PYTHON

Types of Codes

- Sequential
- Repeated (while) (for)
- Conditional

Reserved Words

False	for	nonlocal	elif	
class	lambda	and	try	
return	continue	del	or	
is	True	global	yield	
Finally	def	not	assert	
None	from	with	else	
if	while	as	import	
pass	break	except	in	raise

continue - the continue statement ends the current interation and jumps to the top of the loop and starts the next ~~int~~ iteration.

Indefinate Loop → while loop

Definite loop → for loop

★ 'is' is used basically for True, False or None

★ quit()

7.1/.2 Files / Processing files

★ To read all the lines in a file :-
we use a for loop :

```
Eg => { for line in file :
            print (line) }
```

★ To read the complete file at once :-
we use .read() function

```
Eg. => { file = open ("File.txt")
         inp = file.read() }
```

8.1 Lists

(more than one value in a variable)
collection of

A list can contain any ~~tea~~ type of data.

★ 'strings' are immutable

★ 'range' function returns a list
=> Eg. Print(range (4))
0,1,2,3

★ len → tells the length of the list (no. of items in a list).

★ count → tells no. of items in a list that matches the item given as ~~parameter~~ argument

```
Eg >>> a = [ 1, 2, 3 , 1, 4 , 6, 1, 3 ]
   >>> a.count(1)
   output : 3
```

★ pop
gives you the item & removes it afterwards

★ insert (a, b)
a → index
b → value

★ remove
it removes an item from a list which comes first

★ sort → sorts the list

★ reverse

★ index → tells you the index of an ~~value~~ item

immutable
↓
#8.3 Lists and Strings

★ .split() → it divides the list into list items.

```
Eg. >>>abc = "with these words"
    >>>llist = abc.split()
    >>>print(llist)
    ['with', 'these', 'words']
```

If we don't define deliminiter in the split function then it splits according to the spaces between the words.

List Summary (★ lists are mutable)

- Lists and definite loops
- Indexing and lookup
- List mutability
- Functions: len, min, max, sum
- Slicing lists
- list methods: append, remove
- Sorting lists
- Splitting strings into lists of words
- Using split to parse strings.

9.1 Dictionary

★ Very useful in finding word frequency

★ Get method

The pattern of checking to see if a key is already in a dictionary and assuming a default value if the key is not ~~p~~ there ~~in~~.

Eg. for this program

```
if name in counts :
    x = counts [name]
else :
    x = 0
```

⇓

Important

```
x = counts. get (name, 0)
                  ↑     ↑
                 key   Default
```

★ .keys() → gives a list of keys

★ .values() → gives a list of values

★ .items() → gives a list of (key, value) pair ie a list of ~~of~~ (key, value) tuple.

10 Tuples (immutable)

Tuples are another kind of sequence that functions much like a list - they have elements which are indexed starting at 0.

★ Can't sort a tuple

★ Can't append a tuple can't extend a tuple

★ Can't reverse a tuple

⇒ it only supports [count, index]

★ We can also put a tuple on the left hand side of an assignment statement

eg. (x, y) = (4, 6)

y = 6 , x = 4

★ Tuples are comparable

eg. (0,1,2) < (5, 1, 2)

True

★ .items( )

~~# [illegible]~~

Relational operators

x	!=	y	not equal to	type(boolean) bool
x	>	y	greater then	
x	<	y	less then	
x	>=	y	greater equal to	
x	<=	y	less equal to	

Logical operators (three types)

and	both condition	type(bool)
or	either condition	
not		

★ Recursion

A function that calls itself is recursive.