

turtleSpaces 3D Shape Guide

Note: the pipe symbol | denotes the separation of parameters but is not used when calling the primitives.

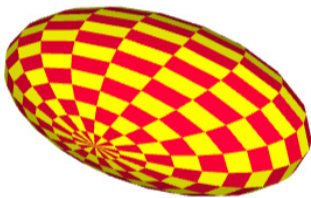


amigaball

radius (number) | stacks (number) | sectors (number)

Creates an 'Amiga ball', a sphere that alternates sectors between the pen color and the fill color.

`amigaball 40 20 20`



amigaballoid

radius (number) | stacks (number) | sectors (number) | ratio (number)

Creates a stretched or flattened Amiga ball, based on the supplied ratio. Respects setspheroidaxis

`amigaballoid 30 20 20 2`

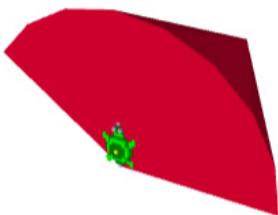


cone

radius (number) | depth (number) | sides (number)

Creates a cone pointing downward from and centered on the calling turtle.

`cone 30 100 20`

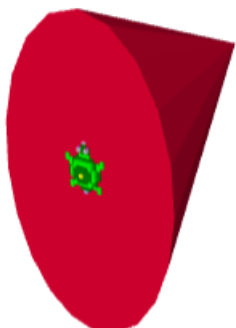


coneslice

radius (number) | depth (number) | sidestotal (number) | sidesused (number)

Creates a 'sliced' or partial cone, based on its sides.

`coneslice 50 100 20 8`



conoid

radius (number) | depth (number) | sides (number) | ratio (number)

Creates a conoid, a cone stretched or compressed based on the provided ratio (1 is normal, 2 is stretched twice as much, 0.5 half as much, etc.)

`conoid 50 100 20 0.5`

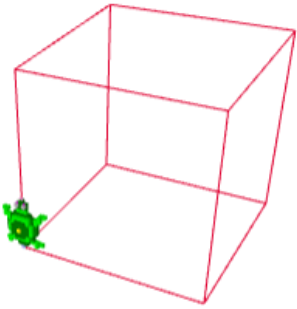


conoidslice

radius | depth | sidestotal | sidesused | ratio

Creates a sliced conoid.

`conoidslice 50 100 20 8 0.5`

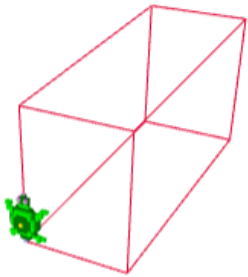


cube

size

Creates a wireframe cube of the specified size.

`cube 50`



cuboid

width | height | depth

Creates a wireframe cuboid of the specified dimensions.

`cuboid 30 40 80`



cutcone

radiusnear (number) | radiusfar (number) | depth (number) | sides (number)

Creates a truncated, or cut, cone. See cone

`cutcone 20 50 60 20`



cutconeslice

radiusnear | radiusfar | depth | sidestotal | sidesused

Creates a cut, sliced cone. See coneslice

`cutconeslice 20 50 60 20 5`

Note: Views of the interior of 'open' shapes are shown as they are when the **twosided** primitive is called before they are used. They will otherwise be non-reflective (matte) inside.

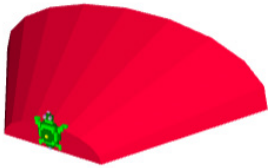


cutconoid

`radiusnear` | `radiusfar` | `depth` | `sides` | `ratio`

Creates a truncated, or cut, conoid, stretched based on the provided ratio. See conoid

`cutconoid 20 50 60 20 0.5`

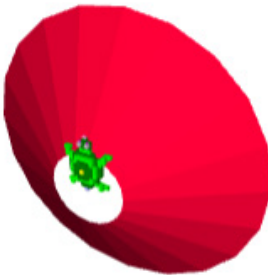


cutconoidslice

`radiusnear` | `radiusfar` | `depth` | `sides` | `ratio`

Creates a cut, sliced conoid, stretched based on the provided ratio. See conoid

`cutconoidslice 10 30 60 20 10 2`



cutfunnel

`radiusnear (number)` | `radiusfar (number)` | `depth (number)` | `sides (number)`

Creates a cut funnel (an uncapped cut cone).

`cutfunnel 10 40 20 20`



cutfunneloid

`radiusnear` | `radiusfar` | `depth` | `sides` | `ratio`

Creates a cut funneloid (an uncapped cut conoid).

`cutfunneloid 20 40 40 20 0.5`



cutfunneloidslice

`radiusnear` | `radiusfar` | `depth` | `sides` | `ratio`

Creates a cut, sliced funneloid (an uncapped cut conoid).

`cutfunneloidslice 20 40 40 20 10 0.5`



cutfunnelslice

[radiusnear](#) | [radiusfar](#) | [depth](#) | [sidesused](#) | [sidesused](#)

Creates a sliced, cut funnel (an uncapped cut cone).

`cutfunnelslice 20 40 40 20 10`

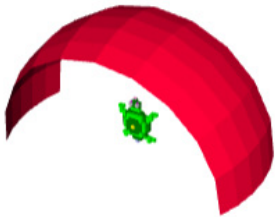


cutsphere

[radius](#) | [stacks](#) | [sectors](#) | [minstack](#) | [maxstack](#)

Creates a truncated, or cut, sphere. Can be cut on the top, bottom, or both based on specified minstack and maxstack values. See sphere

`cutsphere 40 20 20 8 12`

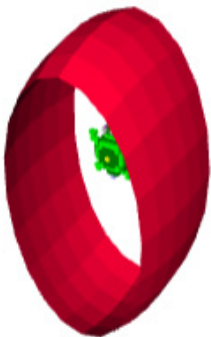


cutsphereslice

[radius](#) | [stacks](#) | [totalsectors](#) | [minstack](#) | [maxstack](#) | [usedsectors](#)

Creates a sliced and cut sphere. Can be cut on the top, bottom, or both based on specified minstack and maxstack values. Displays the portion specified by totalsectors and usedsectors. See sphereslice, cutsphere

`cutsphereslice 40 20 20 8 12 10`



cutspheroid

[radius](#) | [stacks](#) | [sectors](#) | [minstack](#) | [maxstack](#) | [ratio](#)

Creates a truncated, or cut, sphere, stretched by the provided ratio. Can be cut on the top, bottom, or both based on specified minstack and maxstack values. See spheroid

`cutspheroid 40 20 20 8 12 0.5`



cutspheroidslice

[radius](#) | [stacks](#) | [totalsectors](#) | [minstack](#) | [maxstack](#) | [usedsectors](#) | [ratio](#)

Creates a sliced and cut spheroid. Can be cut on the top, bottom, or both based on specified minstack and maxstack values. Displays the portion specified by totalsectors and usedsectors. Is stretched by the ratio provided. See spheroidslice, cutspheroid

`cutspheroidslice 40 20 20 8 12 12 0.5`



cylinder

radius (number) | depth (number) | sides (number)

Creates a cylinder below the turtle.

`cylinder 20 40 20`

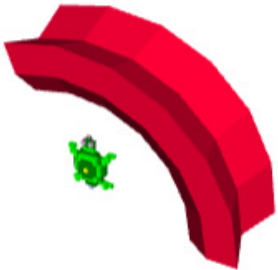


cylinderarc

thickness | radius | sides | totalsegments | arcsegments

Creates a partial torus, or curved cylinder, around the turtle, starting on the right.

`cylinderarc 8 30 10 20 6`



cylinderarcslice

thickness | radius | sides | totalsegments | arcsegments | startsides | endsides

Creates a slice of a partial torus, or curved cylinder. Note: If you invert the thickness value, you can 'slice' the opposite side.

`cylinderarcslice 12 40 10 20 6 3 8`



cylinderslice

radius (number) | depth (number) | totalsides (number) | slicesides (number)

Creates a rounded wedge shape, like a slice of a round of cheese, starting at the turtle's right vector and moving counter-clockwise.

`cylinderslice 30 20 20 15`



cylindroid

radius (number) | depth (number) | sides (number) | ratio (number)

Creates a cylindroid, a distorted cylinder based on the ratio.

`cylindroid 30 30 10 0.5`



cylindroidarc

thickness | radius | sides | totalsegments | arcsegments | ratio

Creates a cylinder arc extruded by the provided ratio.

`cylindroidarc 10 30 10 20 7 2`



cylindroidarcslice

thickness | radius | totalsides | totalsegments | arcsegments | startsides | endsides | ratio

Creates a cylinder arc sliced by the provided sides values and extruded by the provided ratio.

`cylindroidarcslice 20 40 10 20 7 6 8 2`

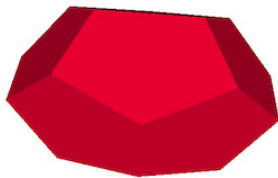


cylindroidslice

radius | depth | totalsides | slicesides | ratio

Creates a stretched rounded wedge shape, like a slice of a round of cheese, starting at the turtle's right vector and moving counter-clockwise, and stretched using the provided ratio, where 1 is normal, 0.5 is half, 2 is double etc.

`cylindroidslice 40 50 10 4 0.5`



dodecahedroid

size (number) | ratio (number)

Creates a dodecahedroid (12-sided shape) of the specified size and stretched using the specified ratio.

`dodecahedroid 30 1.5`

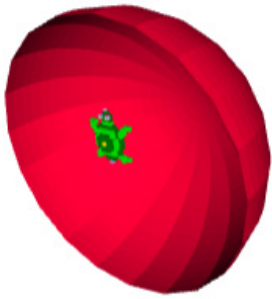


dodecahedron

size

Creates a dodecahedron (12-sided shape) of the specified size.

`dodecahedron 30`



dome

radius (number) | stacks (number) | sectors (number)

Creates a dome below the turtle.

```
dome 40 20 20
```



domoid

radius (number) | stacks (number) | sectors (number) | stretch (ratio)

Creates a stretched dome below the turtle.

```
domoid 40 20 20 0.5
```

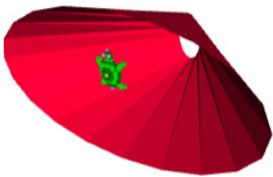


duocutconoid

radiusnear | radiusfar | depth | sides | ratio1 | ratio2

Creates a cut conoid with two different oid (stretch) ratios, one for each end. See cutconoid, duocutfunneloid

```
duocutconoid 10 40 50 20 2 0.5
```

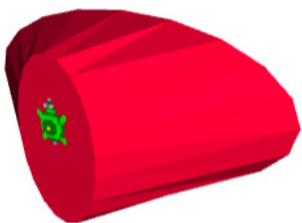


duocutfunneloid

radiusnear | radiusfar | depth | sides | ratio1 | ratio2

Creates a cut funneloid (an uncapped cut conoid) with two different oid (stretch) ratios, one for each end. See cutfunneloid

```
duocutfunneloid 30 10 50 20 2 0.5
```

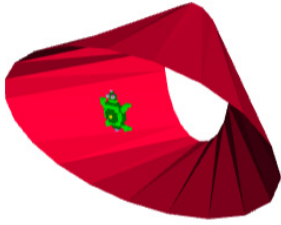


duocylindroid

radius | depth | sides | ratio1 | ratio2

Creates a cylindroid with two different oid (stretch) ratios, one for each end. See cylindroid, duotuboid

```
duocylindroid 30 40 20 0.5 2
```

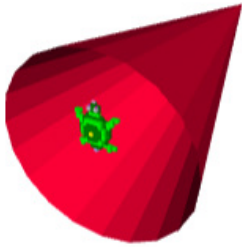


duotuboid

radius | depth | sides | ratio1 | ratio2

Creates a tuboid, or stretched tube, with two different oid (stretch) values, one for each end. Can be used to transition between non-oids and oids, or between two oids. See tuboid

`duotuboid 30 50 20 2 0.5`



funnel

radius (number) | depth (number) | sides (number)

Creates a funnel (an uncapped cone). Note that this is the 'twosided' view, the default view does not reflect light on the inside of the funnel. To make it reflect, first call the twosided primitive.

`funnel 30 100 20`



funneloid

radius (number) | depth (number) | sides (number) | ratio (number)

Creates a funneloid (an uncapped conoid).

`funneloid 40 30 20 0.5`

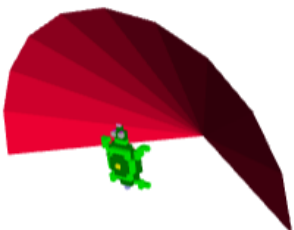


funneloidslice

radius | depth | sidestotal | sidesused | ratio

Creates a sliced funneloid.

`funneloidslice 40 30 20 13 0.5`



funnelslice

radius (number) | depth (number) | sidestotal (number) | sidesused (number)

Creates a sliced funnel.

`funnelslice 40 30 20 10`

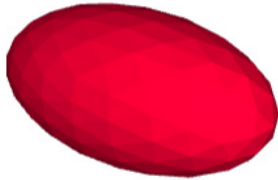


icosphere (ico)

size

Creates an icosphere, an approximation of a sphere as a simplicial polyhedron, formed by subdividing the triangles of a regular icosahedron into smaller triangles.

icosphere 30



icospheroid

size

Creates an icospheroid of a specified radius, stretched using the provided multiplier.

icospheroid 20 2



inscribe

text (word), number or list

Renders text shapes alongside and to the front of the turtle rather than to the side (as with typeset) using the same size and color as typeset. Unlike typeset, the turtle does not move. See typeset.

inscribe "turtle"



octahedroid

size | ratio

Creates a octahedroid (8-sided shape) of the specified size and stretched using the specified ratio.

octahedroid 80 0.5

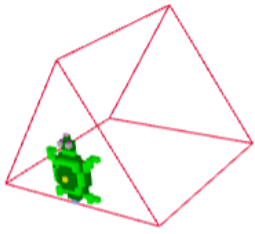


octahedron

size

Creates a octahedron (8-sided shape) of the specified size. Note that the provided size value indicates the length of the triangles that make up the shape.

octahedron 60

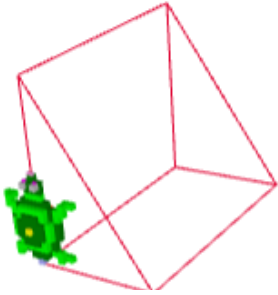


peak

size

Creates a lined wedge around and beneath the calling turtle. The width distance is the distance to one side of the turtle (the total width of the peak is double the provided width value).

peak 20 30 40



prism

width | height | depth

Creates a lined prism to the right, forward and under the turtle.

prism 20 30 40



pyramid

size

Creates a pyramid of the given size.

pyramid 40



pyramid [list]

[width height depth]

Creates a pyramid using the list of supplied dimensions

pyramid [80 50 20]



pyramidoid

size | ratio

Creates a pyramid (stretched pyramid) of the specified size and stretch ratio.

pyramidoid 30 2



pyramidoid [list]

[width height depth] | ratio

Creates a pyramid (stretched pyramid) of the specified dimensions and ratio.

pyramidoid [80 50 20] 0.5

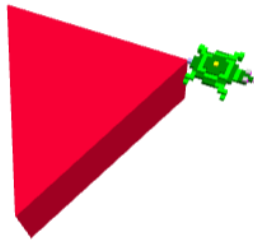


ramp

width (number) | height (number) | depth (number)

Creates a filled wedge, or 'ramp' using the provided dimensions.

```
ramp 20 30 40
```



shard

depth

Creates a 'shard' or a three-dimensional fragment using the current and two previous turtle positions with a depth specified by its input parameter.

```
fd 60 rt 120 fd 50 shard 20
```



skewpyramid

size (turtle-units) OR [width height depth] (list) | degrees X | degrees Z

Creates a pyramid "skewed" by the specified number of degrees in the X and Z directions.

```
skewpyramid 50 -20 20
```



skewpyramidoid

size (turtle-units) | stretch (ratio) | degrees X | degrees Z

Creates a pyramid "skewed" by the specified number of degrees in the X and Z directions, and stretched by the specified ratio

```
skewpyramidoid 50 2 -20 20
```



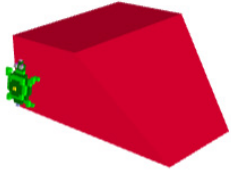
skewpyramidoid [list]

[width height depth] | stretch (ratio) | degrees X | degrees Z

Creates a pyramid "skewed" by the specified number of degrees in the X and Z directions, and stretched by the specified ratio

```
skewpyramidoid [40 20 50] 2 -20 20
```

skewtrapevoxeloid



size

Creates a skewed trapezoidal voxel, the width and depth of whose “top” or side in front of the turtle is stretched or contracted by the given ratios (depth and width), and who is then skewed by the given degree values in the X and Z axis, which can be negative. See trapevoxeloid

```
skewtrapevoxeloid 50 40 30 0.5 2 -45 45
```

skewvoxeloid



width (turtle-units) | height | depth | degrees X | degrees Z

Creates a “skewed” voxeloid of the specified dimensions, distorted on the X and Z axis by the specified number of degrees, which can be negative.

```
skewvoxeloid 20 40 30 30 20
```

sphere



radius (number) | stacks (number) | sectors (number)

Creates a sphere of the specified radius using the specified number of stacks and the specified number of sectors. Small stack and sector values create interesting shapes. As opposed to icosphere

```
sphere 30 20 20
```

sphereslice



radius (number) | stacks (number) | sectorstotal (number) | sectorsused (number)

Creates a partial sphere of the overall specified radius, using the specified number of vertical ‘stacks’, and specified number of horizontal ‘sectors’, based on the given total number of sectors, and the used (displayed) number of sectors. Like an orange cut vertically. See cutsphere, cutsphereslice

```
sphereslice 40 20 20 3
```

spheroid



radius (number) | stacks (number) | sectors (number) | multiplier (number)

Creates an spheroid of a specified radius, stacks and sectors, stretched using the provided multiplier.

```
spheroid 30 20 20 2
```

spheroidslice

radius | stacks | sectorstotal | sectorsused | multiplier



Creates a partial spheroid of the overall specified radius, using the specified number of vertical 'stacks', and specified number of horizontal 'sectors', based on the given total number of sectors, and the used (displayed) number of sectors, and stretched using the provided multiplier. Like a squashed orange cut vertically. See cutspheroid, cutspheroidslice

spheroidslice 30 20 20 3 2

tent

half-width (number) | height (number) | depth (number)



Creates an opaque (solid) prism to the left, right and front of the turtle.

tent 20 50 30

tetrahedroid

size (number) | ratio (number)



Creates a tetrahedroid (4-sided shape) of the specified size and stretched using the specified ratio.

tetrahedroid 40 2

tetrahedron

size (number)

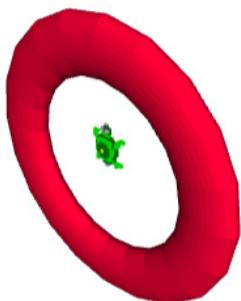


Creates a tetrahedron (4-sided shape) of the specified size.

tetrahedron 60

toroid

thickness | radius | sides | segments | ratio



Creates a torusoid, an extruded three-dimensional ring, stretched by a ratio where 1 is a normal torus. See torus

toroid 10 40 30 20 0.5



toroidslice

thickness | radius | sides | totalsegments | startsides | endsides | ratio

Creates a 'slice' of a torus, based on the startsides and endsides value. Note that using a negative thickness value inverts the slice.

```
toroidslice 10 40 30 20 10 15 2
```



torus

thickness (number) | radius (number) | sides (number) | segments (number)

Creates a torus, a three-dimensional ring. Sides and segments can be as low as 3.

```
torus 10 40 20 20
```

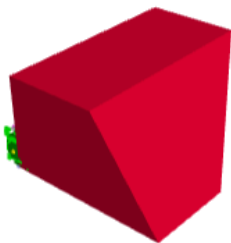


torusslice

thickness | radius | sides | totalsegments | startsides | endsides

Creates a 'slice' of a torus, based on the startsides and endsides value. Note that using a negative thickness value inverts the slice.

```
torusslice 10 40 30 20 10 2.5
```

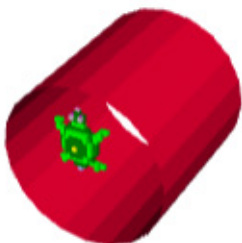


trapevoxeloid

width (turtle-units) | height | depth | ratio-W | ratio-D

Creates a trapezoidal voxel, whose far 'end' is affected by the given width and depth ratios.

```
trapevoxeloid 50 40 30 0.5 2
```



tube

radius (number) | depth (number) | sides (number)

Creates a hollow tube under the turtle.

```
tube 20 40 20
```



tubearc

thickness | radius | sides | totalsegments | arcsegments

Creates a tube arc, a section of a torus.

```
tubearc 10 30 20 20 4
```



tubearcslice

thickness | radius | sides | totalsegments | arcsegments | startsides | endsides

Creates a slice of a partial tube. Note: If you invert the thickness value, you can 'slice' the opposite side.

```
tubearcslice 10 30 20 20 7 10 20
```



tubeslice

radius (number) | depth (number) | sidestotal (number) | sidesused (number)

Creates a 'slice' of a tube, based on the sidestotal and sidesused values.

```
tubeslice 30 20 20 10
```



tuboid

radius (number) | depth (number) | sides (number) | ratio (number)

Creates a tuboid, or stretched tube.

```
tuboid 30 20 20 0.5
```



tuboidarc

thickness | radius | sides | totalsegments | arcsegments | ratio

Creates a tube arcoid (extruded based on the provided ratio).

```
tuboidarc 10 30 20 20 4 2
```



tuboidarcslice

[thickness](#) | [radius](#) | [totalsides](#) | [totalsegments](#) | [arcsegments](#) | [startsides](#) | [endsides](#) | [ratio](#)

Creates a tube arc sliced by the provided sides values and extruded by the provided ratio. (Whew!)

```
tuboidarcslice 10 30 20 20 7 10 20 2
```



tuboidslice

[radius](#) | [depth](#) | [sidestotal](#) | [sidesused](#) | [ratio](#)

Creates a stretched 'slice' of a tube, based on the sidestotal and sidesused values, and the stretching ratio (normal is 1, half is 0.5, double is 2 etc.)

```
tuboidslice 40 20 20 10 0.5
```

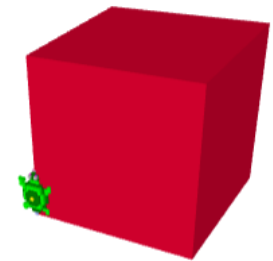


typeset

[word](#) or [\[list\]](#)

Typeset renders the provided word or list as three-dimensional text to the right of the turtle one character at a time, moving the turtle to the right as it does so. You can use `settypesize`, `settypefont`, `settypedepth` to change different parameters.

```
typeset "Turtles!"
```

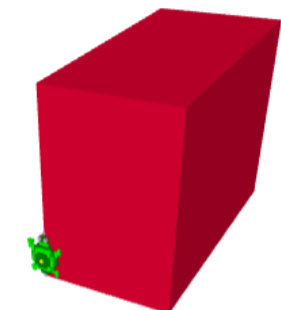


voxel

[size](#)

Creates a voxel (a filled cube) of the specified size. You can use the `speckle` primitives to create pixelated patterns on voxels, see the website for details or type `help "speckle` inside the (web) application.

```
voxel 50
```



voxeloid

[turtle-steps-right](#) [turtle-steps-forward](#) [turtle-steps-deep](#)

Creates a filled cuboid to the right, forward and down of the turtle.

```
voxeloid 40 60 80
```