

Function

print()	display given info.
input()	receives info from user
int()	converts a value to an integer
float()	converts a value to a floating point
str()	converts a value to a string
len()	the length of the string
==	equal to
<=	less than or equal to
>=	more than or equal to
%	Modulo, Find the remainder
#	Comment, no effect on programming
while	continues
True	
for	looping forever
if...then	"if true" will work, "if false" won't work loop stops
//	int outcome
/	float outcome

Multiplication and Exponent

string * string	Crash
string*number	combines the string multiple times
number*number	Math - Multiply

Multiplication and Exponent (cont)

string**number	Crash
number**number	Math - Exponent
string**string	Crash

Palindrome

```

"""
Cliff 1003
" " "
def isPalindrome (word):
    index = 0
    # word[0] len(word)-1 -0
    number OfLoops
= 0
    while index <
1/2*len (w ord):
        num ber -
OfLoops += 1
        pri -
nt( 'Co mpa rin g', wor d[i -
ndex] ,word[ len (wo rd) -1- -
index])
        if word[i -
index] == word[1 en( wor d)- 1-
i ndex]:
            index = index + 1
        else:
            print ('loop s:' ,nu mbe rOf -
Loops)
            return False
        print ('loop -
s:' ,nu mbe rOf Loops)
        return True
while True:
    use r_input =input -
("what is your word? ")
    if user_input == " qui -
t":

```

Palindrome (cont)

```

> break
print (len(user_input))
myword = isPalindrome(user_input)
if myword == True:
    print ((user_input),"is a palindrome")
else:
    print ((user_input),"isn't a palindrome")

```

Max Value in list / Max value

```

def max2 (num1,num2):
    max value = num1
    if num2 > maxvalue:
        max value = num2
    return maxvalue
print (max2 (4,3))
print (max2 (3,22))
answer = max2 (1,5)
print (answer)
def max3 (n um1 ,nu m2, num3):
    max value = num1
    if num2 > maxvalue:
        max value = num2
    if num3 > maxvalue:
        max value = num3
    return maxvalue
print (max3 (10,7,8))
print (max3 (7,10,8))
print (max3 (7,8,10))
def maxlis t(l ist):

```

Max Value in list / Max value (cont)

```
> maxvalue = list[0]
for num in list:
    if maxvalue < num:
        maxvalue = num
return maxvalue
print (maxlist(range(0,101)))
mylist = 1,5,76,23,78,34,5678,2,5,8,675,2,-
6,86,54,23,6,8
print (maxlist(mylist))
```

Randomizer

```
import random
intlist = [1,2,3 ,4,5]
random_int = random.choice(intlist)
print (intlist, random_int)
fplist = [2.0,2.1, -
2.2 ,2.3 ,2.4,2.5]
random_fp = random.choice(fplist)
print (fplist, random_fp)
strlist = ['a',' -
b', 'c', 'd', 'e','f']
random_str = random.choice(strlist)
print (strlist, random_str)
mylist = [1,9.9 ,"hello "]
random_mylist = random.choice(mylist)
print (mylist, random_mylist)
myvar1 = 1
myvar2 = 2
myvar3 = 3
varlist = [myvar1, myvar2, -
myvar3]
random_var = random.choice(varlist)
```

Randomizer (cont)

```
> print (varlist, random_var)
```

commands

import	imports program
random	given written program
random.choice()	random items in the list

Vocabulary

Variable something that can change

String a list of characters

Print display given info.

Syntax Grammar/Structure of language

Modulo Find the remainder

Boolean True/False

Reverse word

```
while True
word = input( " Please enter a
word")
index = 0
reverse = " "
while int(index) < len(word)
    reverse = word[index] +
(reverse)
    index = int(index) + 1
print ("Re verse: ", reverse)
```

Conditionals

if if the statement is true then do the
else command under. Else do command
under else

Conditionals (cont)

elif Similar to if else, but allows more
conditions. (short abbreviation for
if else)

for Will loop through every element
of the set

while A loop condition with conditions
loop 1.initial value 2.ending condition
3.update

while While the statement is true keep
true looping

Concat Joins the strings by linking then
enation end to end

Convert into binary

```
user_number = ""
while user_number != '0 ':
    user_number = input
("Enter a number to convert into
binary ")
    number = int(user_number)
    binary_string = " "
    while (number > 0):
        remainder = number%2
        binary_string =
str(remainder) + binary_
_string
        number = number // 2
print ("Binary string is " ,bi -
nary_string)
```

Finding the triangle (area,volume)

```
def areaoftriangle(num1,num2):
    area = 1/2*num1*num2
    return area

user_base = float( input( 'what is your base of the triangle; '))
user_height = float( inp -
ut( 'what is your height of the triangle; '))

print( 'The area of the triangle is: ',area of triangle( user _ba se, use r_h eight))

def volumeofprism( base ,height ,p ris m_h eight):
    volume = areaof tri -
angle( base,height) *
prism_ height
    return volume

user_prism = int(input( 'Enter the prism height; '))
print( "The volume of the prism is; ", volumeof prism( user _ba se, use r_h eig ht, use r_p -
rism))
```

Guessing Game

```
import random
score = 0
chances = 5
print( "Sc ore :", score)
print( "Ch anc es: ", chances)
mylist = ['apple', 'banana',
'orange', 'mango', 'cherry']
print( mylist)
random_ item = random.ch oic -
e(m ylist)
while chances > 0 :
```

Guessing Game (cont)

```
> user_guess = input("Guess a word:")
if user_guess == random_item:
    print ("That's correct!")
    score = score+100
    print ("Score:", score)
    print ("Chances:", chances)
    random_item = random.choice(mylist)
else:
    if user_guess in mylist:
        print ("Sorry, wrong choice!")
        chances = chances-1
        print ("Score:", score)
        print ("Chances:", chances)
    else:
        print ("Sorry, that is not even in the list")
        chances = chances-1
        print ("Score:", score)
        print ("Chances:", chances)
print ("Game over! The word was", random_
item)
print ("Final Score", score)
```

Countdown Machine

```
user_number = input("What number do you want to countdown? ")
number = int(user_number)
countdown_string = ''
while number > 0
countdown = countdown_string - str(number) + " "
number = number - 1
```

Countdown Machine (cont)

```
#print(number)
print( countdown_string)
```

Math Operation Function Writing

```
def calc(num1,num2,operation):
    if operation == " sum ":
        return sum( nu -
m1, num2)

        elif operation == " pro -
duc t":
            return produc -
t( n um1 ,num2)

            elif operation == " dif -
f":
                return diff( n -
um1 ,num2)

                elif operation == " -
div ":
                    return div( nu -
m1, num2)

def sum(a,b):
    return (a+b)

def produc t(a,b):
    return (a*b)

def diff(a,b):
    return (a-b)

def div(a,b):
    if b != 0 :
        return a//b
    else:
        print ("ER -
ROR ")

print( cal c(1 0,0 ,"di v"))
print( cal c(1 ,2, " sum "))
print( cal c(4 ,2, " dif f"))
print( cal c(9 ,3, " div "))
print( cal c(2 ,12 ,"pr odu -
ct"))
```

Naming Convention

Rules for giving names

- letter
- numbers
- underscore_

Valid name

- _mystr
- my6
- Kawazoe_Kyousuke

Invalid name

- 3my = "whatever" #can't start with a number
- Kawa zoe = "whatever" #can't have space
- first-name = "something" #can't have "-"

True / False

True or ... / ... or True	True
False and ... / ... and False	False

22/03/16 code

```
'''
theList = ['1','2','3']
for item in theList:
    print( item)
'''
'''
index = 0
whileList = ['1','2', '3', '4']
while index < len(whileList):
    print (whileList[index])
    index = index + 1
```

22/03/16 code (cont)

```
> '''
'''
while True:
    user_input = (input("your word?: "))
    print ("your length of the word is: ",len(user_input))
    if user_input == "exit":
        break
'''
'''
def theFunction():
    while True:
        user_input = input("word: ")
        if user_input == "stop":
            break
theFunction()
'''
'''
def computeThis(a1,b2):
    product = a1*b2
    print (product)
    return
computeThis (2,5)
'''
'''
def finalFunction(string):
    print ("**,(string),**")
    return
```

22/03/16 code (cont)

```
> finalFunction ("cliff")
'''
```

print item using while

```
index = 0
whileList = ['1','2', '3', '4']
while index < len(whileList):
    print (whileList[index])
    index = index + 1
```