Flow Control
(while Loop and break and continue Statement)

while Loop – break and continue

- Do something WHILE a condition is **True**.
- Loop will STOP when condition is **False**.
- **break** statement is executed to **exit the while loop**.
- when **continue** statement is executed, control **returns to beginning of loop**.

TASK: **Find the SUM of numbers from 1 – 10.** **Save As: today's date1.py**

REVIEW: Pseudocode Algorithm

```
set the variable lastNum to 10
set the variable sum to 0
set the variable num to 1
while test expression:
        add the number to the sum
        increase num by 1 (to go to the next number)
        repeat the loop

outside the loop print the final sum (when the loop is done and you've reached
the lastNum)
```

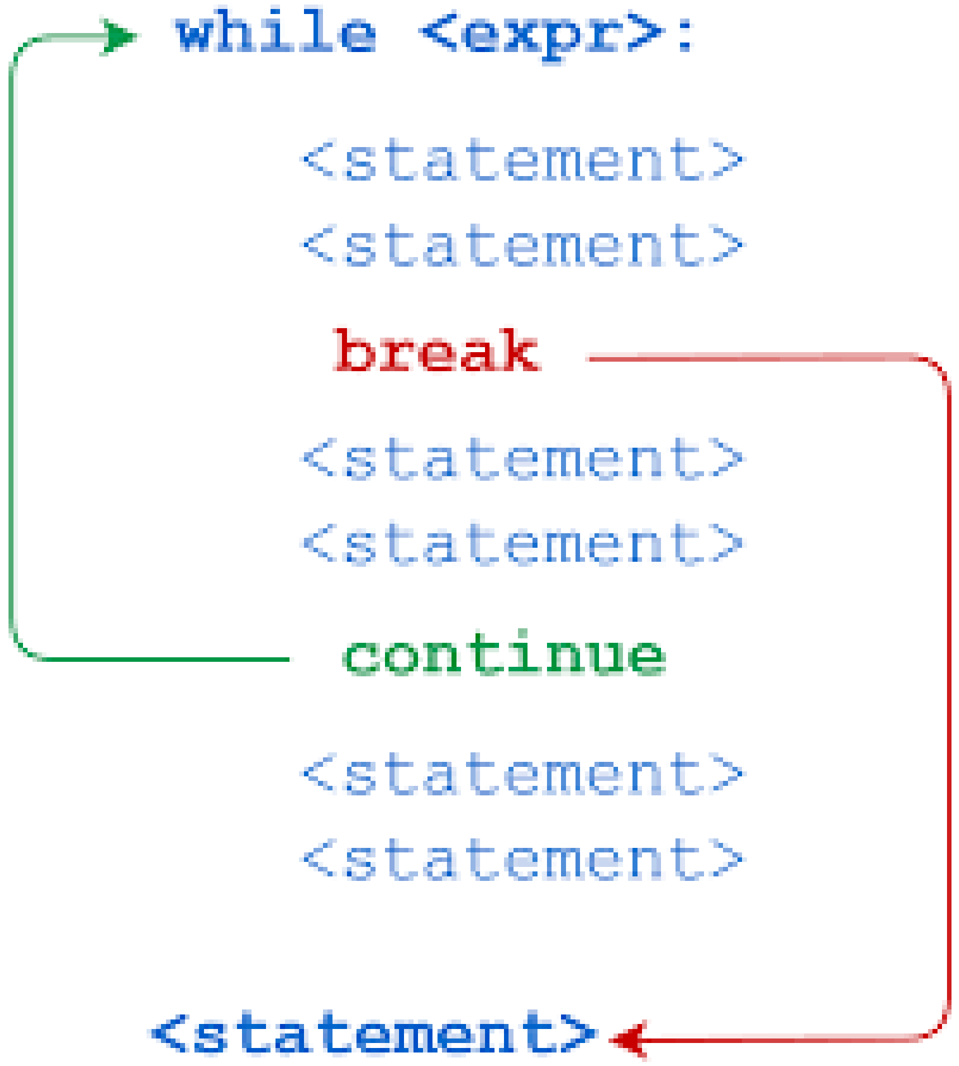
TASK: **Find the SUM of numbers entered from a user, or just press enter to stop. Loop will stop when user presses <enter> (empty string “”)** **Save As: today’s date2.py**

REVIEW: Pseudocode Algorithm

```
set the theSum to 0.0
prompt the user to input a number or press <enter> to stop the loop, assign to the variable data.
while test expression:
        convert the string to a float, assign to the variable number
        add the float to the variable theSum
        prompt the user to input again

print the variable theSum (outside the loop)
```

while Loop using a break and/or continue Statement



```
while <expr>:
    <statement>
    <statement>
    break
    <statement>
    <statement>
    continue
    <statement>
    <statement>

<statement>
```

TASK:

Find the SUM of numbers entered from a user, or just press enter to stop.

Enter Code: today’s_date3.py

```
theSum = 0.0
while True:
    data = input(“Enter a number or just press enter to quit: “)
    if data == “”:
        break
    number = float(data)
    theSum = theSum + number

#The next statement is outside the loop
print(“The sum is “, theSum)
```

```
theSum = 0.0                                   #Same with Comments
while True:
    data = input(“Enter a number or just press enter to quit: “)
    if data == “”:                             #empty string “”
        break                                  #break will exit the loop
    number = float(data)                       #convert string to float number
    theSum = theSum + number                   #sum all the numbers

#The next statement is outside the loop
print(“The sum is “, theSum)
```

TASK:

Enter a numeric grade from 0 – 100. Make sure to handle any invalid numeric grades.

Enter Code: today’s_date4.py

if number >= 0 and number <= 100

is True, **break** will exit out

of the while loop

```
while True:
    number = int(input(“Enter a numeric grade from 0 – 100 ”))
    if number >= 0 and number <=100:
        break
    else:
        print(“Error: grade must be between 0 – 100 “)

print(“The grade is “, number)
```

#If a user enters a number from 0 to 100, the if condition will be True and the break Statement will cause an exit from the while loop.

#If a user doesn’t enter a number from 0 - 100, the else block will execute and the Error Message will print, and the while loop continues prompting the user for a numeric grade again.

```
var = 10

while var > 0:
    print(“Current Value: “, var)
    var = var – 1
    if var == 5:
        break

print(“Good bye!”)
```

Enter Code:
today’s_date5.py

#if condition var == 5 is True, **break** will exit out of the while loop

while Loop using a continue Statement

#The **continue** statement returns the control to the beginning of the while loop.

#The **continue** statement rejects all the remaining statements in the loop.

```
var = 11
while var > 0:
    var = var – 1
    if var == 5:
        continue
    print(“Current Value: “, var)

print(“Good bye!”)
```

Enter Code: today’s_date5.py

#if condition var == 5 is True, **continue** will return back to the beginning of the loop.

Search the Internet for more while loop break or continue Examples