CP020001 Computer Programming

Lecture: List/Array

https://github.com/kaopanboonyuen/CP020001_ComputerProgramming_2023s1

Contact: teerapong.pa@chula.ac.th

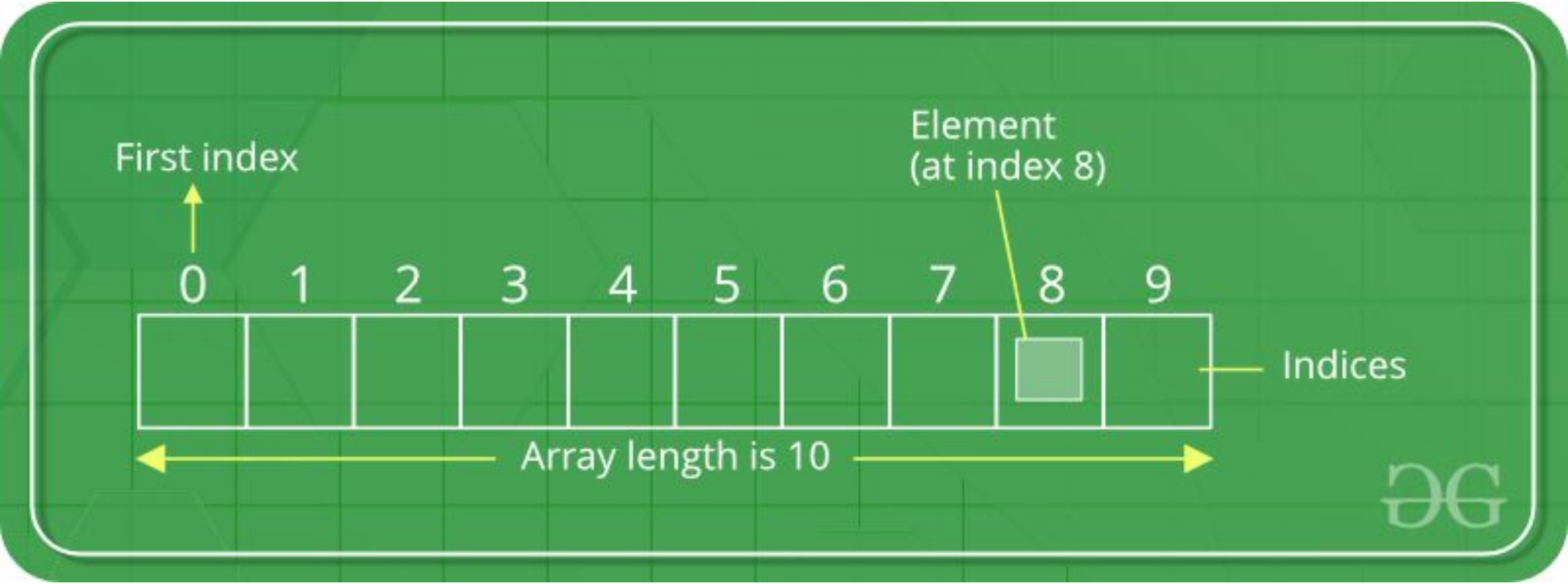
Reference:

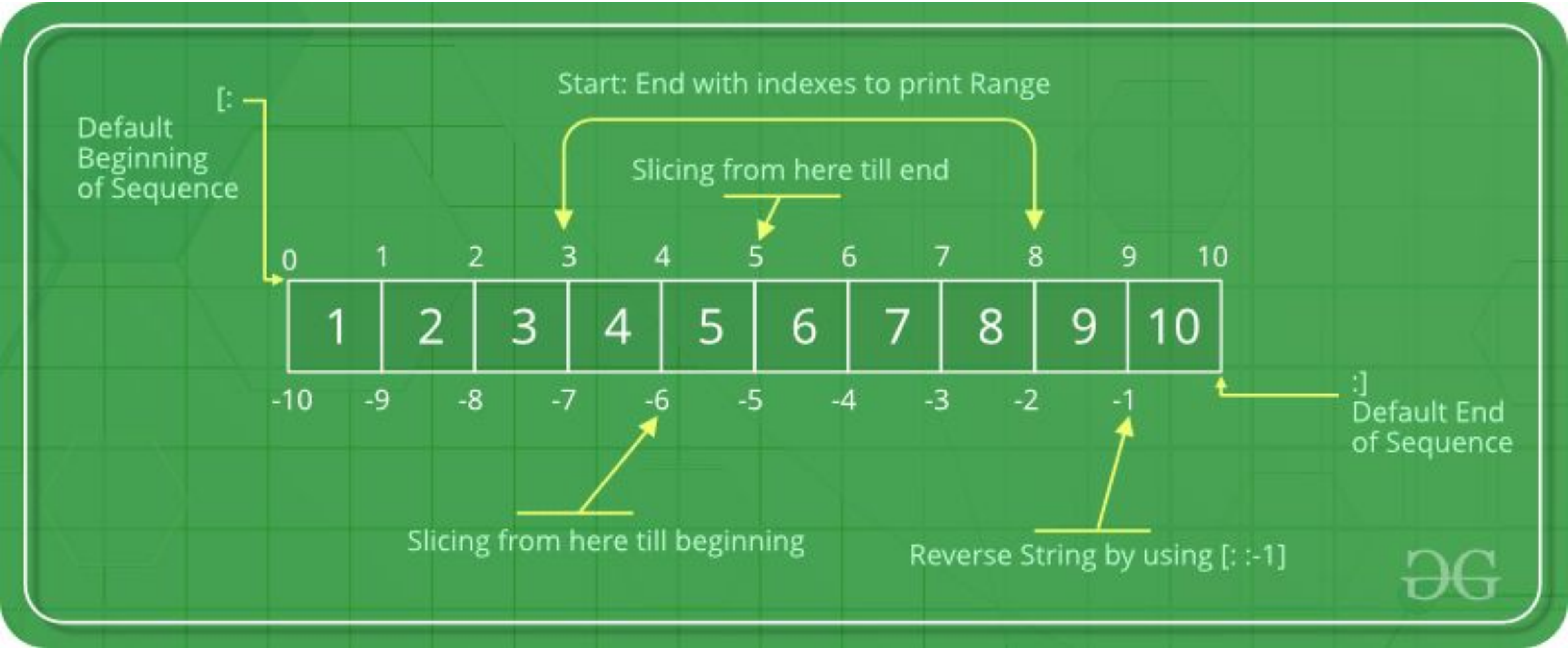
- https://www.scaler.com/topics/list-methods-in-python/
- https://www.w3schools.com/python/python_lists.asp
- https://www.educba.com/arrays-in-python/
- https://www.askpython.com/python/array/reverse-an-array-in-python
- https://www.quora.com/What-is-the-difference-between-lists-and-arrays-in-Python
-

Python Collections (Arrays)

There are four collection data types in the Python programming language:

- **List** is a collection which is ordered and changeable. Allows duplicate members.
- **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.
- **Set** is a collection which is unordered, unchangeable*, and unindexed. No duplicate members.
- **Dictionary** is a collection which is ordered** and changeable. No duplicate members.


First index
Element
(at index 8)
0
1
2
3
4
5
6
7
8
9
Indices
Array length is 10



Start: End with indexes to print Range

[:

Default Beginning of Sequence

Slicing from here till end

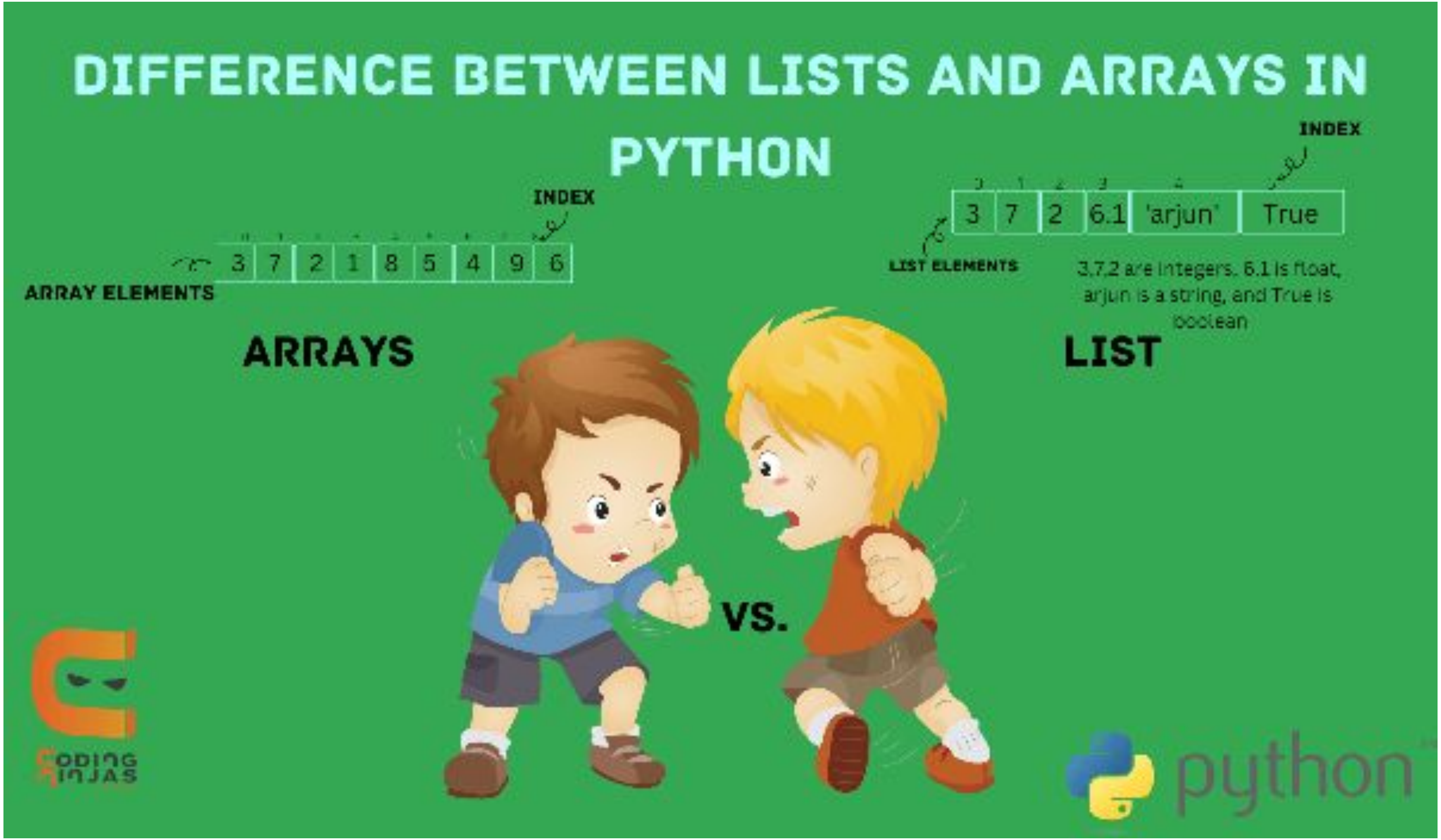
:]

Default End of Sequence

Slicing from here till beginning

Reverse String by using [: :-1]

DIFFERENCE BETWEEN LISTS AND ARRAYS IN PYTHON



INDEX

3	7	2	1	8	5	4	9	6

ARRAY ELEMENTS

ARRAYS

INDEX

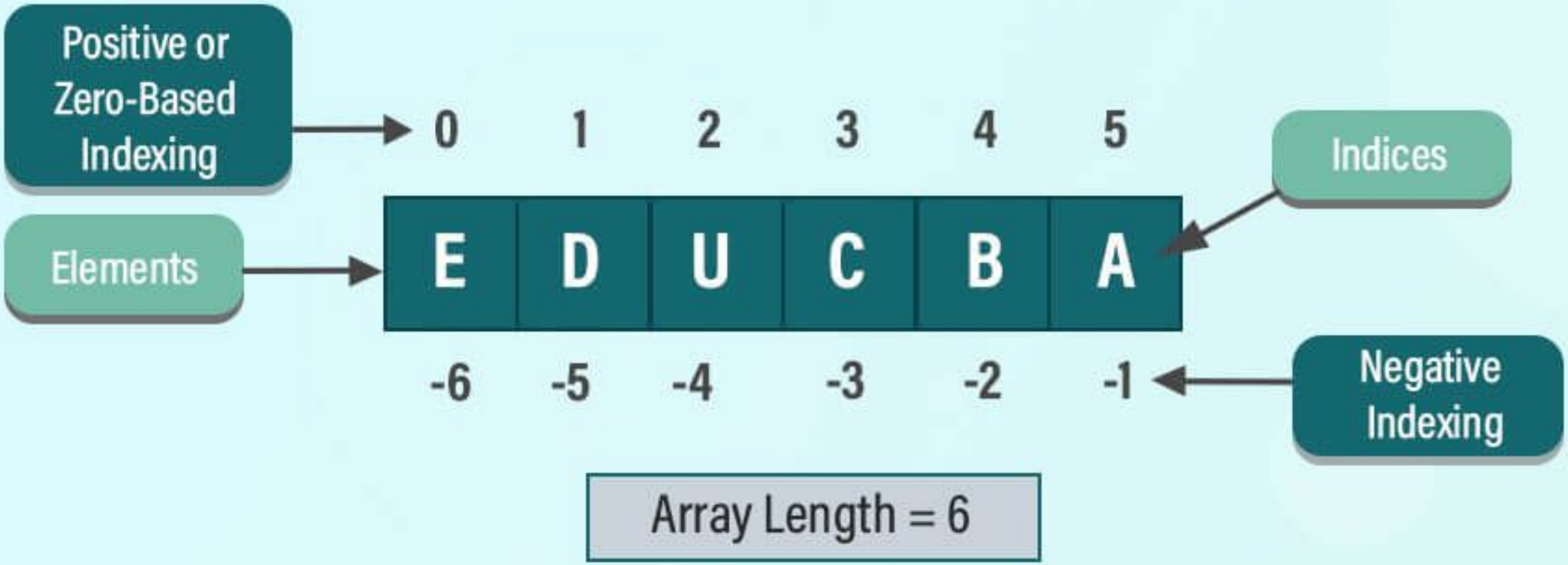
0	1	2	3	4	
3	7	2	6.1	'arjun'	True

LIST ELEMENTS

3,7,2 are integers. 6.1 is float, arjun is a string, and True is boolean

LIST

Arrays in Python



Positive or Zero-Based Indexing

Indices

Elements

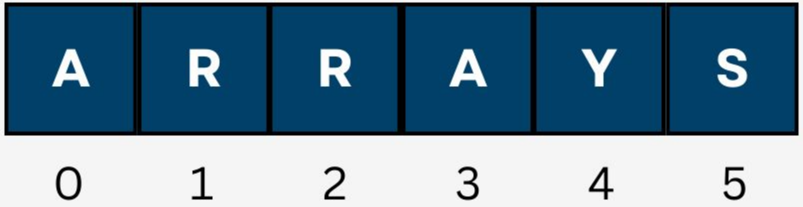
Negative Indexing

0	1	2	3	4	5
E	D	U	C	B	A
-6	-5	-4	-3	-2	-1

Array Length = 6

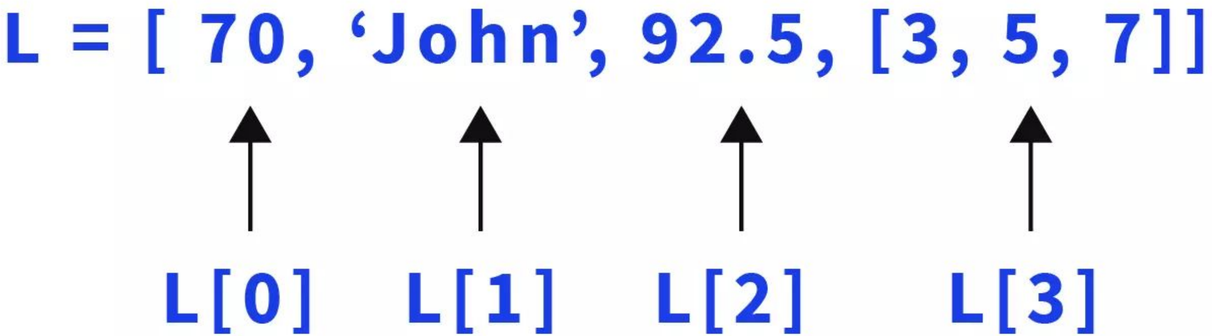
EDUCBA

Arrays in Python



A	R	R	A	Y	S
0	1	2	3	4	5

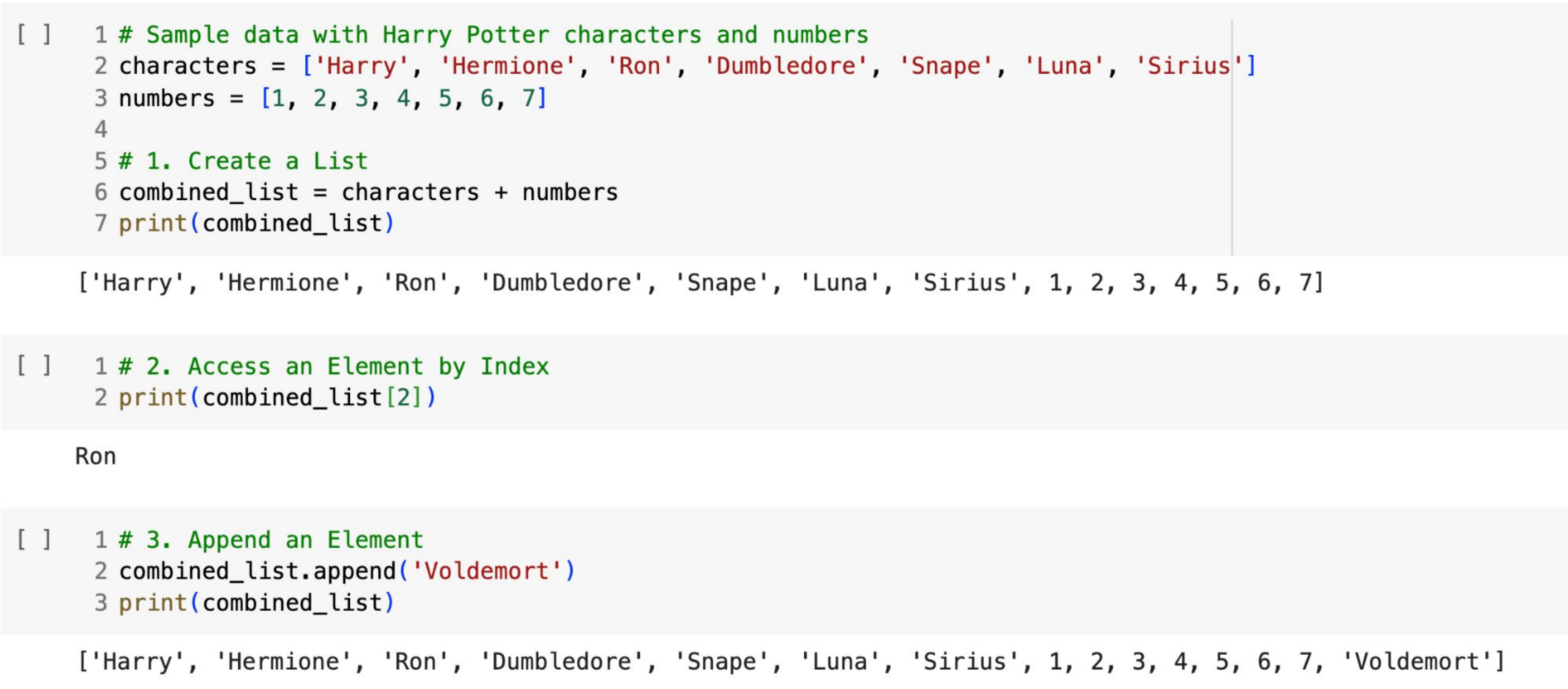
List in Python



L = [70, ‘John’, 92.5, [3, 5, 7]]

L[0] L[1] L[2] L[3]

SCALER Topics



```python
# Sample data with Harry Potter characters and numbers
characters = ['Harry', 'Hermione', 'Ron', 'Dumbledore', 'Snape', 'Luna', 'Sirius']
numbers = [1, 2, 3, 4, 5, 6, 7]

# 1. Create a List
combined_list = characters + numbers
print(combined_list)
```

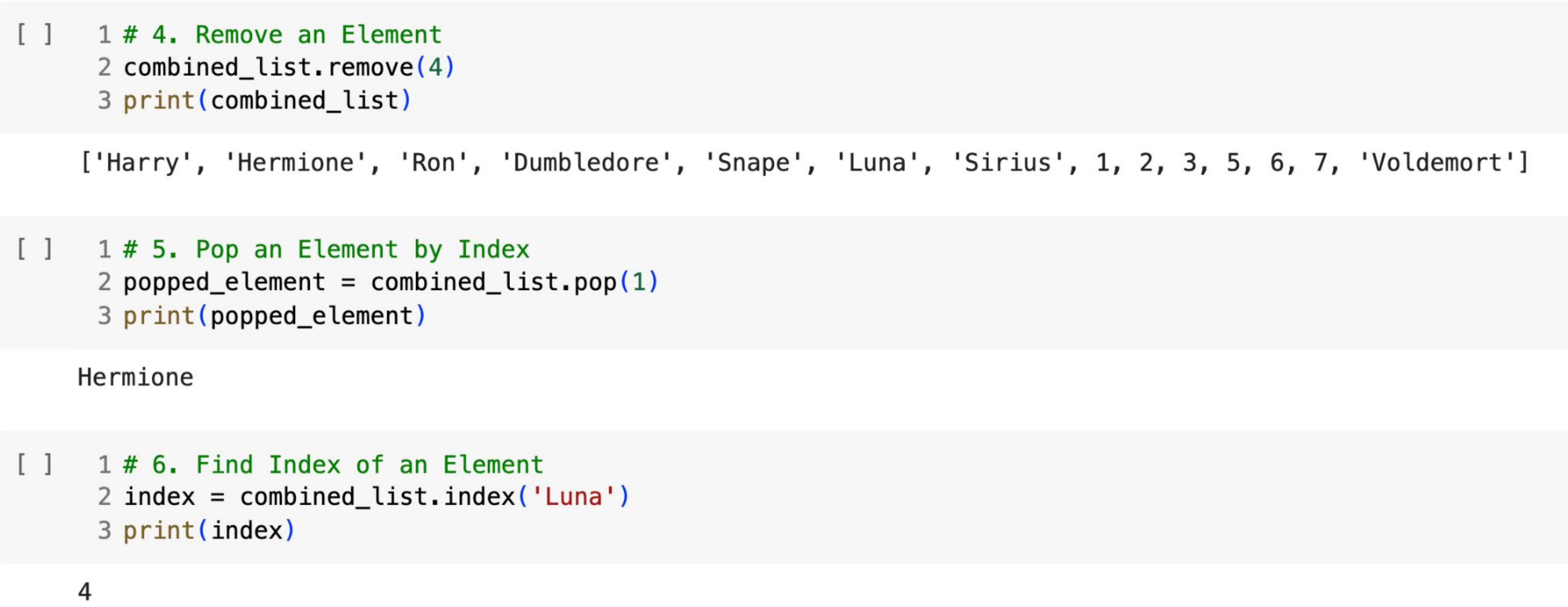
```
['Harry', 'Hermione', 'Ron', 'Dumbledore', 'Snape', 'Luna', 'Sirius', 1, 2, 3, 4, 5, 6, 7]
```

```python
# 2. Access an Element by Index
print(combined_list[2])
```

```
Ron
```

```python
# 3. Append an Element
combined_list.append('Voldemort')
print(combined_list)
```

```
['Harry', 'Hermione', 'Ron', 'Dumbledore', 'Snape', 'Luna', 'Sirius', 1, 2, 3, 4, 5, 6, 7, 'Voldemort']
```



```python
# 4. Remove an Element
combined_list.remove(4)
print(combined_list)
```

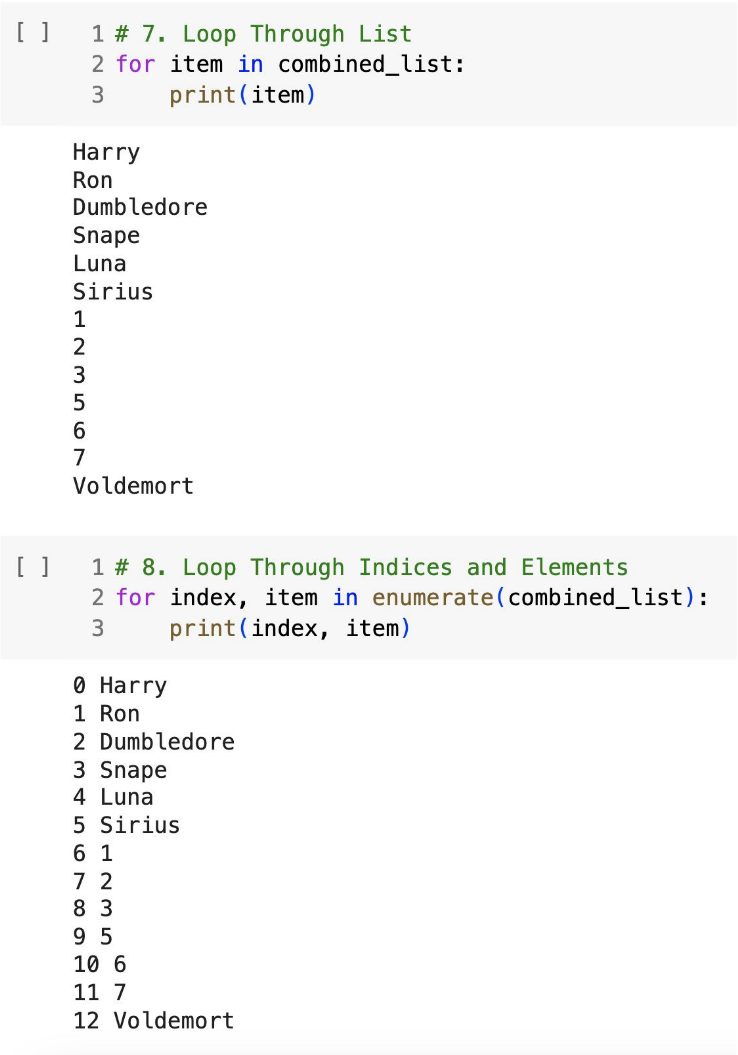
```
['Harry', 'Hermione', 'Ron', 'Dumbledore', 'Snape', 'Luna', 'Sirius', 1, 2, 3, 5, 6, 7, 'Voldemort']
```

```python
# 5. Pop an Element by Index
popped_element = combined_list.pop(1)
print(popped_element)
```

```
Hermione
```

```python
# 6. Find Index of an Element
index = combined_list.index('Luna')
print(index)
```

```
4
```

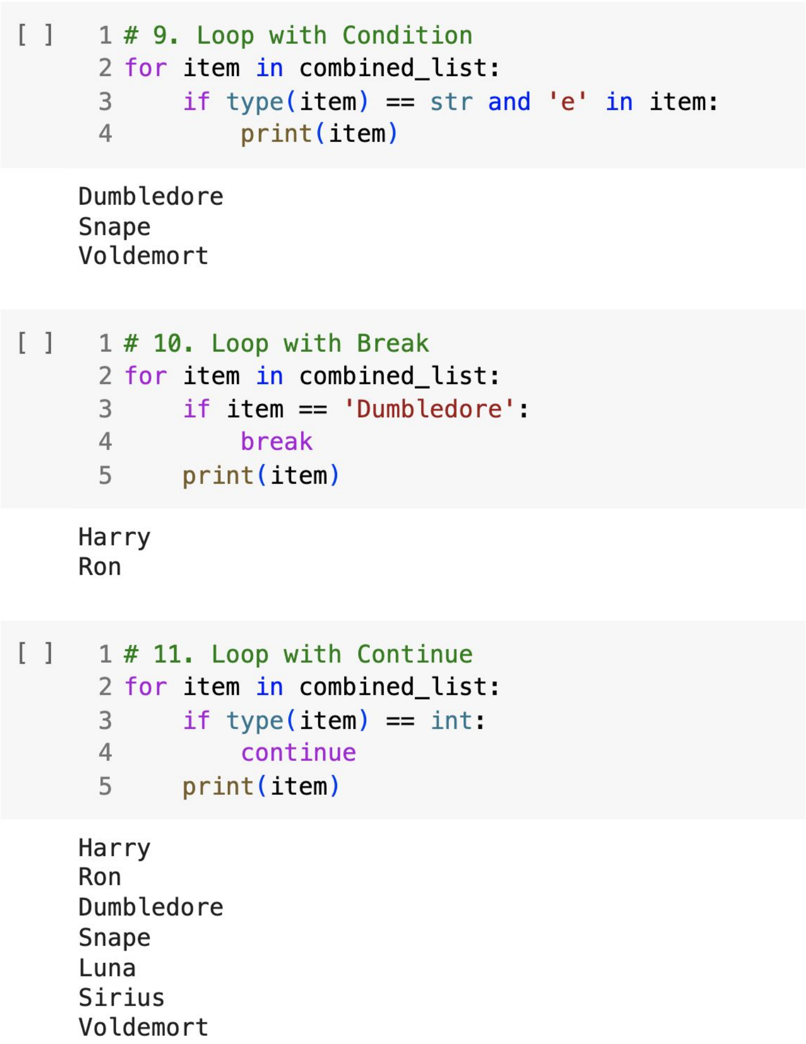


```python
# 7. Loop Through List
for item in combined_list:
    print(item)
```

```
Harry
Ron
Dumbledore
Snape
Luna
Sirius
1
2
3
5
6
7
Voldemort
```

```python
# 8. Loop Through Indices and Elements
for index, item in enumerate(combined_list):
    print(index, item)
```

```
0 Harry
1 Ron
2 Dumbledore
3 Snape
4 Luna
5 Sirius
6 1
7 2
8 3
9 5
10 6
11 7
12 Voldemort
```



```python
# 9. Loop with Condition
for item in combined_list:
    if type(item) == str and 'e' in item:
        print(item)
```

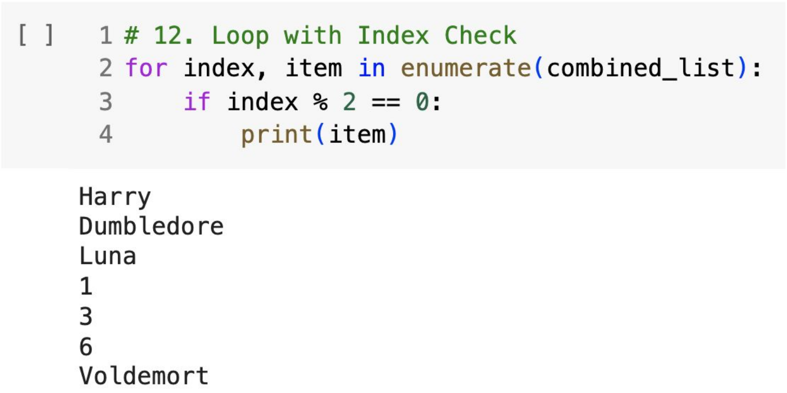
```
Dumbledore
Snape
Voldemort
```

```python
# 10. Loop with Break
for item in combined_list:
    if item == 'Dumbledore':
        break
    print(item)
```

```
Harry
Ron
```

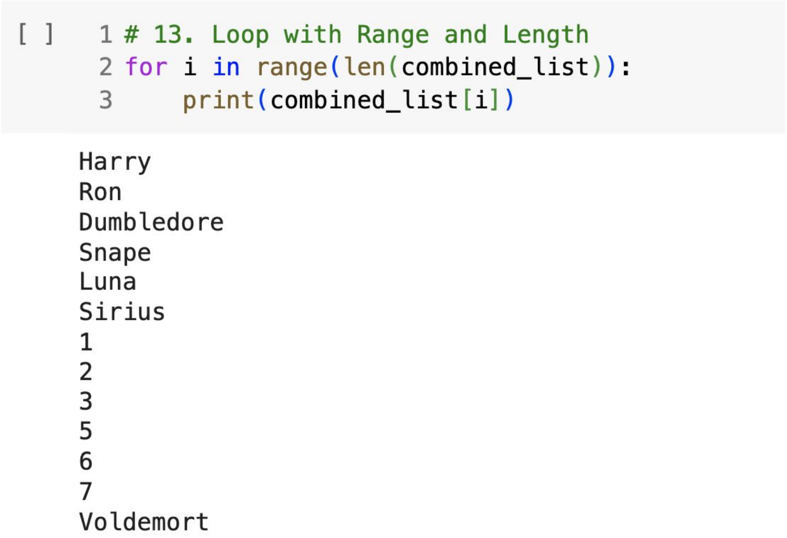
```python
# 11. Loop with Continue
for item in combined_list:
    if type(item) == int:
        continue
    print(item)
```

```
Harry
Ron
Dumbledore
Snape
Luna
Sirius
Voldemort
```



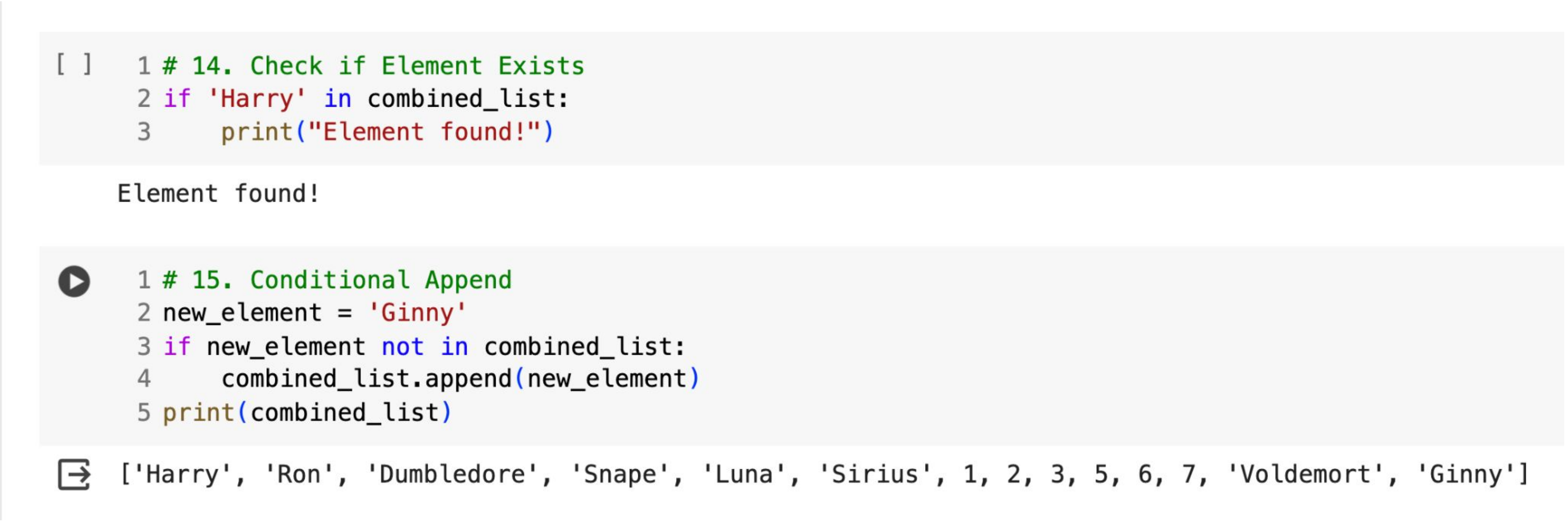
```python
# 12. Loop with Index Check
for index, item in enumerate(combined_list):
    if index % 2 == 0:
        print(item)
```

```
Harry
Dumbledore
Luna
1
3
6
Voldemort
```



```python
# 13. Loop with Range and Length
for i in range(len(combined_list)):
    print(combined_list[i])
```

```
Harry
Ron
Dumbledore
Snape
Luna
Sirius
1
2
3
5
6
7
Voldemort
```

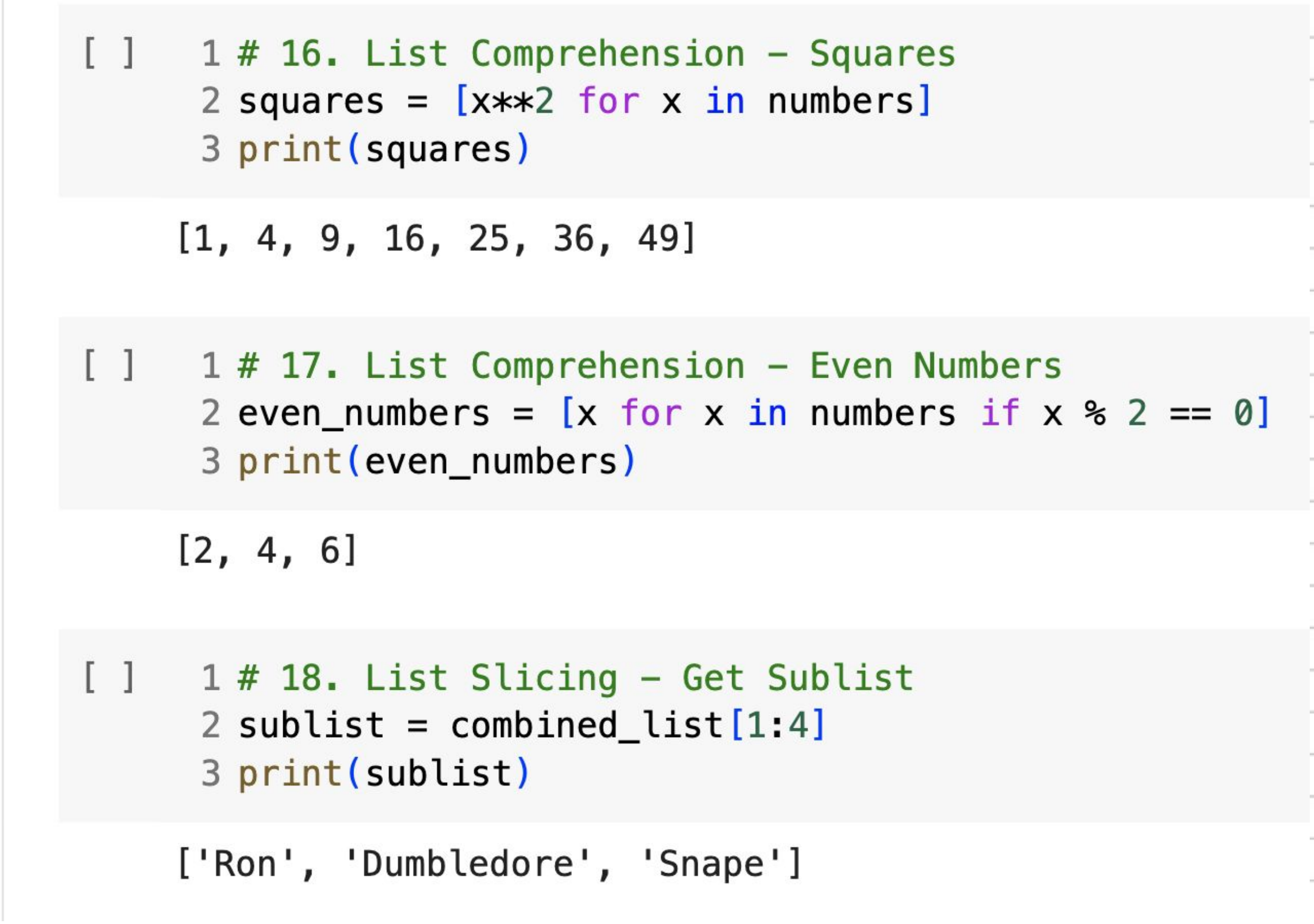


```python
# 14. Check if Element Exists
if 'Harry' in combined_list:
    print("Element found!")
```

```
Element found!
```

```python
# 15. Conditional Append
new_element = 'Ginny'
if new_element not in combined_list:
    combined_list.append(new_element)
print(combined_list)
```

```
['Harry', 'Ron', 'Dumbledore', 'Snape', 'Luna', 'Sirius', 1, 2, 3, 5, 6, 7, 'Voldemort', 'Ginny']
```



```python
# 16. List Comprehension - Squares
squares = [x**2 for x in numbers]
print(squares)
```

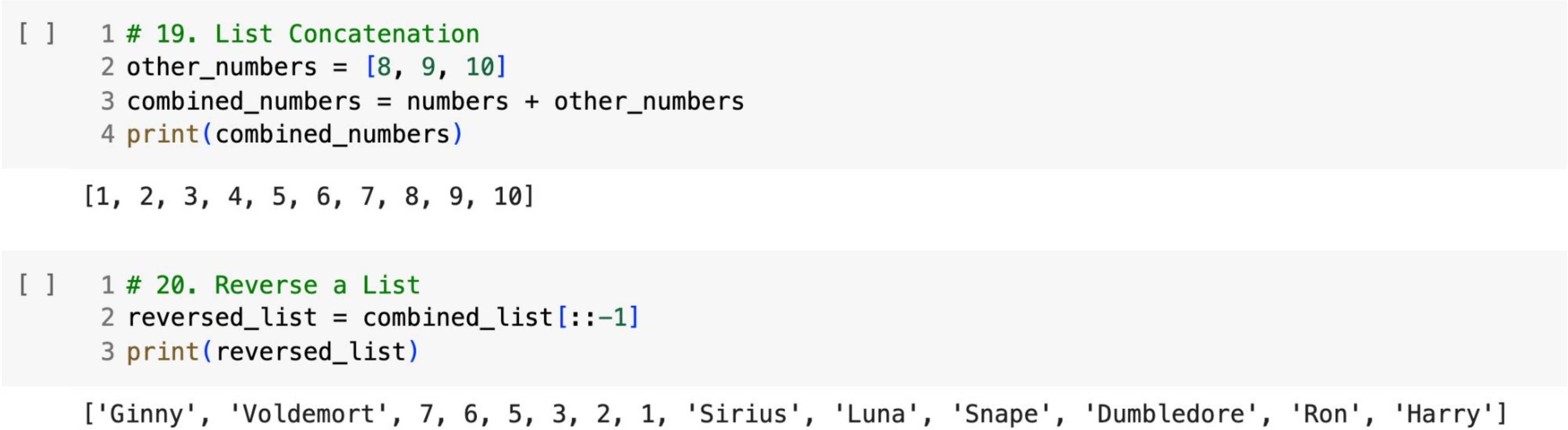
```
[1, 4, 9, 16, 25, 36, 49]
```

```python
# 17. List Comprehension - Even Numbers
even_numbers = [x for x in numbers if x % 2 == 0]
print(even_numbers)
```

```
[2, 4, 6]
```

```python
# 18. List Slicing - Get Sublist
sublist = combined_list[1:4]
print(sublist)
```

```
['Ron', 'Dumbledore', 'Snape']
```



```
[ ]
# 19. List Concatenation
other_numbers = [8, 9, 10]
combined_numbers = numbers + other_numbers
print(combined_numbers)
```

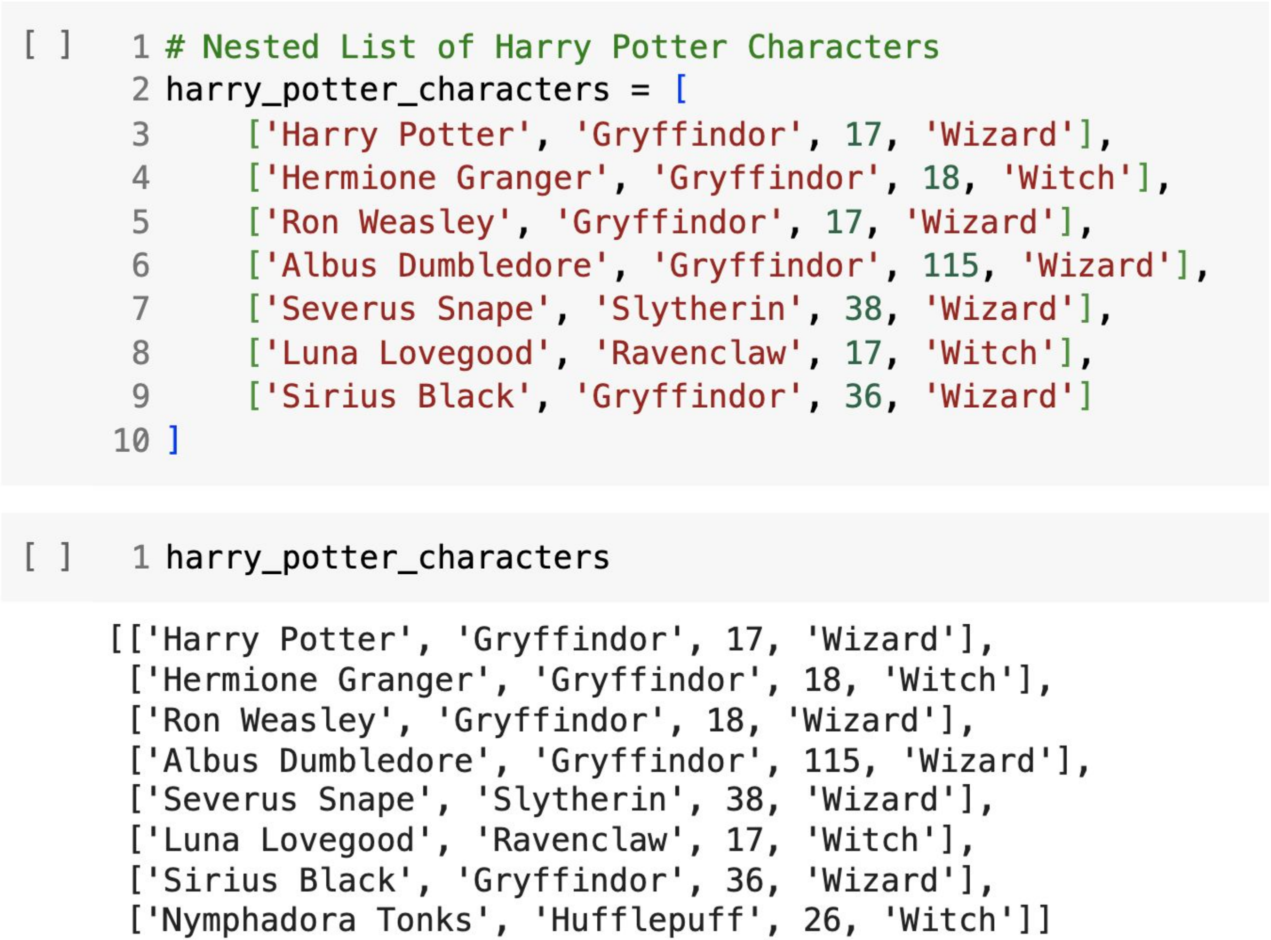
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
[ ]
# 20. Reverse a List
reversed_list = combined_list[::-1]
print(reversed_list)
```

```
['Ginny', 'Voldemort', 7, 6, 5, 3, 2, 1, 'Sirius', 'Luna', 'Snape', 'Dumbledore', 'Ron', 'Harry']
```

Nested List with Harry Potter Data

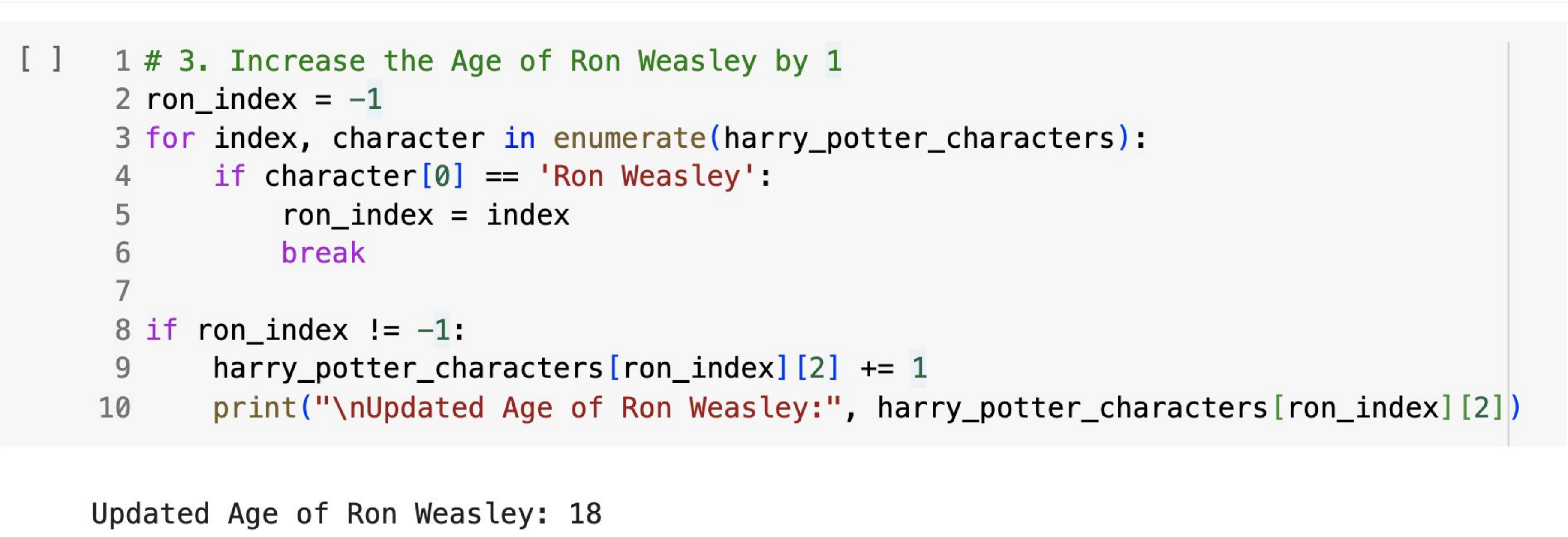
"STUDENT SAFETY IS MY TOP PRIORITY"
REPEATEDLY HIRES UNSTABLE AND BAD PROFESSORS



```python
# Nested List of Harry Potter Characters
harry_potter_characters = [
    ['Harry Potter', 'Gryffindor', 17, 'Wizard'],
    ['Hermione Granger', 'Gryffindor', 18, 'Witch'],
    ['Ron Weasley', 'Gryffindor', 17, 'Wizard'],
    ['Albus Dumbledore', 'Gryffindor', 115, 'Wizard'],
    ['Severus Snape', 'Slytherin', 38, 'Wizard'],
    ['Luna Lovegood', 'Ravenclaw', 17, 'Witch'],
    ['Sirius Black', 'Gryffindor', 36, 'Wizard']
]
```

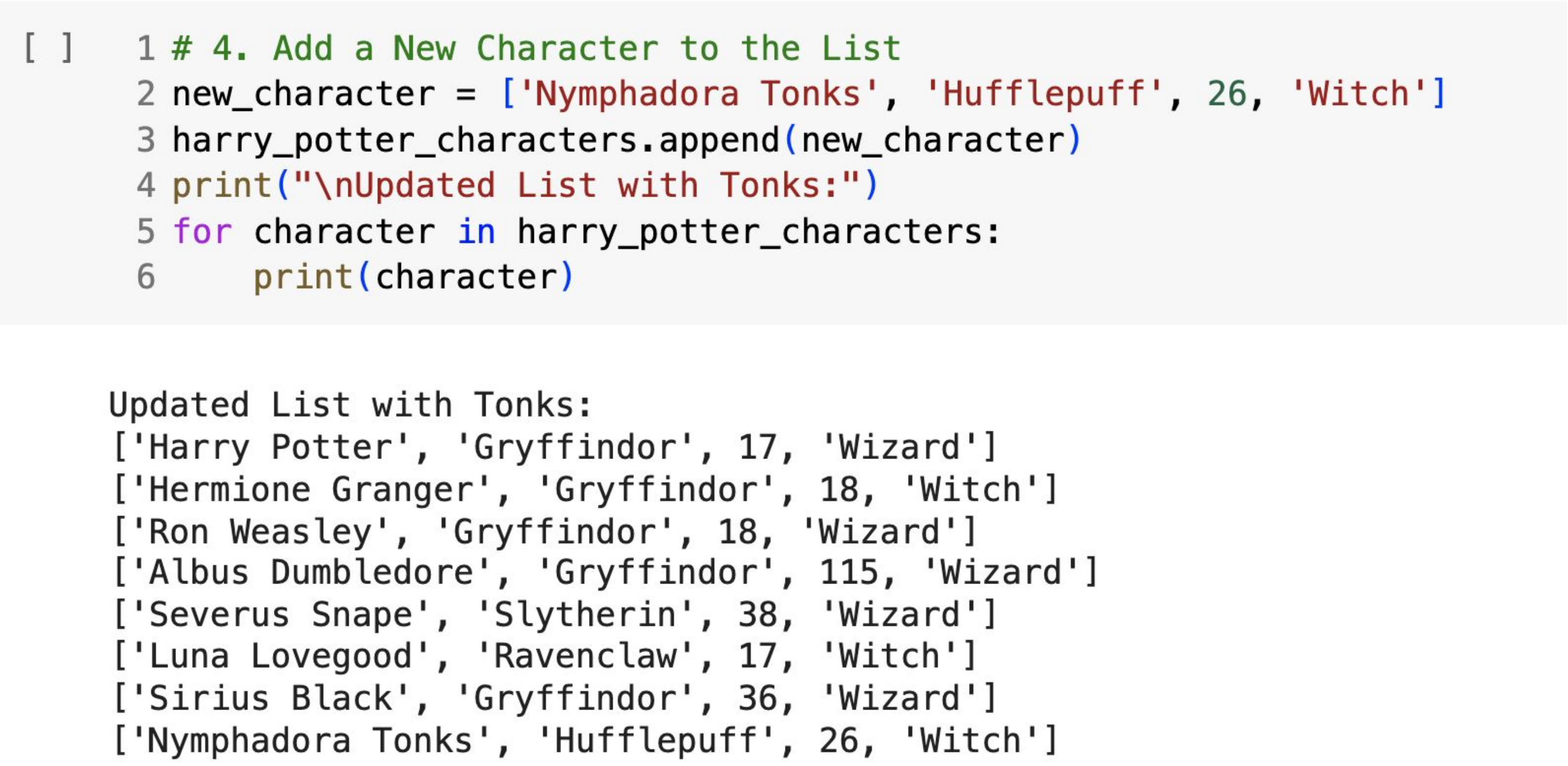
```python
harry_potter_characters
```

```
[['Harry Potter', 'Gryffindor', 17, 'Wizard'],
 ['Hermione Granger', 'Gryffindor', 18, 'Witch'],
 ['Ron Weasley', 'Gryffindor', 18, 'Wizard'],
 ['Albus Dumbledore', 'Gryffindor', 115, 'Wizard'],
 ['Severus Snape', 'Slytherin', 38, 'Wizard'],
 ['Luna Lovegood', 'Ravenclaw', 17, 'Witch'],
 ['Sirius Black', 'Gryffindor', 36, 'Wizard'],
 ['Nymphadora Tonks', 'Hufflepuff', 26, 'Witch']]
```



```python
# 3. Increase the Age of Ron Weasley by 1
ron_index = -1
for index, character in enumerate(harry_potter_characters):
    if character[0] == 'Ron Weasley':
        ron_index = index
        break

if ron_index != -1:
    harry_potter_characters[ron_index][2] += 1
    print("\nUpdated Age of Ron Weasley:", harry_potter_characters[ron_index][2])
```

```
Updated Age of Ron Weasley: 18
```

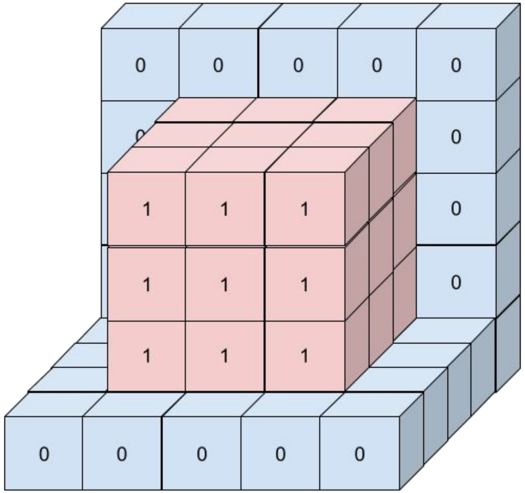


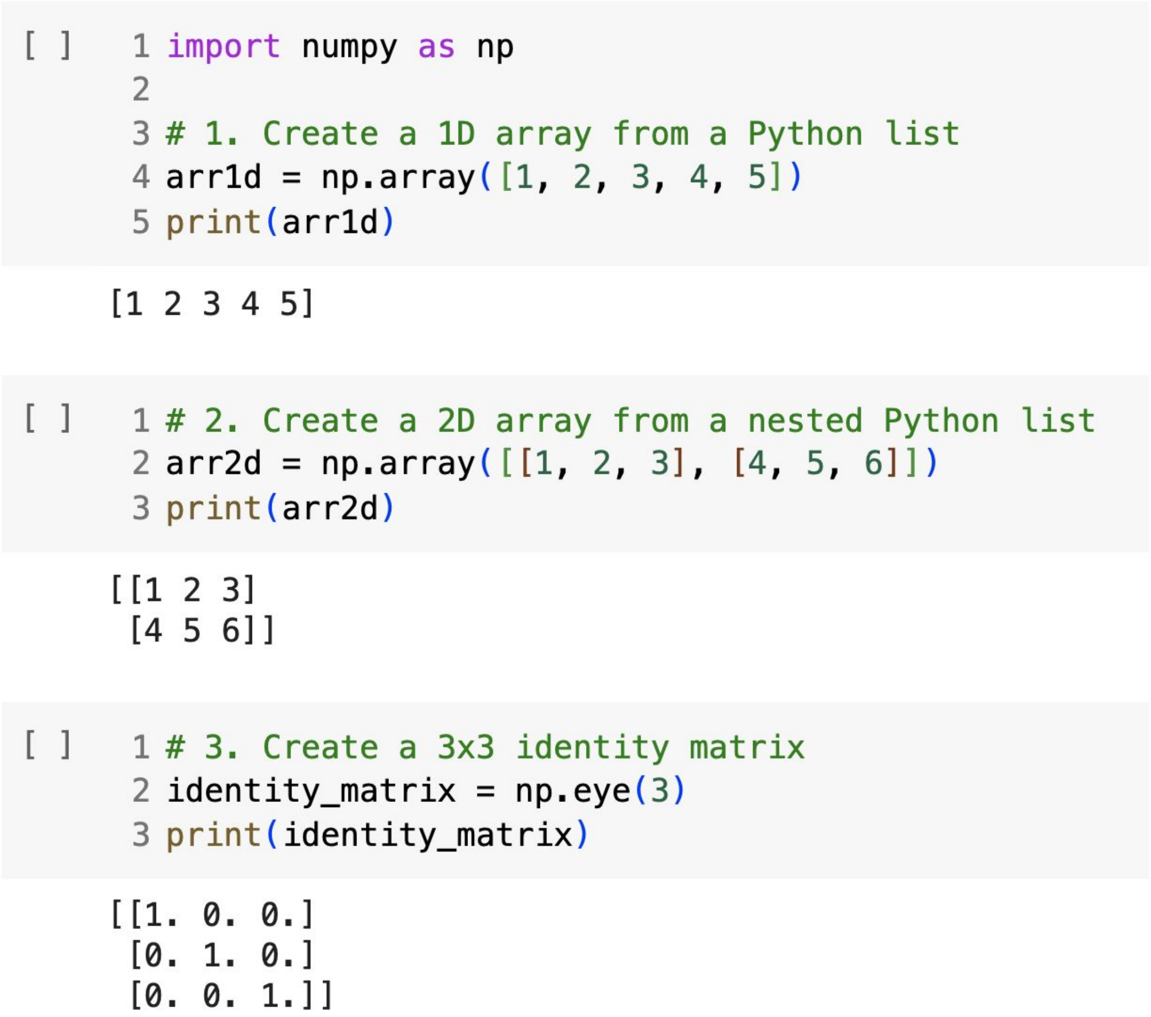
```python
# 4. Add a New Character to the List
new_character = ['Nymphadora Tonks', 'Hufflepuff', 26, 'Witch']
harry_potter_characters.append(new_character)
print("\nUpdated List with Tonks:")
for character in harry_potter_characters:
    print(character)
```

```
Updated List with Tonks:
['Harry Potter', 'Gryffindor', 17, 'Wizard']
['Hermione Granger', 'Gryffindor', 18, 'Witch']
['Ron Weasley', 'Gryffindor', 18, 'Wizard']
['Albus Dumbledore', 'Gryffindor', 115, 'Wizard']
['Severus Snape', 'Slytherin', 38, 'Wizard']
['Luna Lovegood', 'Ravenclaw', 17, 'Witch']
['Sirius Black', 'Gryffindor', 36, 'Wizard']
['Nymphadora Tonks', 'Hufflepuff', 26, 'Witch']
```

Numpy Array

NumPy





```python
import numpy as np

# 1. Create a 1D array from a Python list
arr1d = np.array([1, 2, 3, 4, 5])
print(arr1d)
```

```
[1 2 3 4 5]
```

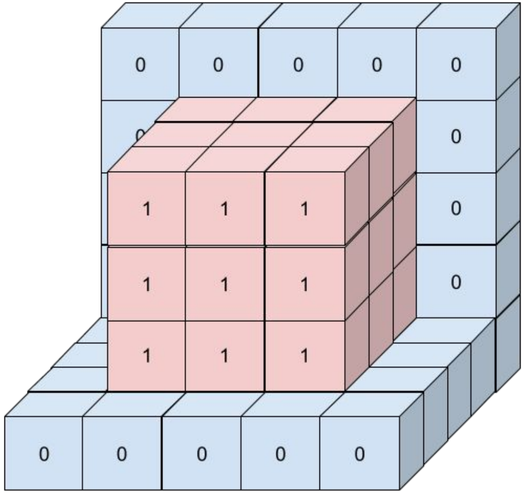
```python
# 2. Create a 2D array from a nested Python list
arr2d = np.array([[1, 2, 3], [4, 5, 6]])
print(arr2d)
```

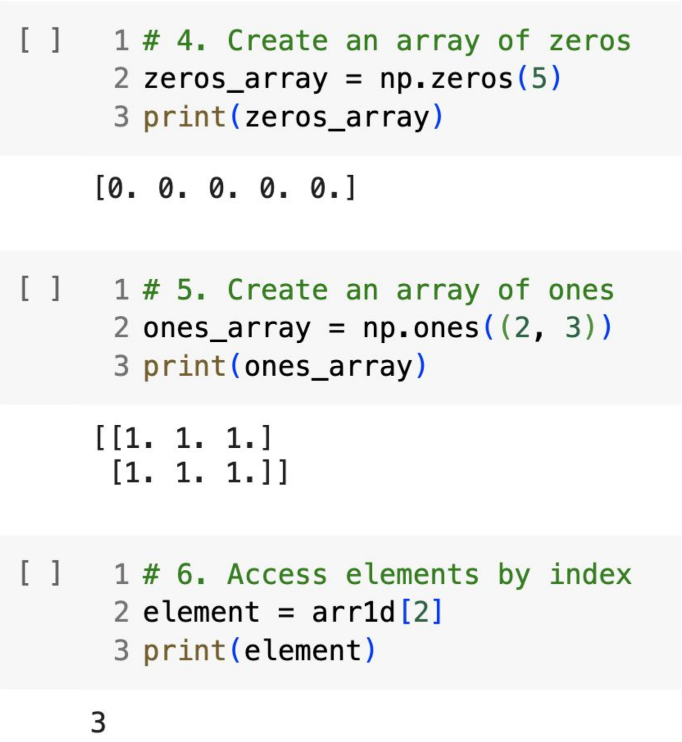
```
[[1 2 3]
 [4 5 6]]
```

```python
# 3. Create a 3x3 identity matrix
identity_matrix = np.eye(3)
print(identity_matrix)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

Numpy Array





```
# 4. Create an array of zeros
zeros_array = np.zeros(5)
print(zeros_array)
```

```
[0. 0. 0. 0. 0.]
```

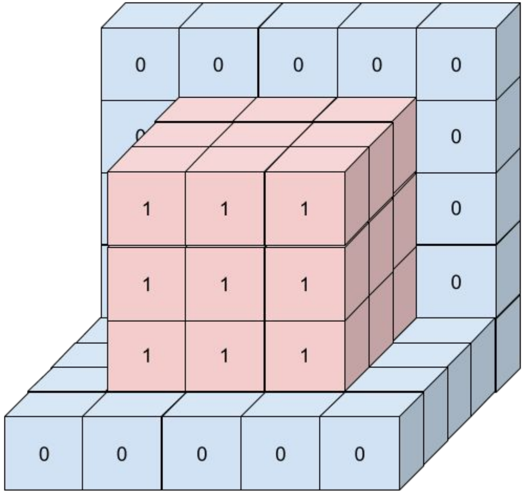
```
# 5. Create an array of ones
ones_array = np.ones((2, 3))
print(ones_array)
```

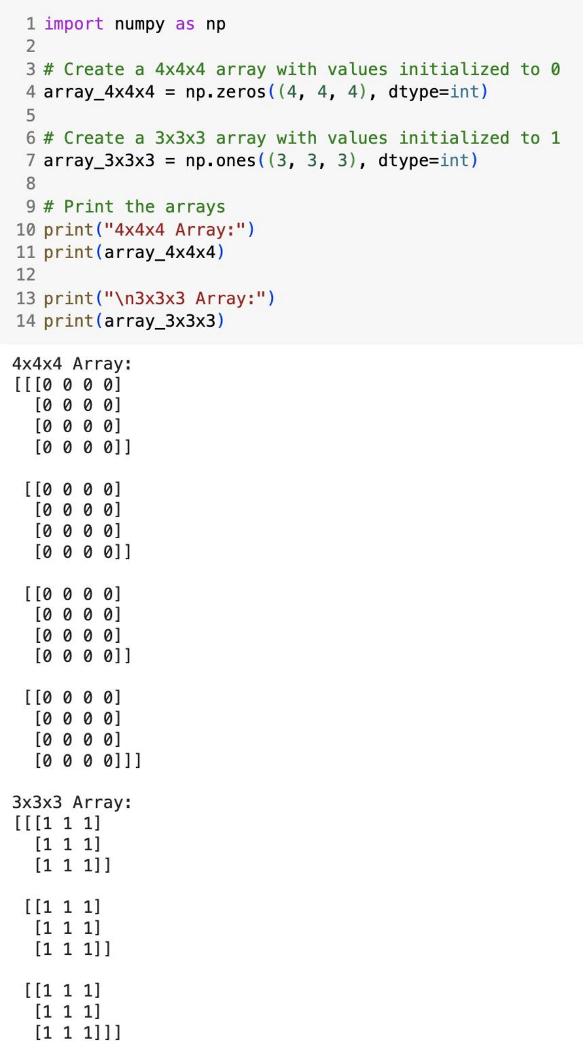
```
[[1. 1. 1.]
 [1. 1. 1.]]
```

```
# 6. Access elements by index
element = arr1d[2]
print(element)
```

```
3
```

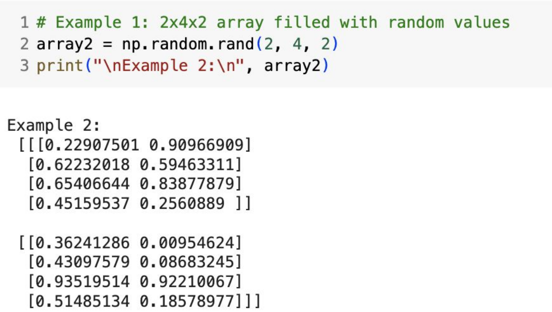
Numpy Array





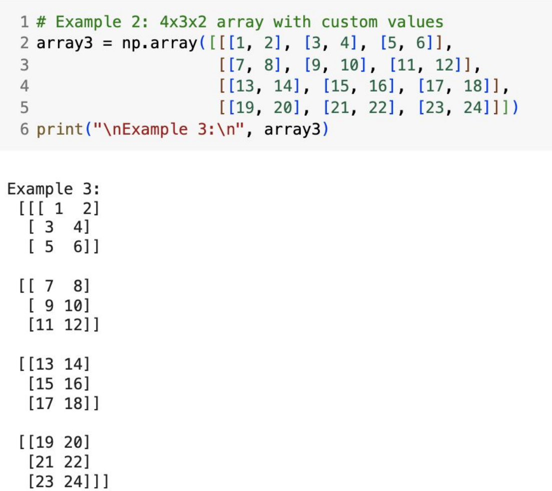
```python
import numpy as np

# Create a 4x4x4 array with values initialized to 0
array_4x4x4 = np.zeros((4, 4, 4), dtype=int)

# Create a 3x3x3 array with values initialized to 1
array_3x3x3 = np.ones((3, 3, 3), dtype=int)

# Print the arrays
print("4x4x4 Array:")
print(array_4x4x4)

print("\n3x3x3 Array:")
print(array_3x3x3)
```

```
4x4x4 Array:
[[[0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]]

 [[0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]]

 [[0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]]

 [[0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]
  [0 0 0 0]]]

3x3x3 Array:
[[[1 1 1]
  [1 1 1]
  [1 1 1]]

 [[1 1 1]
  [1 1 1]
  [1 1 1]]

 [[1 1 1]
  [1 1 1]
  [1 1 1]]]
```



```python
# Example 1: 2x4x2 array filled with random values
array2 = np.random.rand(2, 4, 2)
print("\nExample 2:\n", array2)
```

```
Example 2:
 [[[0.22907501 0.90966909]
  [0.62232018 0.59463311]
  [0.65406644 0.83877879]
  [0.45159537 0.2560889 ]]

 [[0.36241286 0.00954624]
  [0.43097579 0.08683245]
  [0.93519514 0.92210067]
  [0.51485134 0.18578977]]]
```



```python
# Example 2: 4x3x2 array with custom values
array3 = np.array([[[1, 2], [3, 4], [5, 6]],
                   [[7, 8], [9, 10], [11, 12]],
                   [[13, 14], [15, 16], [17, 18]],
                   [[19, 20], [21, 22], [23, 24]]])
print("\nExample 3:\n", array3)
```

```
Example 3:
 [[[ 1  2]
  [ 3  4]
  [ 5  6]]

 [[ 7  8]
  [ 9 10]
  [11 12]]

 [[13 14]
  [15 16]
  [17 18]]

 [[19 20]
  [21 22]
  [23 24]]]
```

Assignment: Exploring AVENGERS Movie Characters

The objective of this assignment is to enhance students' understanding of Python list operations by manipulating and analyzing data related to AVENGERS movie characters using nested lists.