SINCE 2006

FUTURE VISION

UNLOCKING WHILE LOOPS IN PYTHON

8799141678

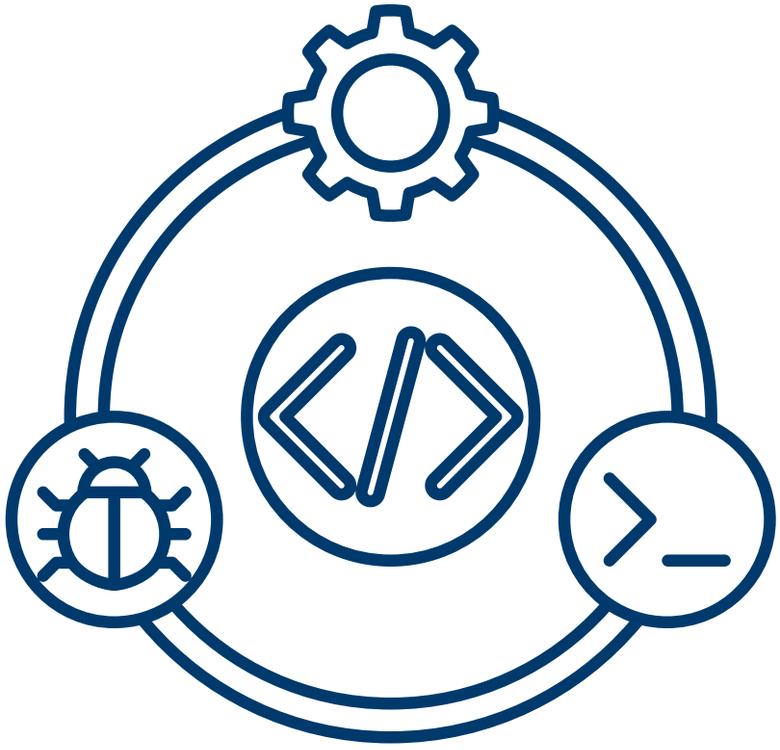
VESU | CITYLIGHT | PAL

SINCE 2006

FUTURE VISION

WHAT IS A WHILE LOOP?

A WHILE LOOP REPEATEDLY EXECUTES A BLOCK OF CODE AS LONG AS A SPECIFIED CONDITION IS TRUE. IT'S GREAT FOR SITUATIONS WHERE YOU DON'T KNOW HOW MANY TIMES YOU NEED TO ITERATE!



BASIC SYNTAX

HERE'S THE BASIC SYNTAX OF A WHILE LOOP:

```
while condition:
    # Code block to execute
```

EXAMPLE 1 - SIMPLE COUNTER

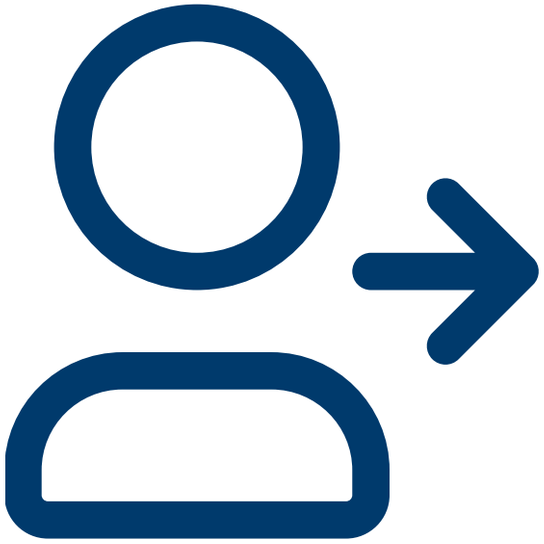
LET'S SEE A BASIC EXAMPLE OF A COUNTER:

```
count = 0
while count < 5:
    print(count)
    count += 1
Output:

Copy code
0
1
2
3
4
```

EXAMPLE 2 - USER INPUT

YOU CAN USE WHILE LOOPS TO TAKE USER INPUT UNTIL A CONDITION IS MET:



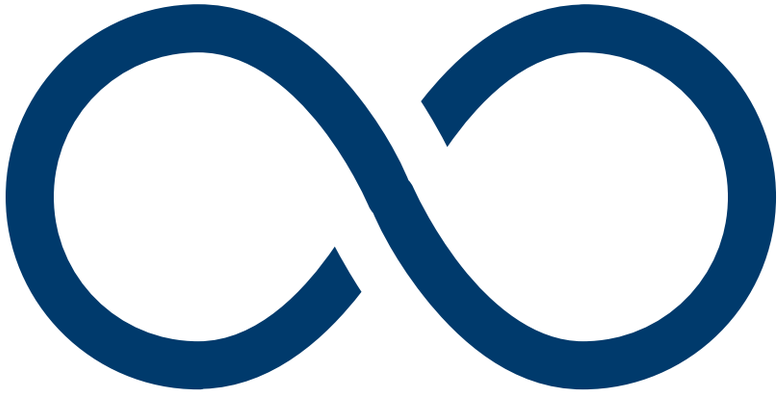
```
user_input = ""
while user_input.lower() != "exit":
    user_input = input("Type 'exit' to quit: ")
```

SINCE 2006

FUTURE VISION

INFINITE LOOPS

BE CAREFUL! IF THE CONDITION NEVER BECOMES FALSE, YOU'LL CREATE AN INFINITE LOOP. ALWAYS ENSURE YOUR LOOP CAN EXIT.



```
# Example of an infinite loop (don't run this!)
while True:
    print("This will run forever!")
```

PRACTICAL EXAMPLE - GUESSING GAME

LET'S CREATE A SIMPLE GUESSING GAME!

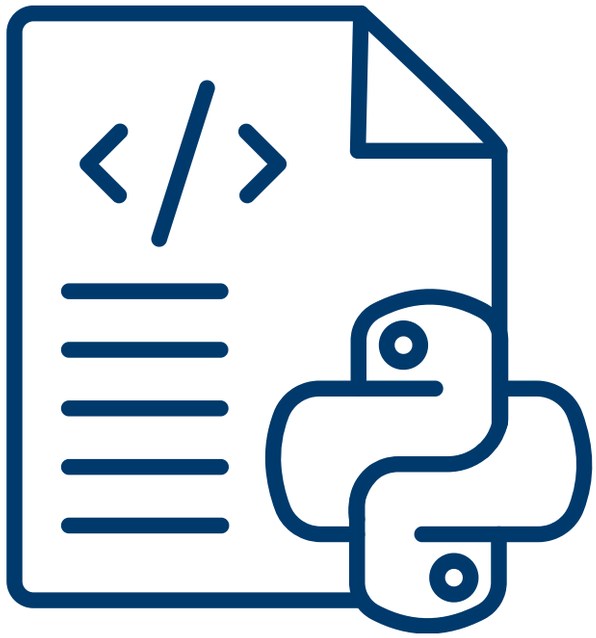
```
secret_number = 7
guess = 0

while guess != secret_number:
    guess = int(input("Guess the number (1-10): "))
    if guess < secret_number:
        print("Too low!")
    elif guess > secret_number:
        print("Too high!")

print("Congratulations! You've guessed it!")
```

CONCLUSION

WHILE LOOPS ARE POWERFUL FOR REPETITIVE TASKS WHEN THE NUMBER OF ITERATIONS ISN'T KNOWN IN ADVANCE. USE THEM WISELY!



FOLLOW US
FOLLOW US
FOLLOW US

Since 2006

FUTURE VISION

Computer institute

Follow Us

+91 87991 41678

futurevisioncomputers.com

PAL, CITYLIGHT, VESU