

Utility Functions: Geometry Functions

Posted At : March 24, 2010 10:16 PM | Posted By : Jon Hartmann

Related Categories: ColdFusion, Utility Function

Today we've got a double header of functions dealing with Geometry. As I said in my last post, I've been spurred into action by a user's question, and in order to solve his problem, I'm going to need to be able to calculate some relatively simple geometry. Click "more" to see how to convert Degrees to Radians and how to calculate the position of a point based on a starting location, angle, and distance.

GeometryDegreesToRadians

This one's the boring one:

```
<cffunction name="GeometryDegreesToRadians" output="false" returnType="numeric">
    <cfargument name="Degrees" type="numeric" required="true" />

    <!-- Convert the Degrees to radians -->
    <cfreturn (arguments.Degrees * Pi()) / 180 />
</cffunction>
```

Normally I'd not even bother to encapsulate something like this, but I'm not sure how often I'm going to need it with this task, and I've already had to use it again in the next function.

GeometryPointOffset

In order to solve the problem I'm working on I need to be able to take a point, an angle, and a distance and figure out a new point that is that distance from the original point along that angle. Its not too complex a task, but it takes some "simple" geometry that I'd not seen since I was in 8th grade. Go Math!

```
<cffunction name="GeometryPointOffset" output="true" returnType="struct">
    <cfargument name="Angle" type="numeric" required="true" />
    <cfargument name="Distance" type="numeric" required="true" />
    <cfargument name="StartX" type="numeric" required="false" default="0" />
    <cfargument name="StartY" type="numeric" required="false" default="0" />

    <cfset var local = StructNew() />

    <!-- Declare return value -->
    <cfset local.offset = StructNew() />

    <!-- Find Angle A -->
    <cfset local.angleA = arguments.Angle />

    <!-- Find Angle B -->
    <cfset local.angleB = 90 - local.angleA />

    <!-- Find X Offset -->
    <cfset local.offset.y = Round(Sin(GeometryDegreesToRadians(local.angleA)) * Distance) />

    <!-- Find Y Offset -->
    <cfset local.offset.x = Round(Sin(GeometryDegreesToRadians(local.angleB)) * Distance) />

    <!-- Return Offset -->
    <cfreturn local.offset />
</cffunction>
```

One thing that you'll notice is that you don't have to supply an initial (x,y) coordinate for this function; all of my uses for this function are starting at (0,0). The results of testing look like this:

Angle Distance End Pointe

0° 5 (5,0)

45°	10	(7,7)
90°	5	(0,5)
135°	10	(-7,7)
180°	5	(-5,0)
270°	10	(0,-10)
360°	5	(5,0)
-360°	10	(10,0)
-270°	5	(0,5)
-180°	10	(-10,0)
-135°	5	(-4,-4)
-90°	10	(0,-10)
-45°	5	(4,-4)