Built-in functions in Remix

`do (executable)`
Execute the `executable` block.
Return: the value of the last statement executed.

`based on (original)`
Create a deep copy of `original`.
Return: the copy.

`show (output)`
Display `output` in the terminal window. A print statement. Does semi-useful things if `output` is a list, range or object.
Return: `none`.

`ask (description)`
Get terminal input from the user after showing `description` string.
Return: the string of input entered by the user.

`if (condition) (consequence)`
The if function. `condition` can be either a boolean value or a block, if it is a block it is evaluated and the result used. The `consequence` is a block which is evaluated if the `condition` is `true`.
Return: `none` if `condition` is `false`, the last statement result of the `consequence` if `condition` is `true`.

`if (condition) (consequence) otherwise (alternative)`
The if/else function. `condition` can be either a boolean value or a block, if it is a block it is evaluated and the result used. The `consequence` is a block which is evaluated if the `condition` is true, otherwise the `alternative` block is evaluated.
Return: if `condition` is true the last statement result of the `consequence`, if `condition` is false the last statement result of the `alternative`.

`type of (thing)`
Find the type of `thing` - currently `list`, `object`, `deferred` (a block of statements), `number`, `string`, `boolean` or `none`.
Return: the type.

`convert (string-input) to integer`
Convert the string `string-input` into an integer.
Return: the integer value.

`convert (item) to string`
Convert the `item` into a string. `item` can be any type.
Return: the string value.

`probe (value)`
Displays the *red* version of `value`. Useful for debugging.
Return: the *red* value as well.

`wait (number) secs`
Pauses for `number` seconds.
Return: `none`.

`length of (list)`
Calculate the length of `list` which is a list, range or object.
Return: the length.

`start (list)`
Begin the `list` iterator. `list` may be a list, range or object.
Return: `none`.

`next (list)`
Retrieve the current item from the `list` iterator and move the iterator on by one. `list` may be a list, range or object.
Return: the list item before moving the iterator on.

`end of (list)`
Determine if the list iterator is at the end of the list (i.e no more items). `list` may be a list, range or object.
Return: `true` or `false`.

`append (value) to (list)`
Append `value` onto the end of `list`. `list` may be a list or a string.
Return: the extended list.

`(start) to (finish)`
Create a range with integer values from `start` to `finish`. If `finish` is less than `start` the range decrements.
Return: the range.

`(list) (index/key)`
Extract the `index` or `key` value from the `list`. `list` can be a list or object. This function is called from the syntactic sugar `list[index/key]`.
Return: the value at the `index` or `key`.

`(list) (index/key) (value)`
Set the `index` or `key` position of `list` to the value. `list` can be a list or object. This function is called from the syntactic sugar `list[index/key] : value`.
Return: `value`.

`randomize`
Set the seed of the random function to the current time value.
Return: the current time.

`randomize with (seed)`
Set the seed of the random function to `seed`.
Return: `seed`.

`random (max-value)`
Generate a random value of the same type as `max-value`. `max-value` can be an integer (the range is from 1 to `max-value` inclusive), real number (the range is from 0 to `max-value` exclusive), colour (produces a colour in RGB format with each component limited to the corresponding value in `max-value`), and other possible types.
Return: the random value.

`sine (degrees)`
Return: the sine of `degrees`.

`cosine (degrees)`
Return: the cosine of `degrees`.

`arctangent (change-y) over (change-x)`
Return: the arctangent of `change-y` / `change-x`.

`√ (value)`
Return: the square root of `value`.

Built-in drawing functions in Remix

`show paper`

Display the graphical output window (paper) with all drawings made so far. As Remix is currently single threaded, this function blocks until the window is closed.
Event handlers such as animations need to be setup before this so they still work while the window is showing.
Return: `true`.

`draw on layer (drawing-layer)`

Change the drawing layer to `drawing-layer`. `drawing-layer` is a positive integer. Layer 0 is the background.
Return: `none`.

`clear layer (drawing-layer)`

Clear all drawing from the `drawing-layer`.
Return: `none`.

`(colour) paper of (width) by (height)`

Prepare the paper with background `colour` and size `width` × `height`. Allocates 2 times each dimension for high definition displays for the background layer. Layer 0 is currently an image which can be drawn to. This layer is filled with `colour` by this function.
Return: `none`.

`no outline`

Prevent shapes from being drawn with outlines in the current drawing layer.
Return: `none`.

`draw with (pen)`

Set the current drawing pen with the colour and size of `pen` in the current drawing layer.
Return: `none`.

`do not fill`

Prevent shapes from being filled in with a colour in the current drawing layer.
Return: `none`.

`fill colour (colour)`

Set the current fill colour to `colour` for shapes in the current drawing layer.
Return: `none`.

`draw line from (start) to (finish)`

Draw a line from `start` to `finish` with the current pen in the current drawing layer.
Return: `none`.

`draw box from (bottom-left) to (top-right)`

Draw a box from `bottom-left` to `top-right` with the current pen and fill colour in the current drawing layer.
Return: `none`.

`draw circle of (radius) at (centre)`

Draw a circle of size `radius` at position `centre` with the current pen and fill colour in the current drawing layer.
Return: `none`.

`draw (shape)`

Draw the `shape` with the current pen in the current drawing layer. A shape is a list of vertices centred on (0, 0) with a size, location, colour, and heading. The heading is in degrees clockwise.

Return: `none`.

`animate (animate-block)`

Using the current animation rate repeatedly execute the statements in `animate-block` when the graphical output window (paper) is shown.

Return: `none`.

`(rate) times per sec`

Set the current animation rate to `rate`.

Return: `none`.

`animation off`

Stop any animations.

Return: `none`.