After having posted about the basics of distance functions in several places (pouet, my blog, shadertoy, private emails, etc), I thought it might make sense to put these together in centralized place. Here you will find the distance functions for basic primitives, plus the formulas for combining them together for building more complex shapes, as well as some distortion functions that you can use to shape your objects. Hopefully this will be usefull for those rendering scenes with raymarching. You can see some of the results you can get by using these techniques in the raymarching distance fields article. Lastly, this article doesn't include lighting tricks, nor marching acceleartion tricks or more advanced techniques as recursive primitives or fractals.

primitives

All primitives are centered at the origin. You will have to transform the point to get arbitrarily rotated, translated and scaled objects (see below).

Sphere - signed - exact

```
float sdSphere( vec3 p, float s )
{
  return length(p)-s;
}
```

Box - unsigned - exact

```
float udBox( vec3 p, vec3 b )
{
  return length(max(abs(p)-b,0.0));
}
```

Round Box - unsigned - exact

```
float udRoundBox( vec3 p, vec3 b, float r )
{
  return length(max(abs(p)-b,0.0))-r;
}
```