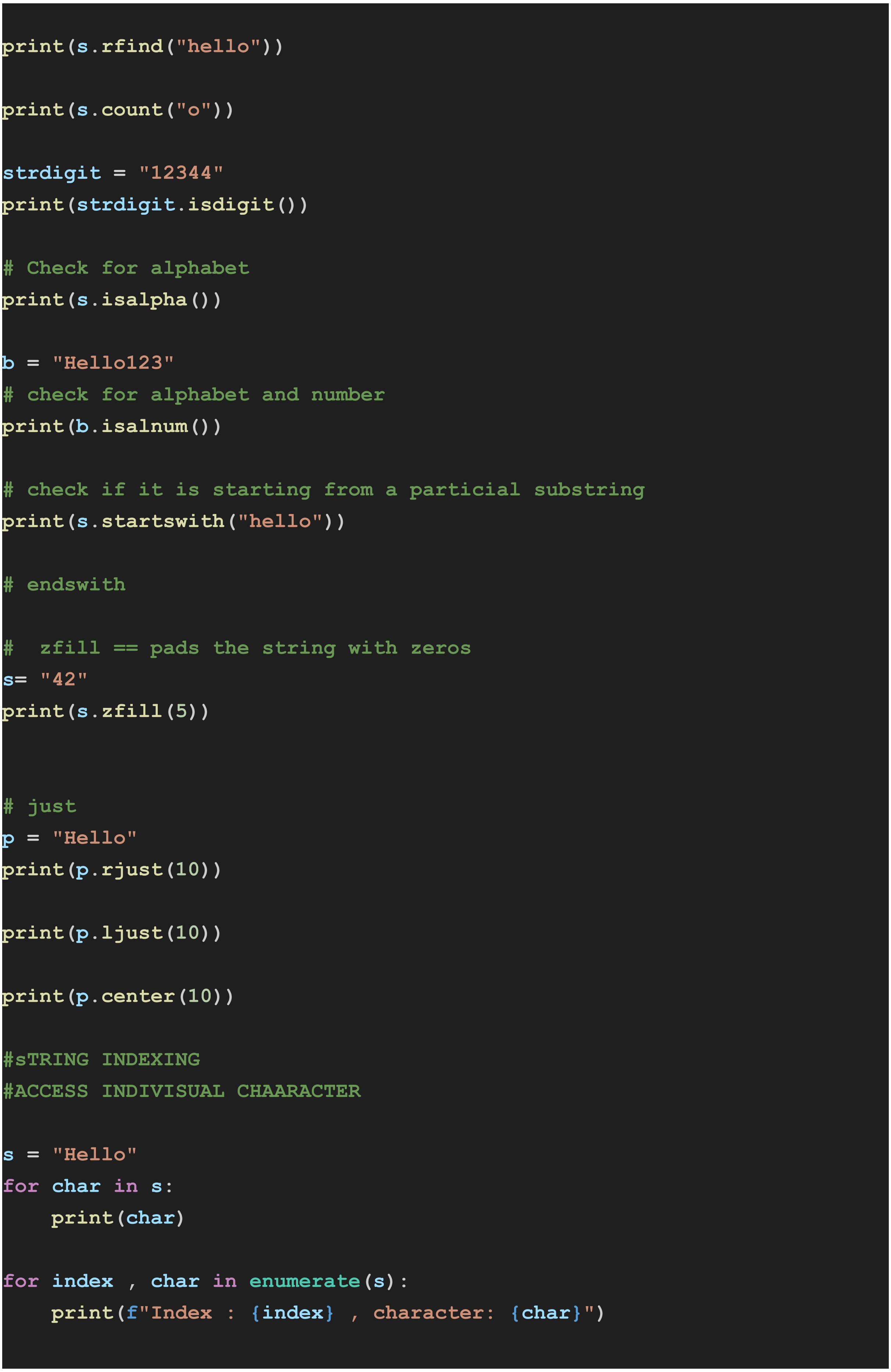
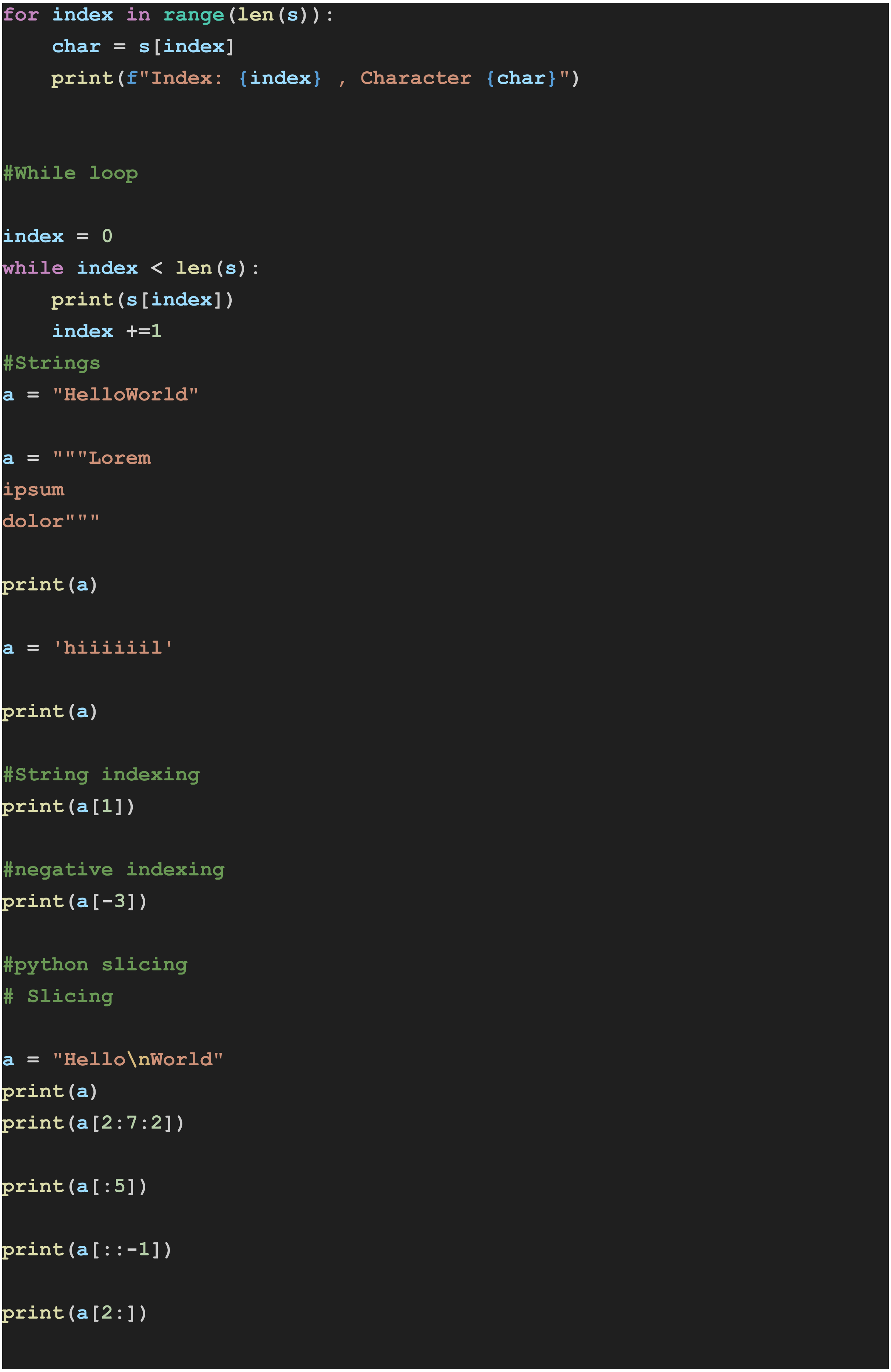
```
# High level level general purpose programming language
# It was created by rossum in 1991
# variables are inferred at runtime --> dynamically types
# Object oriented programming lanugae


x =5
print(x)

# numeric data types
# int
# float
# complex
print(type(x))
```

Strings

```
#Strings
a = "HelloWorld"

a = """Lorem
ipsum
dolor"""

print(a)

a = 'hiiiiiil'

print(a)

#String indexing
print(a[1])

#negative indexing
print(a[-3])

#python slicing
# Slicing

a = "Hello\nWorld"
print(a)
print(a[2:7:2])
print(a[:5])

print(a[::-1])

print(a[2:])

# Convert string to uppercase
print(a.upper())

a = "Hello"
print(a.strip())

print(a.replace("H", "J"))

# # emply seperator does not work
# print(a.split(''))


# string concatenation \
str1 = "hello"
str2 = "World"
print(str1 + " " + str2)


print(str1 * 3)

print(len(str1))



a = "hello world"
print(a)

print(a.title())


s = "helloworldhello"
print(s.lstrip())

print(s.rstrip())

print(s.find("hello"))
```



```
print(s.rfind("hello"))

print(s.count("o"))

strdigit = "12344"
print(strdigit.isdigit())

# Check for alphabet
print(s.isalpha())

b = "Hello123"
# check for alphabet and number
print(b.isalnum())

# check if it is starting from a particial substring
print(s.startswith("hello"))

# endswith

#  zfill == pads the string with zeros
s= "42"
print(s.zfill(5))


# just
p = "Hello"
print(p.rjust(10))

print(p.ljust(10))

print(p.center(10))

#sTRING INDEXING
#ACCESS INDIVISUAL CHAARACTER

s = "Hello"
for char in s:
    print(char)

for index , char in enumerate(s):
    print(f"Index : {index} , character: {char}")
```



```
for index in range(len(s)):
    char = s[index]
    print(f"Index: {index} , Character {char}")


#While loop

index = 0
while index < len(s):
    print(s[index])
    index +=1
#Strings
a = "HelloWorld"

a = """Lorem
ipsum
dolor"""

print(a)

a = 'hiiiiiil'

print(a)

#String indexing
print(a[1])

#negative indexing
print(a[-3])

#python slicing
# Slicing

a = "Hello\nWorld"
print(a)
print(a[2:7:2])

print(a[:5])

print(a[::-1])

print(a[2:])
```

```python
# Convert string to uppercase
print(a.upper())

a = "Hello"
print(a.strip())

print(a.replace("H", "J"))

# # emply seperator does not work
# print(a.split(''))


# string concatenation \
str1 = "hello"
str2 = "World"
print(str1 + " " + str2)


print(str1 * 3)

print(len(str1))



a = "hello world"
print(a)

print(a.title())


s = "helloworldhello"
print(s.lstrip())

print(s.rstrip())

print(s.find("hello"))

print(s.rfind("hello"))

print(s.count("o"))

strdigit = "12344"
print(strdigit.isdigit())
```

```
# Check for alphabet
print(s.isalpha())

b = "Hello123"
# check for alphabet and number
print(b.isalnum())

# check if it is starting from a particial substring
print(s.startswith("hello"))

# endswith

#  zfill == pads the string with zeros
s= "42"
print(s.zfill(5))


# just
p = "Hello"
print(p.rjust(10))

print(p.ljust(10))

print(p.center(10))

#sTRING INDEXING
#ACCESS INDIVISUAL CHAARACTER

s = "Hello"
for char in s:
    print(char)

for index , char in enumerate(s):
    print(f"Index : {index} , character: {char}")

for index in range(len(s)):
    char = s[index]
    print(f"Index: {index} , Character {char}")


#While loop
```

```
index = 0
while index < len(s):
    print(s[index])
    index +=1
```

Exercises

How to reverse a string in Python?

How to check if a string is a palindrome?

How to count occurrences of a substring in a string?

How to check if a string contains only digits?

How to convert a string to uppercase?

How to convert a string to lowercase?

How to capitalize the first letter of each word in a string?

How to remove leading and trailing whitespaces from a string?

How to replace all occurrences of a substring with another substring?

How to find the index of the first occurrence of a substring?

How to check if a string starts with a specific substring?

How to check if a string ends with a specific substring?

How to split a string by a delimiter?

How to join a list of strings into a single string?

How to check if a string contains only alphabetic characters?

How to check if a string contains only alphanumeric characters?

How to remove all vowels from a string?

How to reverse the words in a string?

How to count the number of vowels in a string?

How to convert a string to a list of characters?

How to find the number of words in a string?

How to convert a string to a list of words without spaces?

How to capitalize the first letter of each sentence in a paragraph?

How to convert a string to a list of characters without duplicates?

```
# Python String Questions and Solutions

# 1. How to reverse a string in Python?
def reverse_string(s):
    return s[::-1]

# Example Usage
print(reverse_string("hello"))  # Output: 'olleh'

# 2. How to check if a string is a palindrome?
def is_palindrome(s):
    return s == s[::-1]

# Example Usage
print(is_palindrome("madam"))  # Output: True
print(is_palindrome("hello"))

# 3. How to count occurrences of a substring in a string?
def count_substring(s, sub):
    return s.count(sub)

# Example Usage
print(count_substring("hello world", "o"))  # Output: 2

# 4. How to check if a string contains only digits?
def is_digit(s):
    return s.isdigit()

# Example Usage
print(is_digit("12345"))  # Output: True
print(is_digit("12a45"))  # Output: False

# 5. How to convert a string to uppercase?
def to_uppercase(s):
    return s.upper()

# Example Usage
print(to_uppercase("hello"))  # Output: 'HELLO'

# 6. How to convert a string to lowercase?
def to_lowercase(s):
    return s.lower()

# Example Usage
print(to_lowercase("HELLO"))  # Output: 'hello'

# 7. How to capitalize the first letter of each word in a string?
def capitalize_words(s):
    return s.title()

# Example Usage
print(capitalize_words("hello world"))  # Output: 'Hello World'

# 8. How to remove leading and trailing whitespaces from a string?
```

```python
def remove_whitespace(s):
    return s.strip()

# Example Usage
print(remove_whitespace("   hello   "))  # Output: 'hello'

# 9. How to replace all occurrences of a substring with another substring?
def replace_substring(s, old, new):
    return s.replace(old, new)

# Example Usage
print(replace_substring("hello world", "world", "Python"))  # Output: 'hello Python'

# 10. How to find the index of the first occurrence of a substring?
def find_substring(s, sub):
    return s.find(sub)

# Example Usage
print(find_substring("hello world", "world"))  # Output: 6

# 11. How to check if a string starts with a specific substring?
def starts_with(s, prefix):
    return s.startswith(prefix)

# Example Usage
print(starts_with("hello world", "hello"))  # Output: True

# 12. How to check if a string ends with a specific substring?
def ends_with(s, suffix):
    return s.endswith(suffix)

# Example Usage
print(ends_with("hello world", "world"))  # Output: True

# 13. How to split a string by a delimiter?
def split_string(s, delimiter):
    return s.split(delimiter)

# Example Usage
print(split_string("apple,orange,banana", ","))  # Output: ['apple', 'orange', 'banana']
```

```
# 14. How to join a list of strings into a single string?
def join_strings(lst, delimiter):
    return delimiter.join(lst)

# Example Usage
print(join_strings(['apple', 'orange', 'banana'], ','))  # Output:
'apple,orange,banana'

# 15. How to check if a string contains only alphabetic characters?
def is_alpha(s):
    return s.isalpha()

# Example Usage
print(is_alpha("hello"))  # Output: True
print(is_alpha("hello123"))  # Output: False

# 16. How to check if a string contains only alphanumeric characters?
def is_alphanumeric(s):
    return s.isalnum()

# Example Usage
print(is_alphanumeric("hello123"))  # Output: True
print(is_alphanumeric("hello 123"))  # Output: False

# 17. How to remove all vowels from a string?
def remove_vowels(s):
    return ''.join(char for char in s if char.lower() not in "aeiou")

# Example usage
print(remove_vowels("Hello World"))  # Hll Wrld


# Example Usage

print(remove_vowels("hello world"))

# 18. How to reverse the words in a string?
def reverse_words(s):
    words = s.split()
    return ' '.join(reversed(words))

# Example Usage
```

```
print(reverse_words("hello world"))  # Output: 'world hello'

# 19. How to count the number of vowels in a string?
def count_vowels(s):
    vowels = "aeiouAEIOU"
    return sum(1 for char in s if char in vowels)

# Example Usage
print(count_vowels("hello world"))  # Output: 3

# 20. How to convert a string to a list of characters?
def string_to_list(s):
    return list(s)

# Example Usage
print(string_to_list("hello"))  # Output: ['h', 'e', 'l', 'l', 'o']

# 21. How to find the number of words in a string?
def count_words(s):
    return len(s.split())

# Example Usage
print(count_words("hello world from Python"))  # Output: 4

# 22. How to convert a string to a list of words without spaces?
def string_to_words(s):
    return s.split()

# Example Usage
print(string_to_words("hello world from Python"))  # Output: ['hello', 'world', 'from', 'Python']

# 23. How to capitalize the first letter of each sentence in a paragraph?

def capitalize_sentences(s):
    return '. '.join(sentence.capitalize() for sentence in s.split('. '))
print(capitalize_sentences("hello. how are you? i am fine."))
# Output: "Hello. How are you? I am fine."
# 24. How to convert a string to a list of characters without
duplicates?
def remove_duplicates(s):
    result = ""
    for char in s:
        if char not in result:
            result += char
    return result

# Example usage
print(remove_duplicates("banana"))  # Output: "ban"


# Example Usage
print(remove_duplicates("hello world"))  # Output: 'helo wrd'
```

List

```
# list --> pyhton --> commonly used data type
# ordered items
# Indexed
# Mutable
# Allows duplicate

lst = [1,2,3,4,5]
print(lst)


# mixed data types
mixed_list =[1,"Hello",5]

# nested list
nested_list = [1,2,[3,4]]

# access elements in nested list
print(nested_list[2][1])

#Changing the element
lst[1] = 8
print(lst)
```

```
# Adding the element

# add at the end of the list
lst.append(50)
print(lst)

# Use insert

lst.insert(2,17)
print(lst)

#extend
(lst.extend([60,70]))
print(lst)

lst2 = [90,100]

(lst.extend(lst2))
print(lst)

# remove
# to remove the first occurence
lst.remove(100)
print(lst)

#pop
lst.pop(3)
print(lst)

lst.pop(3)
print(lst)

lst3 = [1,2,3,4]
lst4= [5,6,7,8]
result = lst3 + lst4
print(result)
print(lst3)
print(lst4)


#Membership (in  , not in) --> check if a item is present in list or
not
```

```
print(2 in lst3)
print(2 not in lst3)


# list comprehensions

#create a list of squuare from 0 to 4

squares = [x**2 for x in range(5) ]

print(squares)

# create a list for eve numbers from 0 to 9

evens = [x for x in range(10) if x % 2 == 0]
print(evens)



# Lambda functions
# ANnaymous small func , use lambda keyword for it

# short operations that are passed as argument to functions like map()
, filter()

lst = [1,3,5,6,7]
# doubled_numbers = lambda x: x*2
# print(doubled_numbers(5))


doubled_numbers = list(map(lambda x: x*2 ,lst))
print(doubled_numbers)

numbers = [1,2,3,4,5]
even_number = list(filter(lambda x: x%2 ==0 , numbers))
print(even_number)


words = ["apple" ,"banana" , "cherry"]
firsta = list (filter(lambda word: word[0] == "a" , words))
print(firsta)
```

```
str = "ahello"

uppercase = ''.join(filter((lambda x: x not in["a" ,"e" ,"i" , "o"
,"u"]),(str)))
print(uppercase)
```

Sum Items in List
Multiply Items in List
Get Largest Number in List
Get Smallest Number in List
Count Strings with Same Start and End

```
['abc', 'xyz', 'aba', '1221']
```

Remove Duplicates from List
Hint -> convert to set

Check if List is Empty

Clone or Copy a List
Hint -> use `lst.copy()`

Find Words Longer Than n

Input print(words_longer_than_n(["apple", "banana", "cherry", "date"], 5))
Output ([“banana” , “cherry”])

Check Common Member Between Two Lists
Hint `return any(item in lst1 for item in lst2)`

print(common_member([1, 2, 3, 4], [4, 5, 6, 7]))
Output: True

Remove Specific Elements from List

print(remove_specific_indices(['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow'], {0, 4, 5}))
# Output: ['Green', 'White', 'Black']

Remove Even Numbers from List

Shuffle a List Randomly

```
def shuffle_list(lst):
    random.shuffle(lst)
    return lst
```

Find Second Largest Number in List

Find the Second Smallest Number in a List
Reverse a List Without Using reverse() Method
Find the Cumulative Sum of a List

Input print(cumulative_sum([1, 2, 3, 4]))

Find the Intersection of Two Lists

return list(set(lst1) & set(lst2))

Remove All Occurrences of a Given Element from a List

print(remove_element([1, 2, 3, 4, 2, 2], 2))
Output: [1, 3, 4

Find the Difference Between Two Lists

Find the Frequency of Each Element in a List

```
from collections import Counter

def frequency_dict(lst):
    return dict(Counter(lst))

print(frequency_dict([1, 2, 2, 3, 3, 3, 4]))
# Output: {1: 1, 2: 2, 3: 3, 4: 1}
```

Find the Average of Numbers in a List

Count Occurrences of an Element in a List

Convert a List of Lists into a Single List
Flattern the array
Input

```
print(flatten_list([[1, 2], [3, 4], [5]]))
 # Output: [1, 2, 3, 4, 5]
```

Check If a List Contains Only Unique Elements
Hint : compare original list and set

Rotate a List to the Left by n Places

```
print(rotate_left([1, 2, 3, 4, 5], 2))  # Output: [3, 4, 5, 1, 2]
```

Rotate a List to the Right by n Places

```
print(rotate_right([1, 2, 3, 4, 5], 2))  # Output: [4, 5, 1, 2, 3]
```

Find the Difference Between the Largest and Smallest Number in a List

Reduce

```
const items = ['apple', 'banana', 'apple', 'orange', 'banana', 'banana'];
const count = items.reduce((accumulator, currentValue) => {
  accumulator[currentValue] = (accumulator[currentValue] || 0) + 1;
  return accumulator;
}, {});
console.log(count); // Output: { apple: 2, banana: 3, orange: 1
```

```
from functools import reduce

array = [[1, 2], [3, 4], [5, 6]]
flattened = reduce(lambda x, y: x + y, array)
print(flattened)  # Output: [1, 2, 3, 4, 5, 6]
```

```
from functools import reduce

items = ['apple', 'banana', 'apple', 'orange', 'banana', 'banana']

count = reduce(lambda accumulator, item: accumulator.update({item:
accumulator.get(item, 0) + 1}) or acc, items, {})

print(count)
```