

# rounds val to the nearest fractional value

```
//http://isotopescreenshot.codeplex.com/
//The MIT License (MIT)
namespace Isotope.Math
{
public static class MathUtil
{
public static double Round(double val, double snap_val)
{
return Round(val, System.MidpointRounding.AwayFromZero,
snap_val);
}
}
}

/// rounds val to the nearest fractional value
///
/// the value to round /// what kind of rounding /// round to
this value (must be greater than 0.0) /// the rounded value
public static double Round(double val, System.MidpointRounding
rounding, double frac)
{
/*
if (frac <= 0) { throw new
ArgumentOutOfRangeException("frac","must be greater than or
equal to 0.0"); }*/ double retval =
System.Math.Round((val/frac), rounding)*frac; return retval; }
public static double RoundUp(double v, double amount) { const
System.MidpointRounding rounding =
System.MidpointRounding.ToEven; var result = Round(v +
(amount/2.0), rounding, amount); return result; } } }
[/csharp]
```