addition

string + number	crash
string + string	combine together
number + number	math-addition

multiplication

string * string	CRASH
string * number	combines the strings multiple time
number * number	math (multiply)
string ** number	CRASH
number ** number	Exponent(Math)
string ** number	CRASH

condition

If	If the statement is true then do
:then	command under then else do
else	command under else
while	While this is true loop the command under the conditional
While True	loops forever
for each item in name of list	For every item in the list repeat the command under the loop that many times. (a string is a list too)

condition (cont)

for...in...	loop forever

example3(convert number to hex)

```
user_number = input("please enter a number: ")
number = int(user_number)
hex_string = ' '
while (number > 0):
    remaider = number % 16
    if remaider == 10:
        remaider = 'A'
    elif remaider == 11:
            remaider = 'B'
    elif remaider == 12:
            remaider = 'C'
    elif remaider == 13:
            remaider = 'D'
    elif remaider == 14:
            remaider = 'E'
    elif remaider == 15:
            remaider = 'F'

    hex_string = str(remaider) + str(hex_string)
    number = number // 16
print ("Hexadecimal string is 0x", hex_string)
```

example4(countdown)

```
user_number = input("Please enter a number: ")
number = int(user_number)
countdown_string = ''
while number > 0:
    countdown_string = countdown_string + str(number)
    number = number - 1
print (countdown_string)
```

example5(circle radious)

```
while True:
    user_radius = input("What is your radius of a circle? ")
    radius = float(user_radius)
    pi = float(3.1415)
    area = (pi) * (radius) ** 2
    print("The area of the circle", area)
```

volumeofprism

```
def areaoftriangle(b,h):
    area = 0.5*b*h
    return area
user_base = float(input("enter the base of the triangle: "))
user_height = float(input("enter the height of the triangle: "))
print ('the area of trianglr is', areaoftriangle,(user_base, user_height))
def volumeofprism(b,h,l):
    volume = areaoftriangle(b,h)*l
    return volume
user_lenght = float(input('lenght of prism: '))
print('the volume of prism is', volumeofprism(user_base,user_height,user_lenght))
```

Printing values

Printing values:
print("hello", "there") #displays hello there
print("hello" + "there") #displays hellothere

Combining Strings (Concatenation)

Combining Strings (Concatenation)
"hi" + "there" == "hi there"
"hi" * 5 == "hihihihihi"

loop

While Loop with List:
thelist = [4, 3, 2, 1, 0]
index = 0 # start at the first item
while index < len(thelist):
print (thelist[index]) #prints each item
index = index + 1
For-Loop with List:
forlist = [3, 4, 5, 2, 1]
for item in forlist:
print(item)

Vocabulary

floating point	decimal number
boolean	true or false
variable	hold a value and can be change
string	a list of character such as number, letter and symbol
integer	whole number or counting
syntax	grammar or structure of lan
value	the number or string can be store in valuable

function

print(-)	display information on screen
input(-)	receive information from user
int(-)	converts a value to an integer
float(-)	change number to decimal number
str(-)	a list of number, letter and symbol
len(-)	the length of string
'''	Multi-line comment
#	One line comment not include in code

letter command

```
print (name.upper()) all capital
print (name.lower()) all not capital
print (name.capitalize()) first letter capital
print (name.title()) every first letter of every word capital
```

example (reverse word)

```
word = input("Type in an word: ")
reverse = ""
for letter in word:
    reverse = letter + reverse
print ("Reverse: ", reverse)
```

example6(random)

```
import random
intlist = [1, 2, 3, 4, 5, 6, 7, 8, 9]
random_int = random.choice(intlist)
print (intlist, random_int)
```

example6(random) (cont)

```
fplist = [1.3112354, 2.5145496, 3.857498, 4.65454564, 5.7418523, 6.321956, 7]
random_fp = random.choice(fplist)
print (fplist, random_fp)
strlist = ["a", "s", "d", "f", "g", "h"]
random_item = random.choice(strlist)
print (strlist, random_item)
myvar1 = 1
myvar2 = 2
myvar3 = 3
varlist = [myvar1, myvar2, myvar3]
random_var = random.choice(varlist)
print (varlist, random_var)
```

example 8

```
def printdefinition(word):
    if word=="variable":
        print("""a variable is value that can change""")
    elif word=="function":
        print("""a function is define box of code that can be reuse""")
    elif word=="parameter":
        print("""a parameter is value given to function""")
    elif word=="argument":
        print("""a argument is value given to function""")
    elif word=="function call":
        print("""a function call is use the function code""")
    elif word=="string":
        print("""a string is list of character""")
    else:
        print("""unknown""")
```

example 8 (cont)

```
    return
while True:
    user_input = input("enter word
")
    printdefinition(user_input)
```

example9 (largest number)

```
def max2(num1, num2):
    if num1 < num2:
        maxvalue = num2
    else:
        maxvalue = num1
    return maxvalue
print(max2(4,5))
print(max2(6,5))
def max3(num1,num2,num3):
    maxvalue = num1
    if num2 > maxvalue:
        maxvalue = num2
    if num3 > maxvalue:
        maxvalue = num3
    return maxvalue
print(max3(1,2,3))
print(max3(1,4,3))
print(max3(5,2,3))
def maxlist(list):
    maxvalue = list[0]
    for item in list:
        if item > maxvalue:
            maxvalue = item
    return maxvalue
mylist = [1,2,3,4,5]
print(maxlist(mylist))
```

range

```
numberlist = range(5)
# is the same as creating the following list
numberlist2 = [0, 1, 2, 3, 4]
for num in range(100):
print (num) # prints all numbers from 0 – 99
for num in range(5, 50):
print(num) #prints all numbers from 5 - 49
```

calculation

==	equal
!=	not equal
<	less than
>	more than
<=	less than or equal to
>=	more than or equal to
%	modulo (find remainder)
+	add
-	subtract
*	multiply
/	divide and quotient is float
//	divide and quotient is integer
**	exponent

naming rule

Rules for naming variables:

- letters
- numbers(not first letter)
- underscores (_)
- can start with letters or underscores ONLY
- NO SPACES

example2(convert to binary)

```
user_number = input("Enter number
to convert to binary : ")
number = int(user_number)
binary_string = ''
while (number > 0):
    remainder = number % 2
    binary_string =
str(remainder) +
str(binary_string)
    number = number // 2
print ("Binary string
is",binary_string)
```

example7

```
def bacon():
    print("hello it's bacon")
    return
bacon()
def myprint(text):
    print ("***"+str(text)+"***")
    return
myprint(88)
def myprintnew(text, decoration):
    print(decoration+str(text)+deco
ration)
    return
myprintnew(101, "-=-=-=-=-=-")
def doubleit(number):
    return number*2
print(doubleit(12121212))
print(doubleit(doubleit(12)))
def areaofcircle(radius):
    if radius <= 0:
        return "=-=-=-=-="
    pi=3.1415
    area=pi*radius*2
```

example7 (cont)

```
    return area
user_radius =
float(input("radius:"))
print("the area is
",areaofcircle(user_radius))
```

area of triangle

```
def areaoftriangle(b,h):
    area = 0.5 b h
    return area
user_base = float(input("enter the
base of the triangle: "))
user_height = float(input("enter
the height of the triangle: "))
print ('the area of trianglr is',
areaoftriangle,(user_base,
user_height))
```

palindrome

```
reverse = ""
letter_num = 0
user_input = input("type in a
word:")
user_input = str(user_input)
while letter_num <
len(user_input):
    reverse =
user_input[letter_num] + reverse
    letter_num = letter_num + 1
if reverse == user_input:
    print("the string is
palindrome")
else:
    print ("the string is not
palindrome")
```

vocab

Vocabulary:
syntax, variable, Boolean, string, integer, float, list, comment, character, conditional, modulo, if/elif/else, loop, range, parameter, argument, function call

list

Lists:
mylist = [2,3,4,5] # create a list
#select an item from a list
print (mylist[0]) #selects first item and displays 2
len() determines the length of the list
print (len(mylist)) # displays 4
mylist.append(5) # adds an item to the end of the list