

# **Lesson 4 – Data structures:**

---

## Traversal and search

# Traversal and Search

---

In the case of ordered data structures, such as strings, tuples, and lists, there are two algorithms that are often used:

## **Traversal** and **Search:**

- Traversal: when it must read all elements in the sequence, without exception
- Search: we can stop when we find the element within the sequence. No need to read all elements in the sequence

# Traversal and Search

---

A program implements a **Traversal** when it must read all the elements in a sequence, without exception, to produce the result.

Example: How many vowels are there?

H	e	l	l	o		S	a	m	
0	1	2	3	4	5	6	7	8	9

Example: How many even numbers are there?

```
sequence =[45, 92, 19, 81, 50, 65, 5, 50, 100, 70, 83, 68, 51, 11, 7]
```

# Traversal

---

To go through a data structure from the beginning to the end, we need:

- Variable acting as counter – the index (e.g. `i`)
- To know the length (`len()`)
- Iterative structure `for`

```
for i in range(len(<data structure>)):  
    <actions>
```

Of course, there may be other ways to traverse a data structure, depending on the purpose of the program

# *“Pythonic” traversal*

---

To go through a data structure from the beginning to the end, we need:

- Variable to store the element
- Iterative structure `for`

```
for element in < data structure >:  
    <actions>
```

## Traversal in strings - *Example: Count even numbers*

---

Example: How many even numbers are in a list `l` of nums?

- Variable acting as counter – the index (e.g. `i`)
- Know the length (`len()`)
- Iterative structure `for`

# Traversal in strings - *Example: Count even numbers*

---

Example: How many even numbers are in a list `l` of nums?

- Variable acting as counter – the index (e.g. `i`)
- Know the length (`len()`)
- Iterative structure `for`

```
l =[45,92,19,81,50,65,5,50,100,70,83,68,51,11,7]

numEven = 0
for i in range(len(l)):
    if l[i]%2 == 0:
        numEven +=1

print("List contains " + str(numEven) + " even numbers.")
```

## *“Pythonic” traversal – Example: Count even numbers*

---

To go through a data structure from the beginning to the end we need:

- Variable to store the element
- Iterative structure `for`

```
for element in < data structure >:  
    <actions>
```

```
l = [45, 92, 19, 81, 50, 65, 5, 50, 100, 70, 83, 68, 51, 11, 7]
```

```
numEven = 0
```

```
for element in l:
```

```
    if element%2 == 0:
```

```
        numEven +=1
```

```
print("List contains " + str(numEven) + " even numbers.")
```



# Traversal in strings - *Example: Count vowels*

---

## Example:

Make a program that asks the user to enter a string via keyboard, and counts how many vowels are in the string. The program will display the final result with the message: ‘The string "XXXXXX" has NN vowels.’ where "XXXXXX" is the input string and NN is the number of vowels.

Note: Do NOT count accentuated vowels (only ‘*aeiouAEIOU*’).

## Reminder:

- Variable acting as counter – the index (e.g. `i`)
- Know the length (`len()`)
- Iterative structure `for`

# Traversal in strings - *Example: Count vowels*

---

## Option 1:

```
s = input("Write a string:")

numVowels = 0

for i in range(len(s)):
    if (s[i] in "aeiouAEIOU"):
        numVowels += 1

print("The string \"" + s + "\" has " + str(numVowels) + " vowels.")
```

**\** is used to print "

# Traversal in strings - *Example: Count vowels*

---

## Option 1:

```
s = input("Write a string:")

numVowels = 0

for i in range(len(s)):
    if (s[i] in "aeiouAEIOU"):
        numVowels += 1

print("The string \"" + s + "\" has " + str(numVowels) + " vowels.")
```

**"\"** is used to print "

## Option 2:

```
s = input("Write a string:")
numVowels= 0
for i in s:
    if (i in "aeiouAEIOU"):
        numVowels+= 1

print("The string \"" + s + "\" has " + str(numVowels) + " vowels." )
```

# Traversal vs Search

---

A program implements a **Search** when the result is “yes” or “no”, and in most cases, this result can be often given with absolute certainty without reading all the elements of the sequence.

Example: Is there any blank space?

h	e	l	l	o		T	o	m	
0	1	2	3	4	5	6	7	8	9

Example: Is there any even number?

```
sequence =[45, 92, 19, 81, 50, 65, 5, 50, 100, 70, 83, 68, 51, 11, 7]
```

It is important to detect when a program does **not need to process all the elements of a sequence** to avoid useless computation time, as the sequences can be very long.

# Search

---

To search within a data structure, we need:

- Variable acting as counter – the index (e.g. `i`)
- Variable to know if the desired element has been found (e.g. `Found`)
- To know the length (`len()`)
- Iterative structure `while`

```
found = False
i=0

while ( i < (len(<data structure>)) and not found ):

    ...

    if ( <desired property> ):
        found = True
    else:
        i += 1      # move to the next element

if ( found ):
    <instructions found>
else:
    <instructions NOT found>
```

# Search in lists – Any even number?

---

## Example:

Make a program that asks the user to enter a series of integers and store them in a **list**. When a 0 is entered, it means that the sequence has finished.

Then check for any even numbers. If so, the program should display the message "There is an even number". Otherwise, the message "No even number".

## Reminder:

- Variable acting as counter – the index (e.g. `i`)
- Variable to know if the desired element has been found (e.g. `Found`)
- To know the length (`len()`)
- Iterative structure `while`

# Search in lists – Any even number?

---

Example: We store the elements in the list `l`, and then we check if there is any even number

```
l=[]
num = int(input("Enter a num:"))
while(num!=0):
    l.append(num)
    num = int(input("Enter a num:"))

# for example, we get l=[45,92,19,81,50,65,5,50,100,70,83,68,51,11,7]
```

# Search in lists – Any even number?

---

Example: We store the elements in the list `l`, and then we check if there is any even number

```
l=[]
num = int(input("Enter a num:"))
while(num!=0):
    l.append(num)
    num = int(input("Enter a num:"))

# for example, we get l=[45,92,19,81,50,65,5,50,100,70,83,68,51,11,7]

found = False
i = 0

while (i<len(l) and not found):
    if (l[i]%2 == 0):
        found = True
    else:
        i += 1

if (found):
    print("There is at least one even number")
else:
    print("There is NO even number")
```



# Search in strings – Any blank space?

---

Example: Is there any blank space?

- Variable acting as counter – the index (e.g. `i`)
- Variable to know if the desired element has been found (e.g. `Found`)
- To know the length (`len()`)
- Iterative structure `while`

# Search in strings – Any blank space?

---

## Example: Is there any blank space?

- Variable acting as counter – the index (e.g. `i`)
- Variable to know if the desired element has been found (e.g. `Found`)
- To know the length (`len()`)
- Iterative structure `while`

```
s = input("Write a string: ")

Found = False

i = 0

while (i < len(s) and not Found):
    if (s[i] == " "):
        Found = True
    else:
        i += 1

if (Found):
    print("There is one or more blank spaces")
else:
    print("There is NO blank")
```