Crib Sheet

Built-in Python functions:

`abs(`*`x`*`)`
: Return the absolute value of *`x`* (an `int` or a `float`).

`float(`*`x`*`)`
: Convert argument *`x`* (a string or number) to a real number, if possible.

`int(`*`x`*`)`
: Convert argument *`x`* (a string or number) to an integer, if possible. A real number argument will be truncated towards zero.

`max(`*`a, b, c, ...`*`)`
: With two or more arguments, return the largest argument.

`min(`*`a, b, c, ...`*`)`
: With two or more arguments, return the smallest argument.

`range(`*`stop`*`)`
: Return the list of integers `[0, 1, 2, ..., `*`stop`*` - 1]`.

`range(`*`start, stop`*`)`
: Return the list of integers `[`*`start`*`, `*`start`*` + 1, `*`start`*` + 2, ..., `*`stop`*` - 1]`.

`raw_input(`*`prompt`*`)`
: Read a string from the keyboard and return the string. The *`prompt`* string, if provided, is printed without a trailing newline before reading the string.

`round(`*`x, n`*`)`
: Return the `float` that results when the value of argument *`x`* (a `float`) is rounded to *n* digits after the decimal point.

`str(`*`x`*`)`
: Return a printable string representation of argument *`x`*.

`math` module:

`pi`
: High precision constant for Pi.

`cos(x)`
: Return the cosine of x (measured in radians).

`sin(x)`
: Return the sine of x (measured in radians).

`tan(x)`
: Return the tangent of x (measured in radians).

`atan2(y,x)`
: Return the arc tangent (measured in radians) of y/x.
Unlike atan(y/x), the signs of both x and y are considered.

`CU_media` module:

`choose_file()`
: Launch a file chooser and return a string containing the name of the file that was selected.

`load_picture(`*`filename`*`)`
: Create a `Picture` object from the contents of *`filename`* (a string) and return it.

This crib sheet continues on the following page.

Functions that work with `Picture` objects:

`show_external(`*`pict`*`)`
: Display the picture.

`get_height(`*`pict`*`)`
: Return the height of the picture in pixels (an `int`).

`get_width(`*`pict`*`)`
: Return the width of the picture in pixels (an `int`).

`get_pixels(`*`pict`*`)`
: Return a list containing all the pixels in the picture.

`get_pixel(`*`pict, x, y`*`)`
: Return the pixel at the specified (*`x`*, *`y`*) location in the picture.

Functions that work with `Pixel` objects:

`get_red(`*`pixel`*`)`, `get_green(`*`pixel`*`)`, `get_blue(`*`pixel`*`)`
: Return the values of the pixel's red, green and blue components (an `int` between 0 and 255).

`set_red(`*`pixel, amount`*`)`, `set_green(`*`pixel, amount`*`)`, `set_blue(`*`pixel, amount`*`)`
: Set the values of the pixel's red, green and blue components to the specified *`amount`* (which must be an `int` between 0 and 255).

`get_color(`*`pixel`*`)`
: Return a `Color` object that represents the colour of the pixel.

`set_color(`*`pixel, color`*`)`
: Set the pixel's colour to the specified *`color`* (a `Color` object).

`get_x(`*`pixel`*`)`
: Return the x-coordinate of the pixel's location.

`get_y(`*`pixel`*`)`
: Return the y-coordinate of the pixel`s location.

Functions that work with `Color` objects:

`Color(`*`red`*`, `*`green`*`, `*`blue`*`)`
: Return a `Color` object with the specified amounts of *`red`*, *`green`* and *`blue`*. (Each amount must be an `int` between 0 and 255).

`distance(`*`color1, color2`*`)`
: Return the distance between the `Color` objects bound to *`color1`* and *`color2`*.

`get_red(`*`color`*`)`, `get_green(`*`color`*`)`, `get_blue(`*`color`*`)`
: Return the values of the color's red, green and blue components (an `int` between 0 and 255).