text"></span></p>
<p>current time: <span id="time_text"></span> (seconds) </p>

<script>
"use strict";
var g_is_running = false;
var g_starting_time =0;
var g_element = document.getElementById("myAnimation");
var g_max = 400-50;
var g_id;

function process()
{
  g_is_running=false;
  g_id = window.requestAnimationFrame(do_it);
}

function do_it(time_stamp)
{
  var progress      = 0;
  var new_position = 0;
```

```

if (!g_is_running)
{
    g_is_running = true;
    g_starting_time = time_stamp;
}
else
{
    progress = time_stamp - g_starting_time;
    new_position = Math.min(progress / 20, g_max);
}

g_element.style.left = new_position + 'px';

document.getElementById("position_text").innerHTML = new_position.toFixed(2);
document.getElementById("time_text").innerHTML = ((time_stamp-g_starting_time)/1000).toFixed(2);

if (new_position < g_max)
{
    window.requestAnimationFrame(do_it);
}
}

</script >

</html>

```

### 3 Direct plot of $\sin(x)$ using canvas

Clicking on HTML runs the code. The source code is

```

<!DOCTYPE html>
<!-- Draw sin(x) from 0 to 2 PI. By Nasser M. Abbasi. March 6, 2020 -->
<html>
<head>

</head>
<body>

<canvas id="canvas" width="300" height="200"></canvas>

```

```

<SCRIPT>
var canvas=document.getElementById('canvas');
var ctx=canvas.getContext('2d');

//Scale the width so it is 2 PI and scale the height so it
// goes from -1..1
var horizontal_scale = 2*Math.PI/canvas.width;
var vertical_scale    = canvas.height/2;

//Move the origin from top-left
ctx.translate(0, canvas.height/2);

// and correct the y-scale
ctx.scale(1, -1);

//Draw the sine wave
ctx.beginPath();
ctx.moveTo(0, 0);
for (let x = 0; x <= canvas.width; x=x+canvas.width/50)
{
    ctx.lineTo(x, vertical_scale*Math.sin(x*horizontal_scale) );
}
ctx.strokeStyle = "black";
ctx.lineWidth=2;
ctx.stroke();

// draw x-axis
ctx.beginPath();
ctx.moveTo(0, 0);
ctx.lineTo(canvas.width, 0 );
ctx.lineWidth=1;
ctx.strokeStyle = "red";
ctx.stroke();

// draw y-axis
ctx.beginPath();
ctx.moveTo(0, -canvas.height/2);
ctx.lineTo(0,canvas.height/2);
ctx.stroke();

```

```
</SCRIPT>

</body>
</html>
```

## 4 Direct plot of $\sin(x)$ using canvas, using slider to read x range

Clicking on HTML runs the code. The source code is

```
<!DOCTYPE html>
<!-- Draw sin(x) from 0 to 2 PI. By Nasser M. Abbasi. March 6, 2020 -->
<html>
<head>

</head>
<body>

<canvas id="canvas" width="300" height="200"></canvas>

<div>
  <label>x range (in units of PI)</label>
  <input type="range" id="to" value="2" min="1" max="10" step="1" oninput="update(thi
  <label id="range">2</label>
</div>

<SCRIPT>
var canvas=document.getElementById('canvas');
var ctx=canvas.getContext('2d');

//Move the origin from top-left
ctx.translate(0, canvas.height/2);
// and correct the y-scale
ctx.scale(1, -1);

function update(xrange)
{
//Scale the width so it is 2 PI and scale the hight so it
```

```

// goes from -1..1

document.getElementById('range').innerHTML=xrange; //read x-range

var horizontal_scale = xrange*Math.PI/canvas.width;
var vertical_scale    = canvas.height/2;

//These 4 lines is to clear canvas
ctx.save();
ctx.setTransform(1, 0, 0, 1, 0, 0);
ctx.clearRect(0, 0, canvas.width, canvas.height);
ctx.restore();

//Draw the sine wave
ctx.beginPath();
ctx.moveTo(0, 0);
for (let x = 0; x <= canvas.width; x=x+canvas.width/60)
{
    ctx.lineTo(x, vertical_scale*Math.sin(x*horizontal_scale) );
}
ctx.strokeStyle = "black";
ctx.lineWidth=2;
ctx.stroke();

// draw x-axis
ctx.beginPath();
ctx.moveTo(0, 0);
ctx.lineTo(canvas.width, 0 );
ctx.lineWidth=1;
ctx.strokeStyle = "red";
ctx.stroke();

// draw y-axis
ctx.beginPath();
ctx.moveTo(0, -canvas.height/2);
ctx.lineTo(0,canvas.height/2);
ctx.stroke();
}

window.onload = function ()
{

```

```

update(2)
}
</SCRIPT>

</body>
</html>

```

## 5 Direct plot using canvas, menu to select function, using slider to read x range

Clicking on HTML runs the code. The source code is

```

<!DOCTYPE html>
<!-- Draw sin(x) from 0 to 2 PI. By Nasser M. Abbasi. March 6, 2020 -->
<html>
<head>

</head>
<body>

<canvas id="canvas" width="300" height="200"></canvas>

<div class="tab">
  <button onclick="g_current_function=Math.sin; update()">sin</button>
  <button onclick="g_current_function=Math.cos; update()">cos</button>
  <button onclick="g_current_function=Math.tan; update()">tan</button>
</div>

<div>
  <label>x range (in units of PI)</label>
  <input type="range" id="to" value="2" min="1" max="10" step="1"
    oninput="document.getElementById('range').innerHTML=this.value; g_x_range=this.value;
  <label id="range">2</label>
</div>

<SCRIPT>
var canvas=document.getElementById('canvas');
var ctx=canvas.getContext('2d');

```



```

var g_current_function=Math.sin;
var g_x_range=document.getElementById('range').innerHTML;

//Move the origin from top-left
ctx.translate(0, canvas.height/2);
// and correct the y-scale
ctx.scale(1, -1);

function update()
{
//Scale the width so it is 2 PI and scale the height so it
// goes from -1..1

var horizontal_scale = g_x_range*Math.PI/canvas.width;
var vertical_scale   = canvas.height/2;

//console.log("enter update");
//console.log("g_current_function=",g_current_function);
//console.log("xrange=",xrange);
//console.log("g_current_function(Math.PI/2)",g_current_function(Math.PI/2));

//These 4 lines is to clear canvas
ctx.save();
ctx.setTransform(1, 0, 0, 1, 0, 0);
ctx.clearRect(0, 0, canvas.width, canvas.height);
ctx.restore();

//Draw the sine wave
ctx.beginPath();
ctx.moveTo(0, 0);
for (let x = 0; x <= canvas.width; x=x+canvas.width/60)
{
    //console.log("x=",x,"y=",vertical_scale*g_current_function(x*horizontal_scale));
    ctx.lineTo(x, vertical_scale*g_current_function(x*horizontal_scale) );
}
ctx.strokeStyle = "black";
ctx.lineWidth=2;
ctx.stroke();

// draw x-axis
ctx.beginPath();

```

```

ctx.moveTo(0, 0);
ctx.lineTo(canvas.width, 0 );
ctx.lineWidth=1;
ctx.strokeStyle = "red";
ctx.stroke();

// draw y-axis
ctx.beginPath();
ctx.moveTo(0, -canvas.height/2);
ctx.lineTo(0,canvas.height/2);
ctx.stroke();
}

window.onload = function ()
{
update()
}
</SCRIPT>

</body>
</html>

```

## 6 Direct plot using canvas, menu to select function, using slider to read x range, slider to change sampling

Clicking on HTML runs the code. The source code is

```

<!DOCTYPE html>
<!-- Draw trig fucntions. By Nasser M. Abbasi. March 7, 2020 -->
<html>
<head>

</head>
<body>

<canvas id="canvas" width="300" height="200"></canvas>

<div class="tab">

```

```

<button onclick="g_current_function=Math.sin; update()">sin</button>
<button onclick="g_current_function=Math.cos; update()">cos</button>
<button onclick="g_current_function=Math.tan; update()">tan</button>
</div>

<div>
  <P>
    <label>x range (in units of PI)</label>
    <input type="range" id="range_slider" value="2" min="1" max="10" step="1"
      oninput="document.getElementById('range').innerHTML=this.value; g_x_range=this.value;
    <label id="range">5</label>
  </P>
  <P>
    <label>sampling</label>
    <input type="range" id="sampling_slider" value="10" min="10" max="100" step="1"
      oninput="document.getElementById('sampling').innerHTML=this.value; g_sampling=this.va
    <label id="sampling">70</label>
  </P>
</div>

<SCRIPT>
var canvas=document.getElementById('canvas');
var ctx=canvas.getContext('2d');
var g_current_function=Math.sin;
var g_x_range=document.getElementById('range').innerHTML;
var g_sampling=document.getElementById('sampling').innerHTML;

//Move the origin from top-left
ctx.translate(canvas.width/2, canvas.height/2);
// and correct the y-scale
ctx.scale(1, -1);

function reset()
{
//These 4 lines is to clear canvas
ctx.save();
ctx.setTransform(1, 0, 0, 1, 0, 0);
ctx.clearRect(0, 0, canvas.width, canvas.height);
ctx.restore();
document.getElementById('sampling_slider').value=document.getElementById('sampling').innerHT

```

```

document.getElementById('range_slider').value=document.getElementById('range').innerHTML;
};

function update()
{
//Scale the width so it is 2 PI and scale the height so it
// goes from -1..1

var horizontal_scale = g_x_range*Math.PI/canvas.width;
var vertical_scale   = canvas.height/2;

//console.log("enter update");
//console.log("g_current_function=",g_current_function);
//console.log("xrange=",xrange);
console.log("g_current_function(Math.PI/2)",g_current_function(Math.PI/2));

reset();

//Draw the sine wave
ctx.beginPath();
ctx.moveTo(-canvas.width/2, vertical_scale*g_current_function(-canvas.width/2*horizontal_scale));
for (let x = -canvas.width/2; x <= canvas.width/2; x=x+canvas.width/g_sampling)
{
    console.log("x=",x,"y=",vertical_scale*g_current_function(x*horizontal_scale));
    ctx.lineTo(x, vertical_scale*g_current_function(x*horizontal_scale) );
}
ctx.strokeStyle = "black";
ctx.lineWidth=2;
ctx.stroke();

// draw x-axis
ctx.beginPath();
ctx.moveTo(-canvas.width/2, 0);
ctx.lineTo(canvas.width/2, 0 );
ctx.lineWidth=1;
ctx.strokeStyle = "red";
ctx.stroke();

// draw y-axis

```

```

ctx.beginPath();
ctx.moveTo(0, -canvas.height/2);
ctx.lineTo(0,canvas.height/2);
ctx.stroke();
}

window.onload = function ()
{
update()
}
</SCRIPT>

</body>
</html>

```

## 7 Showing how to start and stop an animation and change speed

This shows how to run animation, change speed of moving block and stop and restart it. This is a modified version of initial code I saw at w3schools web site with lots of changes I made to it.

Clicking on HTML runs the code. The source code is

```

<!DOCTYPE html>
<!-- By Nasser M. Abbasi. Nov 12, 2023-->
<html>
<style>
#container {
width: 400px;
height: 400px;
position: relative;
background: yellow;
}

#animate {
width: 50px;
height: 50px;
position: absolute;
background-color: red;

```

```

    }
</style>

<body>

  <div style=" margin: 8px; display: block;">
    <button onclick="restart()">restart</button>
    <button onclick="stop()">stop</button>
    <button onclick="do_continue()">continue</button>
    <button onclick="faster()">faster</button>
    <button onclick="slower()">slower</button>
  </div>

  <div id="container">
    <div id="animate"></div>
  </div>
  <p id="the_frame_rate"></p>
  <p id="position"></p>

  <script>
    let g_id = null;
    let g_delay = 5;
    let g_pos = 0;
    document.getElementById("position").innerText = "current position : (" + 0 + " , " + 0 +

    function stop()
    {
      if( g_id != null)
      {
        clearInterval(g_id);
        g_id = null;
      }
    }

    function do_continue()
    {
      if( g_id != null)
      {
        clearInterval(g_id);
        g_id = null;
      }
    }
  </script>

```

```

    move_it();
}

function slower()
{
    g_delay = g_delay + 1;
    var myP = document.getElementById("the_frame_rate");
    myP.innerHTML = "current frame rate : " + g_delay;

    if( g_id != null)
    {
        clearInterval(g_id);
        g_id = null;
    }
    move_it();
}

function faster()
{
    if( g_delay>0 )
    {
        g_delay = g_delay - 1;
        var myP = document.getElementById("the_frame_rate");
        myP.innerHTML = "current frame rate : " + g_delay;
    }

    if( g_id != null)
    {
        clearInterval(g_id);
        g_id = null;
    }

    move_it();
}

function restart()
{
    if( g_id != null)
    {

```

```

        clearInterval(g_id);
        g_id = null;
    }
    g_pos = 0;
    move_it();
}

function move_it()
{
    const elem = document.getElementById("animate");
    g_id = setInterval(frame, g_delay);
    function frame()
    {
        var myP1 = document.getElementById("position");
        var myP2 = document.getElementById("the_frame_rate");

        if (g_pos == 350)
        {
            //clearInterval(id);
            g_pos = 0;
            //g_id = null;
        }
        else
        {
            g_pos++;
            elem.style.top = g_pos + "px";
            elem.style.left = g_pos + "px";
        }
        myP1.innerHTML = "current position : (" + elem.style.top + " , " + elem.style.le
        myP2.innerHTML = "current frame rate : " + g_delay;
    }
}
</script>

</body>

</html>

```