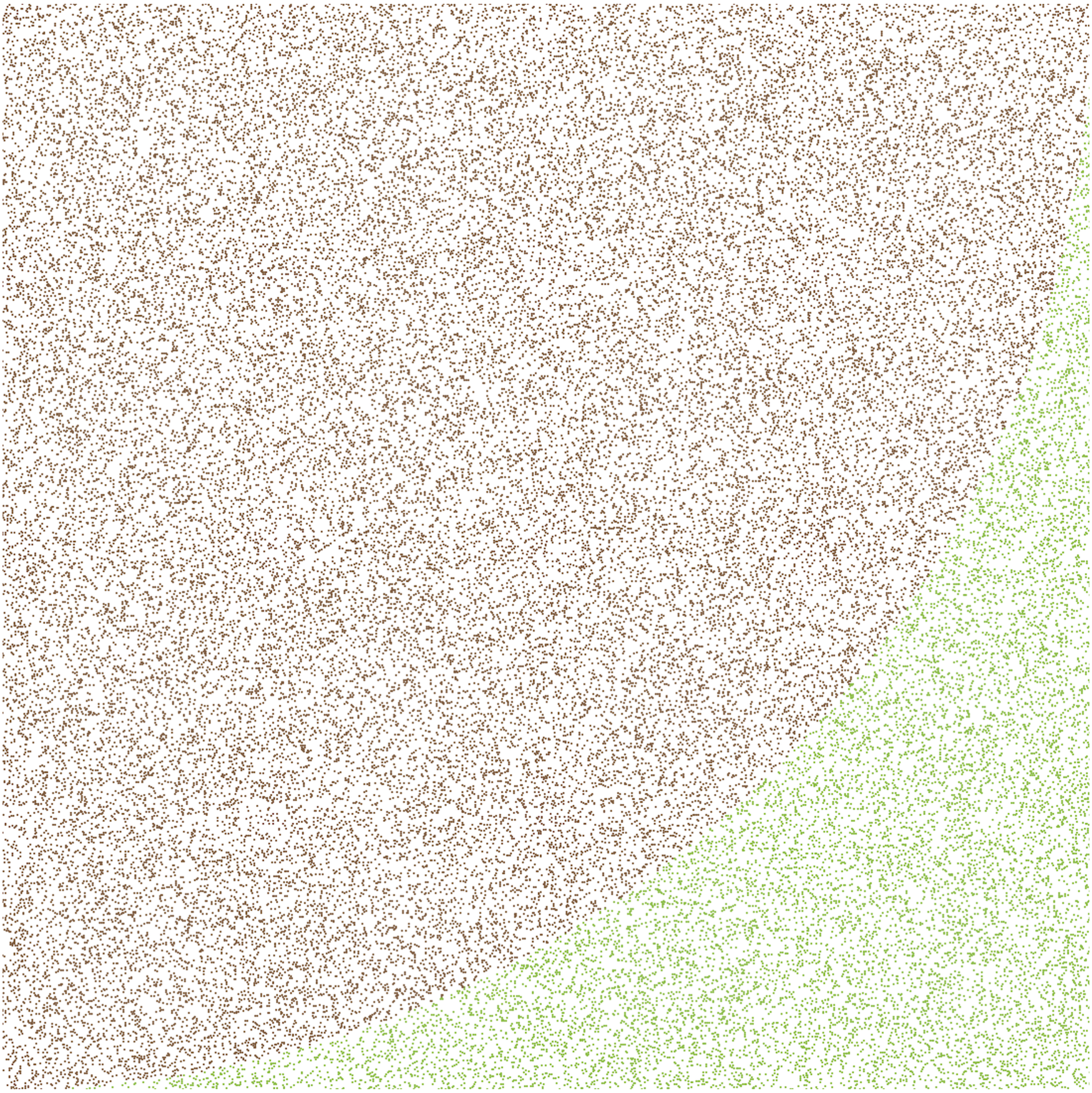
Estimating Pi using the Monte Carlo Method

Softeka - 19 January 2021



Introduction

One method to estimate the value of π (3.141592...) is by using a Monte Carlo method. In the program, we have a circle of radius 0.5, enclosed by a 1 × 1 square. The area of the circle is π sqr(r) = π / 4, the area of the square is 1. If we divide the area of the circle, by the area of the square we get π / 4.

We then generate a large number of uniformly distributed random points and plot them on the graph. These points can be in any position within the square i.e. between (0,0) and (1,1). If they fall within the circle, they are coloured red, otherwise they are coloured blue. We keep track of the total number of points, and the number of points that are inside the circle. If we divide the number of points within the circle, **hit** by the total number of points, **total**, we should get a value that is an approximation of the ratio of the areas we calculated above, π / 4.

In other words,

π / 4 ≈ hit / total

π ≈ 4* hit / total

When we only have a small number of points, the estimation is not very accurate, but when we have hundreds of thousands of points, we get much closer to the actual value - to within around 2 decimal places of accuracy. You can add points one at a time, or you can tick the "animate" checkbox to add many points to the graph very quickly.

Algorithm

The Monte Carlo method to calculate the Pi is implemented with javascript function and PHP. Basically the algorithm for **theLoop(x,y,i)** function that is calculating the data samples is a follows bellow:

1. Initialize setTimeout function which is executing every 10ms but printing values every 10s with variable **tout** .
2. Generate random point **x.**
3. Generate random point **y.**
4. Calculate **r = x*x + y*y**.
5. Increment **total** points (doesn't matter if they are inside or outside of the circle).
6. If **r <= 1100** (square width of the picture equal to radius of circle), increment hit points (points that are inside the circle) .
7. Calculate **pi = 4*(hit / total).**
8. If i > 0, repeat from 1 else exit the function **theLoop(x,y,i)**.

The javascript function that is doing the calculations is shown bellow

```
(function theLoop (x, y, i) {
  setTimeout(function () {
    console.log(i);

    var x =  Math.floor(Math.random() * (max - min + 1) + min);
    var y =  Math.floor(Math.random() * (max - min + 1) + min);

    var r = Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2));

    total++;
    if (r < 1100)
    {
     hit++;
     drawCoordinatesRed(context,x,y);
     ////console.log(x+y+r);
    }
    else
    {
     drawCoordinatesBlue(context,x,y);
    }

    if  (tout==10)
    {
       if (total < 10000)
       {
         pival.value = roundNumber(( hit / total )* 4, 10);
       }
       else
       {
         pival.value = roundNumber(( hit / total )* 4, 11);
       }
       tout=0;
    }
    else
    {
       tout++;
    }
       pointin.value    = hit;
       pointout.value = total;

    if (--i) {                  // If i > 0, keep going
      theLoop(x,y,i);    // Call the loop again
    }
  }, 10);
})(x, y, i);
```

The other components (HTML 5, CSS) of the web app are used for visualisation and controls to start stop or download the data:

- HTML 5 is used to create the canvas upon which we can draw the points;
- CSS is used for the styling the html elements.

The program can be executed on the following link https://nis.softeka.ch/ . The full source code is shown bellow and you can copy and paste into one file named <your file name>.php and execute on your web server:

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8" />
  <title>Random Pi</title>
  <meta charset="utf-8">
  <meta name="author" content="Softeka.ch">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <style>

  .inpbox {
    background-color: #f0f0f0;
    border: 0px solid;
    width: 150px;
    color: #000000;
    font-size: large;
    text-align: center;
    border-radius: 5px;
  }


  .inpboxexec {
    background-color: #f0f0f0;
    border: 0px solid;
    width: 75px;
    color: #000000;
    font-size: large;
    text-align: center;
    border-radius: 5px;
  }
  * {
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}



.header {
  overflow: hidden;
  background-color: #C0C0C0;
  padding: 20px 10px;
}

.header a {
  float: left;
  color: white;
  text-align: center;
  padding: 10px;
  text-decoration: none;
  font-size: 18px;
  line-height: 25px;
  border-radius: 4px;
}

.header a.logo {
  font-size: 25px;
  font-weight: bold;
}

.header a:hover {
  background-color: #ddd;
  color: black;
}

.header a.active {
  background-color: dodgerblue;
  color: white;
```

```
}

.header-right {
  float: center;
  color: white;
  text-align: center;
  vertical-align: center;
  padding: 10px;
  text-decoration: none;
  font-size: 25px;
}

/* Create two columns/boxes that floats next to each other */
nav {
  float: left;
  width: 15%;
  background: #fff;
  padding: 20px;
}

/* Style the list inside the menu */
nav ul {
  list-style-type: none;
  padding: 0;
}

article {
  float: left;
  padding: 20px;
  width: 70%;
  background-color: #fff;
}

/* Clear floats after the columns */
section:after {
  content: "";
  display: table;
  clear: both;
}

/* Style the footer */
footer {
  background-color: #C0C0C0;
  padding: 10px;
  text-align: center;
  color: white;
}

/* Responsive layout - makes the two columns/boxes stack on top of each other instead of next to each other, on small screens */
@media (max-width: 600px) {
  nav, article {
    width: 100%;
    height: auto;
  }
}

.wrap {
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
}

.stbutton {
  min-width: 100px;
  min-height: 60px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 22px;
  text-transform: uppercase;
  letter-spacing: 1.3px;
  font-weight: 350;
  color: #fff;
  background: #86BE3E;
background: linear-gradient(90deg, rgba(134,190,62,1) 0%, rgba(109,154,50,1) 100%);
  border: none;
  border-radius: 1000px;
  box-shadow: 12px 12px 24px rgba(242, 242, 242,.64);
  transition: all 0.3s ease-in-out 0s;
  cursor: pointer;
  outline: none;
  position: relative;
  padding: 10px;
  }

stbutton::before {
content: '';
  border-radius: 1000px;
  min-width: calc(100px + 12px);
  min-height: calc(60px + 12px);
  border: 3px solid #86BE3E;
  box-shadow: 0 0 60px rgba(0,255,203,.64);
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  opacity: 0;
  transition: all .3s ease-in-out 0s;
}

.stbutton:hover, .stbutton:focus {
  color: #fff;
  transform: translateY(-3px);
}

stbutton:hover::before, stbutton:focus::before {
  opacity: 1;
}

stbutton::after {
  content: '';
  width: 10px; height: 10px;
  border-radius: 100%;
  border: 3px solid #86BE3E;
  position: absolute;
  z-index: -1;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  animation: ring 1.5s infinite;
}

stbutton:hover::after, stbutton:focus::after {
  animation: none;
  display: none;
```

```
}

@keyframes ring {
  0% {
    width: 30px;
    height: 30px;
    opacity: 1;
  }
  100% {
    width: 100px;
    height: 100px;
    opacity: 0;
  }
}

.stbutton1 {
  min-width: 100px;
  min-height: 60px;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 22px;
  text-transform: uppercase;
  letter-spacing: 1.3px;
  font-weight: 350;
  color: #fff;
  background: #bd3d3e;
background: linear-gradient(90deg, rgba(189, 61, 61,1) 0%, rgba(154, 50, 50,1) 
100%);
  border: none;
  border-radius: 1000px;
  box-shadow: 12px 12px 24px rgba(242, 242, 242,.64);
  transition: all 0.3s ease-in-out 0s;
  cursor: pointer;
  outline: none;
  position: relative;
  padding: 10px;
  }

stbutton1::before {
content: '';
  border-radius: 1000px;
  min-width: calc(100px + 12px);
  min-height: calc(60px + 12px);
  border: 3px solid #bd3d3e;
  box-shadow: 0 0 60px rgba(242, 242, 242,.64);
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  opacity: 0;
  transition: all .3s ease-in-out 0s;
}

.stbutton1:hover, .stbutton1:focus {
  color: #fff;
  transform: translateY(-3px);
}

stbutton1:hover::before, stbutton1:focus::before {
  opacity: 1;
}

stbutton1::after {
  content: '';
  width: 10px; height: 10px;
  border-radius: 100%;
  border: 3px solid #bd3d3e;
  position: absolute;
  z-index: -1;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  animation: ring 1.5s infinite;
}

stbutton1:hover::after, stbutton1:focus::after {
  animation: none;
  display: none;
}

@keyframes ring {
  0% {
    width: 30px;
    height: 30px;
    opacity: 1;
```

```
          }
          100% {
            width: 100px;
            height: 100px;
            opacity: 0;
          }
        }
        .stbutton2 {
          min-width: 100px;
          min-height: 60px;
          font-family: Arial, Helvetica, sans-serif;
          font-size: 22px;
          text-transform: uppercase;
          letter-spacing: 1.3px;
          font-weight: 350;
          color: #fff;
          background: #f2f2f2;
        background: linear-gradient(90deg, rgba(242, 242, 242,1) 0%, rgba(217, 217, 217,1) 100%);
          border: none;
          border-radius: 1000px;
          box-shadow: 12px 12px 24px rgba(242, 242, 242,.64);
          transition: all 0.3s ease-in-out 0s;
          cursor: pointer;
          outline: none;
          position: relative;
          padding: 10px;
          }

        stbutton2::before {
        content: '';
          border-radius: 1000px;
          min-width: calc(100px + 12px);
          min-height: calc(60px + 12px);
          border: 3px solid #bd3d3e;
          box-shadow: 0 0 60px rgba(242, 242, 242,.64);
          position: absolute;
          top: 50%;
          left: 50%;
          transform: translate(-50%, -50%);
          opacity: 0;
          transition: all .3s ease-in-out 0s;
```

```
}

.stbutton2:hover, .stbutton2:focus {
  color: #fff;
  transform: translateY(-3px);
}

stbutton2:hover::before, stbutton2:focus::before {
  opacity: 1;
}

stbutton2::after {
  content: '';
  width: 10px; height: 10px;
  border-radius: 100%;
  border: 3px solid #bd3d3e;
  position: absolute;
  z-index: -1;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  animation: ring 1.5s infinite;
}

stbutton2:hover::after, stbutton2:focus::after {
  animation: none;
  display: none;
}

@keyframes ring {
  0% {
    width: 30px;
    height: 30px;
    opacity: 1;
  }
  100% {
    width: 100px;
    height: 100px;
    opacity: 0;
  }
}
```

```
          </style>


         </head>
         <body>


        <div class="header">
          <a class="logo" href="https://www.nis.edu.kz/en/" target="_blank" ><img
src="logo.jpeg" class="img-thumbnail" width="90" alt="University"></a>
          <div class="header-right">
              ESTIMATE PI
              </br>
              <p>
               <label for="pi" style="font-size: large;">pi:</label>
               <input type="text" name="pi" id="Text1" class="inpbox" value="0"
readonly="true" />

               &nbsp;&nbsp;&nbsp;&nbsp;<label for="pi1" style="font-size: large;">Drops in
the quarter-circle:</label>
               <input type="text" name="pi1" id="PointIn" class="inpbox" value="0"
readonly="true" />

               &nbsp;&nbsp;<label for="pi2" style="font-size: large;">Drops in the square:</
label>
               <input type="text" name="pi2" id="PointOut" class="inpbox" value="0"
readonly="true" />
             </p>
          </div>
        </div>


        <section>
          <nav>
            <ul>
              <li>
              <hr>
              <div class="wrap">
                <button id="st1" class="stbutton" onclick="myFunction()">START</button>
                <input type="hidden" name="stval" id="stval" value="0" readonly="true" />
              </div>
              <hr>

              <div class="wrap">
              <br>
              </div>

              <hr>
              <div class="wrap">
                <label for="perexec" style="font-size: large;">Drop Rate &nbsp;&nbsp;</
label>
                <br>
                <input type="text" class="inpboxexec" name="perexec" id="perexec"
value="10"  />ms
              </div>
              <hr>

              <div class="wrap">
              <br>
              </div>


              <div class="wrap">
              <br>
              </div>


              <div class="wrap">
              <br>
              </div>


              <div class="wrap">
              <br>
              </div>

              <div class="wrap">
              <br>
              </div>
```

```
            <div class="wrap">
            <br>
            </div>

            <div class="wrap">
            <br>
            </div>

            <div class="wrap">
            <br>
            </div>

            <hr>
            <div class="wrap">
              <button id="st1" class="stbutton2" onclick="refreshPage()">RESET</button>
            </div>
            <hr>

            <div class="wrap">
            <br>
            </div>

            </li>
          </ul>
        </nav>

        <article>
          <h1></h1>
             <canvas id="mycan" style="width:1100px; height:1100px; border:1px solid
#C0C0C0;" >
             </canvas>
             <table>
          <tr><td>Data values:</td></tr>
          <tr>
              <td colspan="3">
                  <textarea id="inputTextToSave" cols="179" rows="25"></textarea>
              </td>
          </tr>
          <tr>

              <td><input type="hidden" id="inputFileNameToSaveAs"
value="values.csv"></input></td>
```

```
                <td><button class="stbutton"onclick="saveTextAsFile()">Save Data Values</
button></td>
            </tr>

        </table>
          </article>
        </section>

        <footer>
          <p>powered by <a href="https://softeka.ch/" target="_blank"> Softeka </a></p>
        </footer>



          <script type="text/javascript">
           var pointSize = 1;
           var num = 1000000;
           var total = 0;
           var hit = 0;
           var max = 1100;
           var min = 1;
           var tout = 0;
           var pival = document.getElementById('Text1');
           var pointin  = document.getElementById('PointIn');
           var pointout = document.getElementById('PointOut');
           var d = new Date();
           var intervalexec = 10;


           function myFunction() {
            var x = document.getElementById("st1");
            if (x.innerHTML === "START") {
             x.innerHTML = "STOP";
             (document.getElementById("st1")).setAttribute("class", "stbutton1");
             document.getElementById("stval").value='1';
             //startProc();
            } else {
             x.innerHTML = "START";
             (document.getElementById("st1")).setAttribute("class", "stbutton");
             document.getElementById("stval").value='0';
            }
           }
```

```
 function setupCanvas(canvas) {
    // Get the device pixel ratio, falling back to 1.
    var dpr = window.devicePixelRatio || 1;
    // Get the size of the canvas in CSS pixels.
    var rect = canvas.getBoundingClientRect();
    // Give the canvas pixel dimensions of their CSS
    // size * the device pixel ratio.
    canvas.width = rect.width * dpr;
    canvas.height = rect.height * dpr;
    var ctx = canvas.getContext('2d');
    // Scale all drawing operations by the dpr, so you
    // don't have to worry about the difference.
    ctx.scale(dpr, dpr);
    return ctx;
 }

 var context = setupCanvas(document.getElementById('mycan'));

 i = num;
 var x = Math.random();
 var y = Math.random();
 (function theLoop (x, y, i) {
setTimeout(function () {
  var contparse  = document.getElementById('stval');
   intervalexec = document.getElementById('perexec').value;

if (contparse.value>0)
{
  var x =  Math.floor(Math.random() * (max - min + 1) + min);
  var y =  Math.floor(Math.random() * (max - min + 1) + min);

  var r = Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2));
  total++;

  d = new Date();

  if (r < 1100)
  {
    hit++;
    drawCoordinatesRed(context,x,y);
  }
```

```
            else
            {
              drawCoordinatesBlue(context,x,y);
            }

            if  (tout==10)
            {
               if (total < 10000)
               {
                 pival.value = roundNumber (( hit / total )* 4, 10);
               }
               else
               {
                 pival.value = roundNumber (( hit / total )* 4, 11);
               }

               if (r < 1100)
               {
                  console.log(d.toLocaleString()+' -> PI:'+pival.value+' HIT:'+hit+'
TOTAL:'+total+' X:'+x+' Y:'+y+' in quarter-circle'+ ' Exec:'+ intervalexec+'ms');
                  document.getElementById("inputTextToSave").value =
document.getElementById("inputTextToSave").value + '\n' + d.toLocaleString()+' ->
PI:'+pival.value+' HIT:'+hit+' TOTAL:'+total+' X:'+x+' Y:'+y+' in quarter-circle'+ ' Exec:'+
intervalexec+'ms';
               }
               else
               {
                  console.log(d.toLocaleString()+' -> PI:'+pival.value+' HIT:'+hit+'
TOTAL:'+total+' X:'+x+' Y:'+y + ' Exec:'+intervalexec+'ms');
                  document.getElementById("inputTextToSave").value =
document.getElementById("inputTextToSave").value + '\n' + d.toLocaleString()+' ->
PI:'+pival.value+' HIT:'+hit+' TOTAL:'+total+' X:'+x+' Y:'+y + ' Exec:'+intervalexec+'ms';
               }

               tout=0;
            }
            else
            {
               tout++;
            }
               pointin.value  = hit;
               pointout.value = total;
```

```
  }


    if (--i) {         // If i > 0, keep going
      theLoop(x,y,i);  // Call the loop again
    }



  }, intervalexec);
})(x, y, i);




   function roundNumber(num, scale) {
  if(!("" + num).includes("e")) {
    return +(Math.round(num + "e+" + scale)  + "e-" + scale);
  } else {
    var arr = ("" + num).split("e");
    var sig = ""
    if(+arr[1] + scale > 0) {
      sig = "+";
    }
    return +(Math.round(+arr[0] + "e" + sig + (+arr[1] + scale)) + "e-" + scale);
  }
}



   function drawCoordinatesRed(context,x,y){
            context.fillStyle = "#7D5A3E"; // brown color
    context.beginPath();
    context.arc(x, y, pointSize, 0, Math.PI * 2, true);
    context.fill();
   }

   function drawCoordinatesBlue(context,x,y){
            context.fillStyle = "#86BE3E"; // green color
    context.beginPath();
```

```
            context.arc(x, y, pointSize, 0, Math.PI * 2, true);
            context.fill();
           }


           // Function to download data to a file
        function download(data, filename, type) {
            var file = new Blob([data], {type: type});
            if (window.navigator.msSaveOrOpenBlob) // IE10+
                window.navigator.msSaveOrOpenBlob(file, filename);
            else { // Others
                var a = document.createElement("a"),
                        url = URL.createObjectURL(file);
                a.href = url;
                a.download = filename;
                document.body.appendChild(a);
                a.click();
                setTimeout(function() {
                    document.body.removeChild(a);
                    window.URL.revokeObjectURL(url);
                }, 0);
            }
        }



        function refreshPage(){
             if( ! confirm("This action will stop execution and ALL collected data will be
deleted !!!!   Do you really want to reset data? ") ){

                } else {
                     saveTextAsFile();
                     window.location.reload();
                }

        }

        function saveTextAsFile()
        {
            var textToSave = document.getElementById("inputTextToSave").value;
            var textToSaveAsBlob = new Blob([textToSave], {type:"text/plain"});
            var textToSaveAsURL = window.URL.createObjectURL(textToSaveAsBlob);
            var fileNameToSaveAs =
document.getElementById("inputFileNameToSaveAs").value;

            var downloadLink = document.createElement("a");
            downloadLink.download = fileNameToSaveAs;
            downloadLink.innerHTML = "Download File";
            downloadLink.href = textToSaveAsURL;
            downloadLink.onclick = destroyClickedElement;
            downloadLink.style.display = "none";
            document.body.appendChild(downloadLink);

            downloadLink.click();
        }

        function destroyClickedElement(event)
        {
            document.body.removeChild(event.target);
        }


          </script>

         </body>
        </html>
```