enactus
University of Sheffield

CODE CREATORS
SHEFFIELD

2024 PYTHON BEGINNER COURSE

Lesson 3

SHALEV LAU
/in/shalevl
/mushroomhater07

Loops

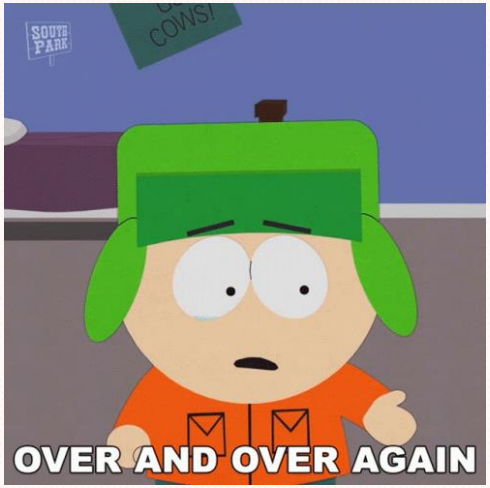
control statements that allow blocks of code to be repeated

If I want to print:

Hi
Hi
Hi
Hi
Hi
Hi

What I have to code

```
print('Hi')
print('Hi')
print('Hi')
print('Hi)
print('Hi')
print('Hi)
(...)
```



https://tenor.com/view/over-and-over-again-kyle-broflovski-south-park-probrem-gif-21171264

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Indefinite iteration: While Loop

Same as if condition:
Loops while the condition is true
Stops when the condition becomes false

Always check stopping condition before starting the loop

```
while count < 6:
    print("Hi")
    count += 1
```

https://tenor.com/view/too-many-counting-gif-15929632

enactus University of Sheffield

CODE CREATORS SHEFFIELD

While Loop :detailed

What: repeatedly execute block of code as long as the test expression (condition) is true.

When useful: We generally use this loop when we don't know the exact number of times to iterate.

```
count = 0
while count < 6:
    print("Hi")
    count += 1
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD

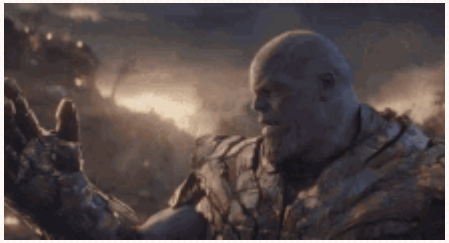
While Loop :detailed

What: repeatedly execute block of code as long as the test expression (condition) is true.

really ?

When useful: We generally use this loop when we don't know the exact number of times to iterate.

```
count = 0
while count < 6:
    print("Hi")
    count += 1
```



enactus University of Sheffield

CODE CREATORS SHEFFIELD

Until Loop :example

```
# Calculate the sum of numbers until user enters
0
number = int(input('Enter a number: '))

total = 0

# iterate until the user enters 0
until number == 0:
    total += number
    number = int(input('Enter a number: '))

print('The sum is', total)
```

Until
(NOT IN PYTHON)

Stop when condition is true

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Endless (infinite) Loop

1. No stopping condition

```
while true:
    print("Iteration:", count)
    #no stopping condition
```

2. Condition that never met + causing the loop to start over

```
while (count < 4):
     count = 0 # loop start over
     print("Iteration:", count)
     count +=1
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Do While (NOT IN PYTHON)

Run once before checking stopping condition

We want user to input a 4 digit passcode:

```
do {
    passcode = input("enter 4 digit passcode")}
while(len(passcode) < 4);
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Structure Programming

Aim to write programs which are easy to understand and work efficiently.

Dry-Run

Used to test an algorithm or program code without the use of a computer
Trace tables: a technique used to test algorithms to make sure that no logical errors occur

- are used by programmers to track the values of variables as they change throughout the program (see what the code will do before executing it)
- useful when a program is not producing the desired result (find where errors are in code)

n	n * n	total	n <> -1
0		0	True
2	4	4	True

enactus University of Sheffield
CODE CREATORS SHEFFIELD

To make a trace table

1. VARIABLES: note each variable in the piece of code you are looking at as a heading; (this includes arrays).

2. OUTPUTS: note if there is an output and put this as a heading

3. INPUTS: if there are inputs specified, put an inputs column and be prepared to fill it in.

4. **Operations/manipulations to input, i.e. any conditions, formulae or comparisons**

Make a Dry Run Table

```
x ← 4
y ← 3
Result ← 1

While y != 0
      Result ← Result * x
       y ← y – 1
```

x	y	y != 0	Result
4	3		1

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Your Turn

```
i = 0
j = 10
WHILE i < j
i = i + 2
j = j – 2
k = j + i
END WHILE
```

i	j	i < j	k

Definite iteration: For loop

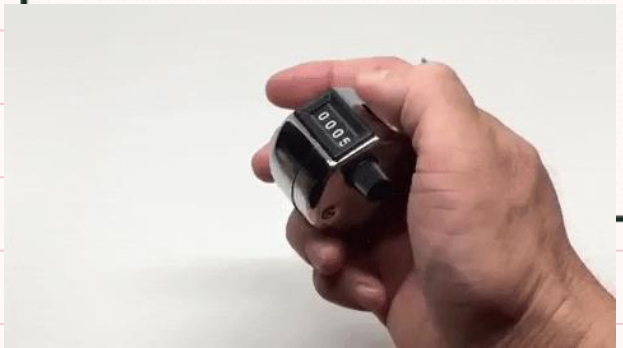
Known as fixed loop → number of iteration is predictable

Always has stepper variable → counting for every iteration started

"count 1 to ~~10~~ 9" →

1, 2, 3, 4, 5,6,7,8,9,~~10~~

```
for item in range(10):
    print(item)
```



https://giphy.com/gifs/harborneweb design-clicker-edcadmin-edc-admin-fVi8M1YPrfJgaqkJOA

enactus University of Sheffield

CODE CREATORS SHEFFIELD

For Loop :detailed

What: execute block of code set number of times.

When useful: We use this loop when we want to iterate over the sequence (range of numbers, string, list, etc.).

Definite iteration: For loop

```
for item in x:
    print(item)
```

When `x` is a list	e.g. `x = ['apple', 'orange', 'banana']`
When `x` is a string	e.g. `x = 'apple'`
When `x` is `range(end)`	e.g. `x = range(4)`
When `x` is `range(start, end)`	e.g. `x = range(2, 4)`
When `x` is `range(start, end, step)`	e.g. `x = range(0, 10, 3)`

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

When `x` is a list

e.g. `x = ['apple', 'orange', 'banana']`

```
for item in ['apple', 'orange', 'banana'] :
    print(item)
```

The list length is known before running the loop

return each element in order

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

```
for item in x:
    print(item)
```

When `x` is a list	e.g. `x = ['apple', 'orange', 'banana']`
When `x` is a string	e.g. `x = 'apple'`
When `x` is `range(end)`	e.g. `x = range(4)`
When `x` is `range(start, end)`	e.g. `x = range(2, 4)`
When `x` is `range(start, end, step)`	e.g. `x = range(0, 10, 3)`

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

When `x` is a string

e.g. `x = 'apple'`

```
for item in 'apple' :
    print(item)
```

When we iterate through "apple" (an example of string)
It will loop through each character
Result in a, p, p, l, e

Definite iteration: For loop

```
for item in x:
    print(item)
```

When `x` is a list	e.g. `x = ['apple', 'orange', 'banana']`
When `x` is a string	e.g. `x = 'apple'`
When `x` is `range(end)`	e.g. `x = range(4)`
When `x` is `range(start, end)`	e.g. `x = range(2, 4)`
When `x` is `range(start, end, step)`	e.g. `x = range(0, 10, 3)`

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

```
for item in range(4):
    print(item)
```

When `x` is `range(end)`

e.g. `x = range(4)`

Python count from 0, outputting 4 value
Resulting in 0,1,2,3

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

```
for item in x:
    print(item)
```

When `x` is a list	e.g. `x = ['apple', 'orange', 'banana']`
When `x` is a string	e.g. `x = 'apple'`
When `x` is `range(end)`	e.g. `x = range(4)`
When `x` is `range(start, end)`	e.g. `x = range(2, 4)`
When `x` is `range(start, end, step)`	e.g. `x = range(0, 10, 3)`

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

```
for item in range(2, 4):
    print(item)
```

When `x` is `range(start, end)`

e.g. `x = range(2, 4)`

Count from 2, not outputting the end index
Resulting in 2,3

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

```
for item in x:
    print(item)
```

When `x` is a list	e.g. `x = ['apple', 'orange', 'banana']`
When `x` is a string	e.g. `x = 'apple'`
When `x` is `range(end)`	e.g. `x = range(4)`
When `x` is `range(start, end)`	e.g. `x = range(2, 4)`
When `x` is `range(start, end, step)`	e.g. `x = range(0, 10, 3)`

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Definite iteration: For loop

Count from 0.
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
0, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1

why?

When `x` is `range(start, end, step)` e.g. `x = range(0, 10, 3)`

```
for item in range(1, 10, 3):
    print(item)
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD

Continue statement (while, for)

We use `continue` to skip the current iteration and immediately goes to the next iteration

```
for i in range(5):
    if i == 3:
        continue
    print(i)
```

```
i = 0
while i < 5:
    if i == 3:
        continue
    print(i)
```

Break +else statement (if)

We use `break` to terminate the loop immediately
If `break` is used, it will pass to `else` statement before the rest of the code

```
for i in range(5):
    if i == 3:
        break
    print(i)
print("loop completed")
```

```
i = 0
while i < 5:
    if i == 3:
        break
    print(i)
else:
    print("loop terminated by break")
print("loop completed")
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD

In class exercise

Ask the user to enter 10 numbers. At the end, tell them the mean average, **maximum and minimum of the values entered.**

Project: Number guesser

Write a program that generates a random number between 1 and 100. Allow the user to keep guessing the number until they have guessed the random number.

Challenge: Extend/modify the above program to include the following

a) If they enter a number lower than the number generated, tell them that their number is too low. Do the same if they enter a number higher than the number generated.

b) tell them how many guesses they entered before each guess

c) set the (generated number +1) as landmine, if user entered that number, they lose.

```
import random

randomNumber = random.randrange(1, 50) + 1
# this code generate number from 1 to 50
```

enactus University of Sheffield

CODE CREATORS SHEFFIELD