

Processing Lists and List Comprehension

Programming and Algorithms

Lecture by
Dr Daniil Osudin

```
n = 3  
for i in range(1,n+1):  
    print("Hello World!")
```

Hello World!
Hello World!
Hello World!

What will we Cover?

- Processing lists
- List comprehension
- Writing simple programs incorporating the list data structure

Processing Lists I

Use a FOR loop to print all of the elements of a list and their indices

```
birds = ["sparrow", "dove", "swan", "duck"]  
for i in range(len(birds)):  
    print("index of", birds[i], "in birds is", i)
```

```
index of sparrow in birds is 0  
index of dove in birds is 1  
index of swan in birds is 2  
index of duck in birds is 3
```

range() function is using the default values for start and step and length of the birds list for stop

Processing Lists II

Alternatively enumerate() function can be used with the FOR loop to print all of the elements of a list and their indices

```
birds = ["sparrow", "dove", "swan", "duck"]  
for i, b in enumerate(birds):  
    print("index of", b, "in birds is", i)
```

```
index of sparrow in birds is 0  
index of dove in birds is 1  
index of swan in birds is 2  
index of duck in birds is 3
```

enumerate() function is using two different variables, in this case i and b, to keep the index and the value of the element at that index

Processing Lists III

Use the WHILE loop to remove any elements with a value "a" from the list

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
while "a" in word: ←  
    word.remove("a")  
print("word after removing letter a is:", word)
```

Condition is using the
in operator

```
word after removing letter a is: ['d', 'm', 'n', 't']
```

List Comprehension

Used to create a new list based on the values of the elements in an existing list

```
list = [...]  
new_list = [<expression> for <variable> in list if <condition>]
```

- Go through the elements one by one
- Select the elements, which satisfy the condition
- Modify them using the expression and append to the new list

List Comprehension Example

`i != "a"` is False

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

```
word after removing letter a is: ['d', 'm', 'n', 't']
```

```
i = "a"  
letters = []
```


List Comprehension Example

`i != "a" is True`

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

```
word after removing letter a is: ['d', 'm', 'n', 't']
```

```
i = "d"  
letters = ["d"]
```

List Comprehension Example

`i != "a" is False`

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

word after removing letter a is: ['d', 'm', 'n', 't']

```
i = "a"  
letters = ["d"]
```

List Comprehension Example

`i != "a" is True`

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

word after removing letter a is: ['d', 'm', 'n', 't']

```
i = "m"  
letters = ["d", "m"]
```

List Comprehension Example

`i != "a"` is False

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

```
word after removing letter a is: ['d', 'm', 'n', 't']
```

```
        i = "a"  
letters = ["d", "m"]
```

List Comprehension Example

`i != "a" is True`

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

word after removing letter a is: ['d', 'm', 'n', 't']

```
i = "n"  
letters = ["d", "m", "n"]
```

List Comprehension Example

`i != "a" is True`

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

word after removing letter a is: ['d', 'm', 'n', 't']

```
i = "t"  
letters = ["d", "m", "n", "t"]
```

List Comprehension Example

```
word = ["a", "d", "a", "m", "a", "n", "t"]  
letters = [i for i in word if i != "a"]  
print("word after removing letter a is:", letters)
```

```
word after removing letter a is: ['d', 'm', 'n', 't']
```

print the answer

More List Comprehension Examples

```
letters = ["b", 2, "a", "e", "a", "b", 7, "c", 9]
b_count = [1 for i in letters if i == "b"]
print(sum(b_count))
```

2

If the element is a "b", append a 1,
otherwise skip
Then take the sum to find the number of
occurrences of "b" in the list

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_squares = [i*i for i in numbers if i%2 == 0]
print(even_squares)
```

[4, 16, 36, 64, 100]

If the element is a even, append it
multiplied by itself, otherwise skip

Try It Yourself

Problem 1

Write a program in Python environment that takes an input of 5 numbers and calculates their average

Hint: Use an `input()` function within a loop that executes 5 times.

Calculate the average by using `sum()` and `len()` functions

Try It Yourself

Problem 2

Write a program in Python environment that uses list comprehension, which turns

```
[4, 5, 1, 2, 7, 9, 6, 10]
```

Into

```
[10, 2, 14, 18]
```