What is whlie loop in python

A while loop in Python repeatedly executes a block of code as long as a given condition is True. The loop continues to run until the condition becomes False.

1) odd-even using while loop

In [1]:

```
num = 1

# Loop through numbers from 1 to 5 (or any range you like)
while num <= 5:
    if num % 2 == 0:
        print(f"{num} is even")
    else:
        print(f"{num} is odd")
    num += 1  # Increment the number
```

```
1 is odd
2 is even
3 is odd
4 is even
5 is odd
```

2) using while loop *

In [2]:

```
row = 1

# While loop to print each row
while row <= 5:
    # Print '*' for each row
    print('* ' * row)
    row += 1  # Increment the row counter
```

```
* 
* * 
* * * 
* * * * 
* * * * * 
```

3) creat list 1-20 numbers list using while loop=> [1,2,3... 20

In [8]:

```
numbers = []

# Initialize the counter
num = 1

# While loop to add numbers to the list
while num <= 20:
    numbers.append(num)
    num += 1  # Increment the number

print(numbers)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

4) creat list 20-1 (revers order) using while loop=> [20,19...1]

In [9]:

```
numbers = []

# Initialize the counter starting at 20
num = 20

# While loop to add numbers to the list in reverse order
while num >= 1:
    numbers.append(num)
    num -= 1  # Decrement the number

print(numbers)
```

```
[20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

5) try with one any eg. break, contnue , pass control statement

break:

In Python, the break statement is used to exit or terminate the loop prematurely, regardless of the loop condition. It is commonly used when a certain condition is met and you want to exit the loop early without waiting for the loop to run until its natural end.

continue :

In Python, the continue statement is used inside a loop to skip the current iteration and move to the next iteration. When the continue statement is encountered, the rest of the code inside the loop for that particular iteration is skipped, and the loop proceeds with the next cycle.

pass :

In Python, the pass statement is a null operation. It is used as a placeholder when you need to have a statement syntactically, but you don't want to perform any action. The pass statement does nothing and allows you to maintain the structure of the program.

In [6]:

```
# Initialize the counter
num = 1

# While loop with break, continue, and pass
while num <= 10:
    if num == 5:
        print("Found 5, using break!")
        break  # Exit the loop when num reaches 5  ,break: When num == 5, the Lo
    elif num == 3:
        print("Skipping 3, using continue!")
        num += 1
        continue  # Skip the rest of the code for num == 3 When num == 3, the cu
    elif num == 8:
        print("Passing on 8, using pass!")
        pass  # Do nothing when num is 8 When num == 8, it simply moves to the n
    else:
        print(f"Number is: {num}")

    num += 1  # Increment the number
```

```
Number is: 1
Number is: 2
Skipping 3, using continue!
Number is: 4
Found 5, using break!
```

In [ ]: