Control Statements ¶

- Control statements in Python include break, continue, and pass. These statements are used to modify the flow of a program.
- These control statements provide flexibility in handling various situations within loops and conditional blocks.
- They allow us to customize the behavior of our code based on specific conditions or requirements.

1.1 break statement:

- The break statement is used to exit the current loop prematurely, regardless of the loop's condition.

In [ ]:

```
words = ["go", "gone", "went", "goa"]
for i in words:
    print(i)
    if 'go'  not in i:
        break
```

Q1. Number Search:

Write a program that takes a list of numbers as input. Use a for loop to iterate through the numbers. If a number greater than 50 is found, break out of the loop. Print the number.

In [ ]:

```
numbers = [10, 25, 60, 45, 30, 55, 70]
for num in numbers:
    if num > 50:
        print(num, 'is greater than 50.')
```

In [ ]:

```
numbers = [10, 25, 60, 45, 30, 55, 70]
for num in numbers:
    if num > 50:
        print(num, 'is greater than 50.')
        break
```

Q2. Password Checker:

Write a program that asks the user to enter a password. Use a while loop to keep prompting the user until the correct password is entered. Use the break statement to exit the loop when the correct password is provided.

In [ ]:

```
correct_password = "secret"
while True:
    user_input = input("Enter the password: ")
    if user_input == correct_password:
        print("Password accepted. Access granted!")
        break
    else:
        print("Incorrect password. Try again.")
```

Q3. Sum of Positive Numbers:

Write a program that takes a list of integers as input. Use a for loop to iterate through the numbers. If a negative number is encountered, break out of the loop. Print the sum of all positive numbers encountered before the break.

In [ ]:

```
numbers = [10, 25, -5, 30, 15, -8]
sum_positive = 0
for num in numbers:
    if num < 0:
        break
    sum_positive += num
print(sum_positive)
```

1.2 continue statement:

- The continue statement is used to skip the rest of the code inside a loop for the current iteration and move to the next iteration.

In [ ]:

```
words = ["go", "gone", "went", "goa"]
for i in words:
    if 'go' in i:
        continue
    print(i)
```

Q1. Even Number Printer:

Write a program that takes a list of numbers as input. Use a for loop to iterate through the numbers. If a number is odd, continue to the next iteration. Print only the even numbers.

In [ ]:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for num in numbers:
    if num % 2 != 0:
        continue
    print(num)
```

Q2. Vowel Counter:

Write a program that takes a string as input. Use a for loop to iterate through the characters. If the character is not a vowel, continue to the next iteration. If it is a vowel, increment a counter. Print the total count of vowels at the end.

In [ ]:

```
input_string = "hello world"
vowel_count = 0
for i in input_string:
    if i not in "aeiou":
        continue
    vowel_count += 1

print(vowel_count)
```

Q3. Positive Number Checker:

Write a program that takes a list of integers as input. Use a for loop to iterate through the numbers. If a negative number is encountered, continue to the next iteration. Print only the positive numbers.

In [ ]:

```
numbers = [-2, 5, -8, 10, -3, 15]
for i in numbers:
    if i < 0:
        continue
    print(i)
```

1.3 pass statement:

- The pass statement is a null operation; nothing happens when it executes. It's used as a placeholder where syntactically some code is required, but you don't want to perform any action.

In [ ]:

```
words = ["go", "gone", "went", "goa"]
for i in words:
    if 'go' in i:
        pass
    else:
        print(i)
```

In [ ]:

```
# In this example, the pass statement is used when the condition x > 5 is t
x = 10
if x > 5:
    pass  # do nothing
else:
    print("x is not greater than 5")
```

In [ ]:

```
x = 4
if x > 5:
    pass  # do nothing
else:
    print("x is not greater than 5")
```

Q1. Even Number Checker:

Write a program that takes a list of numbers as input. Use a for loop to iterate through the numbers. If a number is odd, do nothing; otherwise, print a message indicating that the number is even.

In [ ]:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for num in numbers:
    if num % 2 != 0:
        pass
    else:
        print(num)
```

Q2. Password Checker with Placeholder:

Write a program that asks the user to enter a password. If the password is correct, print "Access granted." If the password is incorrect, do nothing. Use the pass statement.

In [ ]:

```
correct_password = "secret"
user_input = input("Enter the password: ")
if user_input == correct_password:
    print("Access granted.")
else:
    pass
```

Q3. Print Odd Numbers:

Write a program that prints all odd numbers between 1 and 20. Use a for loop and include the pass statement for even numbers.

In [ ]:

```
for num in range(1, 21):
    if num % 2 == 0:
        pass
    else:
        print(num)
```

****