CS106A Reference

Karel:

Base Karel commands:

```
move()
turn_left()
put_beeper()
pick_beeper()
```

Conditions:

`if` *condition*:
code run if condition passes

`if` *condition*:
code block for "yes"
`else:`
code block for "no"

Karel program structures:

```
# Comments can be included in any part
# of a program. They start with a #
# and include the rest of the line.

def main() :
```
    code to execute

declarations of other functions

Loops:

`for i in range(`*count*`):`
code to repeat

`while` *condition*:
code to repeat

Names of the conditions:

```
front_is_clear()
front_is_blocked()
beepers_present()
no_beepers_present()
beepers_in_bag()
no_beepers_in_bag()
left_is_clear()
left_is_blocked()
right_is_clear()
right_is_blocked()
facing_north()
not_facing_north()
facing_south()
not_facing_south()
facing_east()
not_facing_east()
facing_west()
not_facing_west()
```

Function Declaration:

`def` *name*`():`
    code in the body of the function.

Common Python List Functions:

`alist.append(x)`	Appends `x` to the end of `alist`
`alist.insert(index, x)`	Inserts `x` into `alist` at `index` After the insert function, x will be at index
`alist.pop()`	Removes the last element of `alist`
`alist.sort()`	Sorts the element of `alist` into ascending / alphabetical order
`len(alist)`	Returns the number of elements (length) of `alist`
`alist[index1:index2]`	Returns a new list from `index1` up to but not including `index2`

Graphics commands:

`canvas = Canvas(width, height, 'Name')`	Creates a new drawing canvas with a width of `width` pixels and a height of `height` pixels with a title of `Name`
`canvas.create_line(x1, y1, x2, y2, color)`	Draws a line from (`x1`, `y1`) to (`x2`, `y2`) in `color`
`canvas.create_rectangle(x1, y1, x2, y2, color)`	Draws a rectangle bounded by (`x1`, `y1`) and (`x2`, `y2`) in `color`
`canvas.create_oval(x1, y1, x2, y2, color)`	Draws an oval bounded by the rectangle (`x1`, `y1`) and (`x2`, `y2`) in `color`.
`canvas.create_text(x, y, anchor=anchor, font=font, font_size=font_size, text='some_text', color=color)`	Draws the text `text` at position (`x`, `y`) in `font` with `font_size` in `color`, anchored by `anchor`. `anchor` can be one of **`"n"`, `"ne"`, `"e"`, `"se"`, `"s"`, `"sw"`, `"w"`, `"nw"`, or `"center"`**, that describes what side of your text you want to position at the (`x, y`)