005: Loops

Learning Outcome:	Loops: while, for

Definitions/Concepts	
Loops	• Loops are iterative statements in Python. • Iteration means repetition. • If we want to execute a command again and again, we use Loops.

Types of Loops	• In python we have two kinds of loops,
	  ○ while loop
	  ○ for loop

while loop	

Syntax:
`while(conditon):`
`  Statement1`
`  Statement2`
	• The condition written in brackets is followed by **colon :**
	• Statement 1 and 2 are the statements of the loop written after indentation (space from left hand side)

• In “While loop”, the set of statements gets executed as long as the given condition is true. • When the condition becomes false, then the statements in the loop are not executed. • The statements of the loop should contain an **update statement**.	

Eg.
`n=1`
`while(n<5):`
`  print(n)`
`  n+=1`
	The three main parts of the loop here are:
	1. Initial value of variable: **n=1**
	2. Conditional Statement: **n<5**
	3. Update Statement: **n+=1**
	The variable which runs the loop is often called control variable, here **n** *is the control variable.*

Example of using while loop	

To print the first 10 natural numbers.	`num=1`
	`while(num<=10):`
	`  print(num)`
	`  num+=1`

Update Statements		
• Update statements can be of two major types: increment and decrement statements. • Increment statement increases the value of the control variable and decrement statement decreases the value of the control variable.		

+=	Eg:
	`>>> a=10`
	`>>> a+=1`
	`>>> a`
	`11`
		• It increases the value of a by 1.
		• It is same as writing **a=a+1**

-=	Eg:
	`>>> a=10`
	`>>> a-=1`
	`>>> a`
	`9`
		• It decreases the value of a by 1.
		• It is same as writing **a=a-1**

input() function
• It is used to take value from the user. • Syntax: `var=input("Input something ")` • The input by the user in this case is stored in the variable named **var** • The data type of var is **String**

- To change the datatype of **var,** we can use different functions like:
  - int()
  - float()
- The process of changing datatype of a variable is called **Type casting**
- We can even change an int or float value to String using str() function.

Type Casting	
int()	• It can convert String or float into integer. • In case of a floating point, it gives the integral value of the number only. • If the String is a letter or special character then the function doesn't work and we get a value error

```
>>> int('90')
90
>>> int(8.9)
8
>>> int('k')
ValueError: invalid literal for int() with base 10: 'k'
```

float()	• It can convert a String or integer into a floating point number. • In case of an integer, it inserts a zero after decimal. • If the String is a letter or special character then the function doesn't work and we get a value error.

```
>>> float('90')
90.0
>>> float(9)
9.0
>>> float('k')
ValueError: could not convert string to float: 'k'
```

str()	• It can convert numbers into String	`>>> str(9.00900)`
		`'9.009'`
		`>>> str(234)`
		`'234'`

for loop	

Syntax:
`for item in sequence:`
`statment1`
`statement2`
	• Statement 1 and 2 are the statements of the loop written after indentation (space from left hand side)
	• **item in sequence** refers to one value in a sequence(series of value) at a time.

• for loop can be used to iterate over a sequence or series of values. • And this sequence can be a list, set, dictionary, tuple or string. • There is no need for an update statement here.	

Eg.
`for i in (1,2,3,4,5):`
`print(i)`	

• The main parts of the loop here are: ○ Iteration: **i in (1,2,3,4,5)** ○ Body of the loop: **print(i)** • **Iteration** can be broken into three parts: ○ The control variable: **i** ○ The keyword: **in** ○ The sequence of values: **(1,2,3,4,5)**	

range() function
• It is used to form a sequence of numbers. • We can input three things in this function, ○ Starting value ○ Ending value ○ Stepping value
Syntax: `range(start,stop,step)`

- The values in range will be less than value of **stop**
  - So, range(10,15,1) will have values 10,11,12,13,14
- Default value of **step** is one.
  - So, range(10,15) will have values 10,11,12,13,14
- Default value of **start** is zero.
  - So, range(5) will have values 0,1,2,3,4

Example of using for loop

To display all elements of a list.

```
a_list=['toppr','learn','practice','ask','codr']
for a in a_list:
    print(a)
```

To display all the even numbers between 100 and 120

```
for i in range(100,121,2):
    print(i)
```

- Here, the stop value of range is 121, because we want to display 120 also.
- The step value is 2, because we wanted to display all the even no.s starting from 100.

Activity links and Solutions

Student Activity 1: While Loop

#Activity 1: Input a number from the user and display all its factors

```
number=int(input("Enter a number: "))
```

The above statement will take input from the user and convert it into an integer.

```
factor=1

while(factor<=number):
    if(number%factor==0):
        print(factor,"is a factor of",number)
    factor+=1
```

- The control variable of the while loop here is **factor**
- The loop will run until the **factor** becomes more than **number**.
- The **if** statement inside the **while** loop is to check if the number stored in the variable **factor** is actually a factor or not.
- **number%factor** gives the remainder when number is divided by factor.
- If this remainder is equal to zero, that means, the number stored in the variable factor is actually a factor.
- To check for the next number, the value of factor increases by 1.

Student Activity 2: For Loop

#Activity 1: Display multiplication table of 25

```
for i in range(1,11):
    print(25,"X",i,"=",(25*i))
```

- range(1,11) will have numbers from 1 to 10, as the default step value of range is 1
- All the things written within quotes will be displayed as it is.
- The numbers and value of variables will be displayed as output.

```
print(25,"X",i,"=",(25*i))
```

For ***i=4***

```
25 X 4 = 100
```

The value of mathematical expression **(25*i)** is also displayed in the output.

#Activity 2: Take a number from the user. Display multiplication table of that number.

```
num=int(input("Enter a number: "))
```

- The first step is to take input from the user and convert it into an integer, here the variable storing the number is **num**.
- The remaining step is quite similar to the previous problem, instead of the number 25, we here will use the variable **num.**

```
for i in range(1,11):
    print(num,"X",i,"=",(num*i))
```

Fun-fact

One can use an “else” clause with a “for” loop in Python.