CSc 110 Final
Cheat Sheet

```
def name(parameters):
    statement(s)
    ...
    return expression
```

```
variable = type(input(prompt))
```

```
for name in range(start, stop + 1):
    statement
    statement
    ...
    statement
```

```
if test:
    statement(s)
elif test:
    statement(s)
else:
    statement(s)
```

`Math` functions

Function name	Description
`math.ceil(`*value*`)`	rounds up
`math.floor(`*value*`)`	rounds down
`math.log(`*value, base*`)`	logarithm
`math.sqrt(`*value*`)`	square root
`abs(`*value*`)`	absolute value
`min(`*value1, value2*`)`	smaller of two values
`max(`*value1, value2*`)`	larger of two values
`round(`*value, digits*`)`	rounds the value to a value with digits numbers after the decimal point

Constant	Description
`math.e`	2.7182818...
`math.pi`	3.1415926...

Random

`random.randint(min, max)`

returns a random number between min and max

`String` functions

Function	Description
str`.lower()`	returns a new string with all lowercase letters
str`.upper()`	returns a new string with all uppercase letters
`len(`*str*`)`	returns the length of the string

File functions

fname = `open(`**file_name**`)` - opens a file

name = **fname**`.read()` – reads all of the text of fname into a string

name = **fname**`.readlines()` – reads the text of fname into a list where each line is its own element

name = **str_name**`.split()` – splits a string on whitespace

name = **str_name**`.strip()` – strips all whitespace off the beginning and end of the string

List functions

List assignment**:** **name**`[`**index**`]` `=` **value**
List creation**:** **name** `=` `[`**value**`]` `*` **length**
name `=` `[`**value**`,` **value**`,` **value**`,` `…,` **value]**
List length**:** `len(`**name**`)`

Sets

Adding to a set**:** **name**`.add(`**value**`)`
Set creation**:** `set()`
Set size**:** `len(`**name**`)`

Dictionaries

Adding to a dictionary**:** **name**`[`**key**`]` `=` **value**
Accessing elements : **othername** `=` **name**`[`**key**`]`
Dictionary creation**:** **name** `=` `{}`
Dictionary size**:** `len(`**name**`)`

in keyword: used to check if an element is in a structure