//Distance

function Point(x,y){

this.x = x;

this.y = y;

this.distanceTo = function (point)

{

var distance = Math.sqrt((Math.pow(point.x-this.x,2))+(Math.pow(point.y-this.y,2)))

return distance;

};

}

var newPoint = new Point (-1.58718,6.6708);

var nextPoint = new Point (-1.5877,6.6712);

console.log(newPoint.distanceTo(nextPoint))

//Bearing

function radians(n) {

return n \* (Math.PI / 180);

}

function degrees(n) {

return n \* (180 / Math.PI);

}

function Point(x,y){

this.x = x;

this.y = y;

function getBearing(Point pt1, Point pt2){

var mAngle;

var mDegree;

int mDegree\_int

//Get Angle Of Line Between Two Points

var yDiff = pt2.X - pt1.X;

var xDiff = pt2.Y - pt1.Y;

//x1 = radians(x1);

// y1 = radians(y1);

// x2 = radians(x2);

//y2 = radians(y2);

//calculate angles

if (yDiff == 0) { mAngle = 90; }

else { mAngle = Math.tan(yDiff / xDiff) \* (180 / Math.PI); }

if (yDiff > 0 && xDiff > 0) { mAngle = mAngle + 0; }

else if (yDiff > 0 && xDiff < 0) { mAngle = mAngle + 180; }

else if (yDiff < 0 && xDiff < 0) { mAngle = mAngle + 180; }

else if (yDiff < 0 && xDiff > 0) { mAngle = mAngle + 360; }

else if (yDiff == 0 && xDiff > 0) { mAngle = mAngle + 0; }

else if (yDiff == 0 && xDiff < 0) { mAngle = 180 - mAngle; }

else if (yDiff > 0 && xDiff == 0) { mAngle = mAngle + 180; }

else if (yDiff < 0 && xDiff == 0) { mAngle = 360 - mAngle; }

var deltay = point.y2 - point.y1;

var deltax = Math.log(Math.tan(point.x/2.0+Math.PI/4.0)/Math.tan(point.x/2.0+Math.PI/4.0));

if (Math.abs(deltay) > Math.PI){

if (deltay > 0.0)

deltay = -(2.0 \* Math.PI - deltay);

else

deltay = (2.0 \* Math.PI + deltay);

}

return (degrees(Math.atan2(deltay, deltax)) + 360.0) % 360.0;

}

var newPoint = new Point (-1.58718,6.6708);

var nextPoint = new Point (-1.5877,6.6712);

console.log(newPoint.distanceTo(nextPoint))