

math functions and errors

```
// whole #      syntax -error in  a+=1 ->
division        code form      a=a+1

% find the      semantic -error  a/=1 ->
remainder      in logic        a=a/1

** read right to left

/ always gives a float

x%y = x-y*(x//y)

{:.xf} -round float to x decimals points
```

Numerical functions

```
abs() -gives absolute values

pow(number,exponent)

int() -turns object to integer

float() -turns object to float

-round(number,what to round to)

-max() -finds max in list

min() -finds min in list
```

Import Math

```
math.pi is pi

math.sqrt(x) -takes square root of x

math.log(base,variable)

math.e is e

math.ceil(x) -returns smallest integer
greater than x

math.trunc(x) -removes decimal values from
x
```

String manipulation

```
\n -end current line and start new

\t -creates a "tab"

' or \" -doesnt interpret as string delimiter

\\ -true backslash

, creates default + creates no space.
space of 1. can   Cant be use to put
be use with all   together string and
object types      integer (or float)

string.lower() -turns all letters to lowercase

string.upper() -turns all letter to uppercase

string.capitalize() -only first letter in string is
uppercase

string.title() -first letter of every word is
uppercase

string.replace(string you want to replace,
what you replace with)

string.find(what    gives    returns -
you want to find)  number of  1 if it
                   location in  cant find
                   string

string.count(what   will give  returns 0
you want to        number     if nothing
count)            counted     to count

string.strip(what you want to remove)

sep="" -used to change default space
between objects seperated by ,
```

Variables and object interactions

```
Variables start with letter or _

Variables can't have spaces

string and integers cant be added

string and sting cant be multiplied
```

Lexographical Order

```
Uppercase < Lowecase      true

a < b                      true
```

Tuples

```
len(tuple) -gives the # of objects in a tuple

cant update a tuple

tuples and strings are immutable

tuples can be added

ex r=(1,1)

r+s=(1,1,'a','b')
```

Boolean Logic

```
If, elif, and else must all end with :

any commands following must be indented

can us <, <=, >, >=, ==, or !=

!= is for not equal to

cant use = since that is for variable
assignment

output values are only true or false
```

Functions

```
def function_name(Variable):

every line which comes below is indented

can end with return statement to return a
variable
```



By Casey Schmidt
(Scribmaster12)

cheatography.com/scribmaster12/

Published 13th February, 2020.
Last updated 13th February, 2020.
Page 1 of 1.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>