Please send any comments and/or corrections to LFS

Javascript functions for GeoGebra - ggb applet part is in NAVY

In and Out Functions

setScalar // assigns the ggb variable "objName" the value objValue
Existing // tests that the ggb object "objName" exists
CheckAnswer // checks numeric value "objName1"=numeric value "objName2" within 2 decimals
GridPoint // tests that the point "objName" is a grid point (has whole number coords)
SamePoint // tests that the points "objName1" & "objName2" are not the same point
GoodPoints // tests that points "objName1" & "objName2" are not on same vertical or horizontal
PointLine // tests that the point "objName" is on the line with slope=za and y-intercept=zb
FindSlopeInt // finds slope and y-intercept of the line through points "objName1" & "objName2"

Some Math Functions

gensetNN2 // function generates 2 unequal random Natural Numbers and assigns them to variables "objName1" & "objName2" in ggb
gensetAngle // generates random integer between 15 and 75, changes it to radians and assigns it to variable "objName" in ggb
gensetSgn // generates random +1 or -1 and assigns to ggb variable "objName" in ggb

my Functions

myRound // function rounds variable x to n decimal places
myFloor // function truncates the number after adding a tiny epsilon to avoid computer rounding errors.

	In and Out Functions	Comments

1.	<script type="text/javascript">
	function setScalar(objName, objValue) {
	// assigns the ggb variable objName the value objValue
	// objValue must be a number, objName will be a numeric variable
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	// e.g. for input objName='num' and objValue=3 function builds string
	// 'num = 3' and lets applet "evaluate this input" in ggb
	applet.evalCommand(objName + " = " + x);	

	} </script>	

2.	<script type="text/javascript">
	function Existing(objName) {
	  // tests that the ggb object objName exists and assigns the ggb variable
	  //okE+objName a boolean value of 1 if exists and 0 if doesn't exist.
	  // substitutes "applet" for "document.ggbApplet" to make function shorter
	  var applet = document.ggbApplet;
	  applet.evalCommand("okE"+objName+"=1");
	  if (!applet.exists(objName)) {
	  applet.evalCommand("okE"+objName+"=0");
	  // if it doesn't exists sends an alert to the user.
	  alert('Point '+objName+' does not exist.\nPlease select it.');}
	}
	</script>	

3.	<script type="text/javascript">
	function CheckAnswer(objName1,objName2) {
	  // checks whether numeric value objName1=numeric value objName2 within 2 decimals
	  //use value as desired to avoid computer rounding problems
	  // substitutes "applet" for "document.ggbApplet" to make function shorter
	  var applet = document.ggbApplet;
	  var x = myRound(applet.getValue(objName1),2);
	  var y = myRound(applet.getValue(objName2),2);
	  // sends proper alert
	if (x == y) {
	  alert('Congratulations! This is correct!');
	  } else if (x != y){
	  alert('Sorry! This is not correct!');}
	}
	</script>
		uses:
		myRound

4.	<script type="text/javascript"> function GridPoint(objName) {	

	// tests that the point "objName" is a grid point (has whole number coords) and
	// assigns the ggb variable okGB+objName a boolean value of 1 if so and 0 if not
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	applet.evalCommand("okGP"+objName+"=1");
	var zx = applet.getXcoord(objName)-Math.round(applet.getXcoord(objName));
	var zy = applet.getYcoord(objName)-Math.round(applet.getYcoord(objName));
	// if not, it deletes the object and sends an alert to the user.
	if ((zx != 0) || (zy != 0)) {
	applet.deleteObject(objName);
	applet.evalCommand("okGP"+objName+"=0");
	alert('Point '+objName+' is not a grid point.\nClick "Reset" to start over.');}
	}
	</script>	

5.	<script type="text/javascript">
	function SamePoint(objName1, objName2) {
	// tests that the point "objName1" and "objName2" are not the same point and
	// assigns the ggb variable okSP a boolean value of 1 if not and 0 if same.
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	applet.evalCommand("okSP=1");
	var zx = applet.getXcoord(objName1)-applet.getXcoord(objName2);
	var zy = applet.getYcoord(objName1)-applet.getYcoord(objName2);
	// if same point, deletes both objects in ggb and alerts the user
	if ((zx == 0) && (zy == 0)) {
	applet.deleteObject(objName1);
	applet.deleteObject(objName2);
	applet.evalCommand("okSP=0");
	alert('You must select different points A and B.');}
	}
	</script>
		I use this function after checking first that the points are grid points – else you may need to round zx and zy to avoid computer rounding errors.
		
		SamePoint has the stricter "and"=&& condition.
		
		GoodPoints has the weaker "or"=|| condition.
		
		i.e. fail SamePoint implies fail GoodPoint, but not vice-versa

6.	<script type="text/javascript"> function GoodPoints(objName1, objName2) {	I use this function after checking first that the points are grid points– else you

	// tests that the points "objName1" and "objName2" are not on the same vertical
	// or horizontal line and assigns ggb object okGP a boolean value of 1 if not and
	// 0 if so.
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	applet.evalCommand("okGP=1");
	// finds delta x and delta y
	var zx = applet.getXcoord(objName1)- applet.getXcoord(objName2);
	var zy = applet.getYcoord(objName1)- applet.getYcoord(objName2);
	// if same vertical or horizontal, deletes the objName2 in ggb and alerts the user
	if ((zx == 0) || (zy == 0)) {
	applet.deleteObject(objName2);
	applet.evalCommand("okGP=0");
	alert('You must not have the same x or y coordinates for A and B. \nClick "Reset" to start over.');}
	}
	</script>
		may need to round zx and zy to avoid computer rounding errors.
		
		SamePoint has the stricter "and"=&& condition.
		
		GoodPoints has the weaker "or"=|| condition.
		
		i.e. fail SamePoint implies fail GoodPoint, but not vice-versa

7.	<script type="text/javascript">
	function PointLine(objName) {
	// tests that the point "objName" is on the line with slope=za and y-intercept=zb
	// where za and zb are existing ggb numbers and assigns the ggb variable
	// okPL+objName a boolean value of 1 if exists and 0 if doesn't exist.
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	applet.evalCommand("okPL"+objName+"=1");
	// substitutes x and y coordinates of objName into line
	// important rounding problem with ggb
	// if using points with decimal values you may need to multiply by say
	// 1000, round and divide by 1000 to get z to be really zero.
	var z = applet.getValue('za')*applet.getXcoord(objName)+ applet.getValue('zb')-applet.getYcoord(objName);
	// if objName is not a point, deletes the object in ggb and alerts the user
	if (z != 0) {
	applet.deleteObject(objName);	

	applet.evalCommand("okPL"+objName+"=0");
	alert('Point '+objName+' is not on the line.\nPlease reselect it.');}
	}
	</script>	

8.	<script type="text/javascript">
	function FindSlopeInt(objName1, objName2) {
	// finds the slope and y-intercept of the line passing through points "objName1" and
	// "objName2" and assigns the ggb variable za=slope, and zb=y-intercept
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	// same code here as for GoodPoint function
	// finds delta x and delta y
	var zx = applet.getXcoord(objName1)-applet.getXcoord(objName2);
	var zy = applet.getYcoord(objName1)-applet.getYcoord(objName2);
	// if same vertical or horizontal, deletes the objName2 in ggb and alerts the user
	if ((zx == 0) || (zy == 0)) {
	applet.deleteObject(objName2);
	applet.evalCommand("okGP=0");
	alert('You must not have the same x or y coordinates for A and B. \nClick "Reset" to start over.');}
	// if points are okay calculates and assigns ggb variables za and zb
	if (applet.getValue('okGP')!= 0) {
	applet.evalCommand("za="+zy/zx);
	applet.evalCommand("zb="+zy/zx*(-1)*applet.getXcoord(objName1)+applet.getYcoord(objName1));}
	}
	</script>	

1.	<script type="text/javascript">
	function gensetNN2(objName1,objName2) {
	// function generates 2 unequal random Natural Numbers between 1 and 10
	// assigns them to ggb variables objName1 and objName2
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;	

	// nn1, nn2 - generates random integer between 1 and 10
	// use Math.floor to get 0-9=10-1 and/or change 10 to get different range!
	var nn1 = Math.ceil(Math.random()*10);
	var nn2 = Math.ceil(Math.random()*10);
	// if equal changes second number, 5 is 10/2
	var sgn=1
	if(nn1==nn2) {
	if(nn1>5) {sgn=-1;}
	nn2=nn2+sgn*Math.ceil(Math.random()*5);}
	// ggb assignment
	applet.evalCommand(objName1 + " = " + nn1);
	applet.evalCommand(objName2 + " = " + nn2);
	}
	</script>	

2.	<script type="text/javascript">
	function gensetAngle(objName) {
	// generates random integer between 15 and 75
	var x = Math.ceil(Math.random()*60)+15;
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	// sends radian value of x to GeoGebra (where - if set to degrees –
	// it will automatically be converted back into degrees)
	var ang = x*3.141592653/180;
	applet.evalCommand(objName + " = " + ang);
	}
	</script>	

3.	<script type="text/javascript">
	function gensetSgn(objName) {
	// generates random plus and minus sign +1 or -1
	// assigns it to ggb variable objName
	// substitutes "applet" for "document.ggbApplet" to make function shorter
	var applet = document.ggbApplet;
	// nn0 generates random integer 0 or 1	

```
  var nn0 = Math.floor(Math.random()*2)
  // sgn=2*{0,1}-1 yields +1 or -1
  var sgn = 2*nn0 -1;
  // ggb assignment
  applet.evalCommand(objName + " = " + sgn );
}
</script>
```

```
<script type="text/javascript">
function myRound(x,n) {
// rounds variable x to n decimal places
var dec = Math.pow(10,n)
return Math.round(dec*x)/dec;
}
</script>
```

```
<script type="text/javascript">
function myFloor(x) {
// adds tiny epsilon before flooring to avoid computer rounding error
return Math.floor(x+Math.pow(10,-5));
}
</script>
```