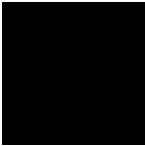




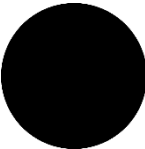





Runes Library

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

Rune (Shape)	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

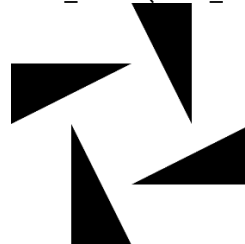
- `scale(ratio, shape)`: Scales the shape to the ratio
- `scale_independent(ratio_x, ratio_y, shape)`: Scales the shape by `ratio_x` along the x-axis and `ratio_y` along the y-axis
- `translate(x, y, shape)`: Shifts the shape by `x` amount along the x-axis and `y` amount along the y-axis
- `rotate(rad, shape)`: Rotates the shape by `rad` amount in radians in the *counter-clockwise direction*
- `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 the ratio of shape1 : shape2 equals to `frac`
- `stackn(n, shape)`: Stacks `n` times of shape with equal proportions (i.e. $\frac{1}{n}$)
- `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
- `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 the ratio of shape1 : shape2 equals to `frac`
- `flip_vert(shape)`: Flips the shape vertically along the x-axis
- `flip_horiz(shape)`: Flips the shape horizontally along the y-axis
- `make_cross(shape)`:

Example

`sail_bb`



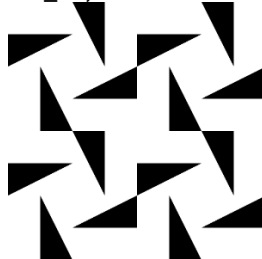
`make_cross(sail_bb)`



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

Example

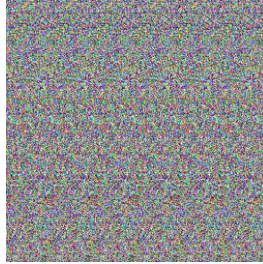
`repeat_pattern(2, make_cross, sail_bb)`



- `overlay(shape1, shape2)`: Overlays shape1 on top of shape2 with equal proportions [3D rune]
- `overlay_frac(frac, shape1, shape2)`: Overlays shape1 on top of shape2 such that the ratio of shape1 : shape2 equals to `frac` [3D rune]

3D Display Functions

- `stereogram(shape):`
Example
`stereogram(heart_bb)`



- `anaglyph(shape):`
Example
`anaglyph(heart_bb)`



- `hollusion(shape)`

Colors

- red
- pink
- purple
- indigo
- blue
- green
- yellow
- orange
- brown
- black
- white

Feel free to be experiment with all these runes and functions at your disposal! The most creative rune master will have his rune displayed on the Grandmaster's wall at the end of the Runes 2D and 3D contests!