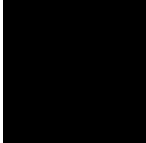





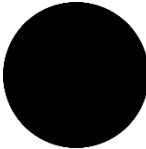





Runes Library

- Use `show(rune)` to display the rune.

Rune	Reference Image
<code>black_bb</code>	
<code>blank_bb</code>	
<code>rcross_bb</code>	
<code>sail_bb</code>	
<code>corner_bb</code>	
<code>nova_bb</code>	
<code>circle_bb</code>	
<code>heart_bb</code>	
<code>pentagram_bb</code>	
<code>ribbon_bb</code>	

Transformation Functions

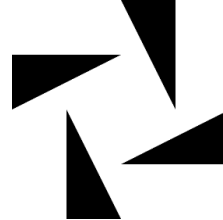
- `scale(ratio, shape)`: Scales the shape to the ratio
- `scale_independent(ratio_x, ratio_y, shape)`: Scales the shape by `ratio_x` along the x-axis and `ratio_y` along the y-axis
- `translate(x, y, shape)`: Shifts the shape by `x` amount along the x-axis and `y` amount along the y-axis
- `rotate(rad, shape)`: Rotates the shape by `rad` amount in radians in the *counter-clockwise direction*
- `quarter_turn_right(shape)`: Rotates the shape clockwise by $\frac{\pi}{2}$
- `quarter_turn_left(shape)`: Rotates the shape counter-clockwise by $\frac{\pi}{2}$
- `turn_upside_down(shape)`: Rotates the shape by π
- `flip_vert(shape)`: Flips the shape vertically along the x-axis
- `flip_horiz(shape)`: Flips the shape horizontally along the y-axis
- `stack(shape1, shape2)`: Stacks shape1 onto shape2 with equal proportions (i.e. $\frac{1}{2}$)
- `stack_frac(frac, shape1, shape2)`: Stacks shape1 onto shape2 such that shape1 takes up `frac` amount of space in the rune
- `stackn(n, shape)`: Stacks `n` times of shape with equal proportions (i.e. $\frac{1}{n}$)
- `beside(shape1, shape2)`: Puts shape1 to the left of shape2 with equal proportions
- `beside_frac(frac, shape1, shape2)`: Puts shape1 to the left of shape2 such that shape1 takes up `frac` amount of space in the rune
- `make_cross(shape)`:

Example

`sail_bb`



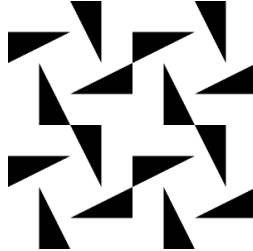
`make_cross(sail_bb)`



- `repeat_pattern(n, pattern, shape)`: Repeats the transformation function `pattern` applied to the shape `n` times

Example

`repeat_pattern(2, make_cross, sail_bb)`



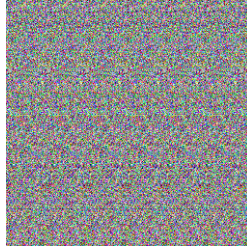
- `overlay(shape1, shape2)`: Overlays shape1 on top of shape2 with equal proportions in the z-axis [3D rune]
- `overlay_frac(frac, shape1, shape2)`: Overlays shape1 on top of shape2 such that shape1 takes up `frac` amount of space in the rune in the z-axis [3D rune]

3D Display Functions

- `stereogram(shape):`

Example

```
stereogram(heart_bb)
```



- `anaglyph(shape):`

Example

```
anaglyph(heart_bb)
```



- `hollusion(shape)`

Color Functions

- `color(shape, r, g, b):` Colors the `shape` with RGB color. Values range from 0 to 255.
- `color(shape, r, g, b, a):` Same as above, but with alpha value to specify opaqueness. Values of `a` range from 0 (transparent) to 1 (opaque).
- **`color`**(`shape`): Colors the `shape` with the standard color function
- `random_color(shape):` Colors the `shape` with a random color

Color	r	g	b	a	Standard color function
red	244	67	54	1	<code>red(shape)</code>
pink	233	30	99	1	<code>pink(shape)</code>
purple	170	0	255	1	<code>purple(shape)</code>
indigo	63	81	181	1	<code>indigo(shape)</code>
blue	33	150	243	1	<code>blue(shape)</code>
green	76	175	80	1	<code>green(shape)</code>
yellow	255	235	59	1	<code>yellow(shape)</code>
orange	255	152	0	1	<code>orange(shape)</code>
brown	121	85	72	1	<code>brown(shape)</code>
black	0	0	0	1	<code>black(shape)</code>
white	255	255	255	1	<code>white(shape)</code>