



# WHILE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

**While loops** run until a given logical expression becomes FALSE

- In other words, the loop runs *while* the expression is TRUE

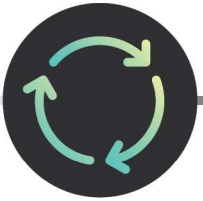
```
while logical expression:
```

Indicates a  
While Loop

A logical expression that  
evaluates to TRUE or FALSE

**Examples:**

- `counter < 10`
- `inventory > 0`
- `revenue > cost`



# WHILE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

**While loops** run until a given logical expression returns FALSE

- In other words, the loop runs *while* the expression is TRUE

```
while logical expression:  
    do this
```

*Code to run while the logical  
expression is TRUE (must be  
indented!)*



# WHILE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

While loops often include **counters** that grow with each iteration

- Counters help us track how many times our loop has run
- They can also serve as a backup condition to exit a loop early (*more on this later!*)

```
counter = 0
while counter < 10:
    counter += 1
    print(counter)
```

This is an “addition assignment”,  
which adds a given number to the  
existing value of a variable:

`counter = counter + 1`

1 ← Counter increases to 1 in the first iteration

2 ← Counter increases to 2 in the second iteration

.

.

10 ← When the counter increments to 10, our condition becomes False,  
and exits the loop



# WHILE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

## EXAMPLE

Run a calculation until a goal is reached

```
# starting portfolio balance is 800000
stock_portfolio = 800000
year_counter = 0

while stock_portfolio < 1000000:
    # calculate annual investment income
    investment_income = stock_portfolio * .05 # 5% interest rate

    # add income to end of year portfolio balance
    stock_portfolio += investment_income

    # add one each year
    year_counter += 1

    print(f'At the end of year {year_counter}: '
          + f'My balance is ${round(stock_portfolio, 2)}')
```

The while loop here will run while stock\_portfolio is less than 1m

stock\_portfolio **starts at 800k** and **increases by 5%** of its value in each run:

1. 800k < 1m is TRUE
2. 840k < 1m is TRUE
3. 882k < 1m is TRUE
4. 926k < 1m is TRUE
5. 972k < 1m is TRUE
6. 1.02m < 1m is FALSE (**exit**)

```
At the end of year 1: My balance is $840000.0
At the end of year 2: My balance is $882000.0
At the end of year 3: My balance is $926100.0
At the end of year 4: My balance is $972405.0
At the end of year 5: My balance is $1021025.25
```



# WHILE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

## EXAMPLE

*Calculating bank balance until we're out of money.*

```
bank_balance = 5000
month_counter = 0

while bank_balance > 0:
    spending = 1000
    bank_balance -= spending
    month_counter += 1
    print(f'At the end of month {month_counter}: '
          + f'My balance is ${round(bank_balance, 2)}')
```

```
At the end of month 1: My balance is $4000
At the end of month 2: My balance is $3000
At the end of month 3: My balance is $2000
At the end of month 4: My balance is $1000
At the end of month 5: My balance is $0
```



**PRO TIP:** Use “-=” to subtract a number from a variable instead (subtraction assignment)



# INFINITE LOOPS

Loop Basics

For Loops

While Loops

Nested Loops

Loop Control

A while loop that *always* meets its logical condition is known as an **infinite loop**

- These can be caused by incorrect logic or uncertainty in the task being solved

```
stock_portfolio = 800000
year_counter = 0

while stock_portfolio < 1000000:
    investment_income = stock_portfolio * 0 # 0% interest rate
    stock_portfolio += investment_income
    year_counter += 1
    print(f'At the end of year {year_counter}: '
          + f'My balance is ${round(stock_portfolio, 2)}')
```

```
At the end of year 51461: My balance is $800000
At the end of year 51462: My balance is $800000
At the end of year 51463: My balance is $800000
```

-----  
KeyboardInterrupt

This indicates a manually  
stopped execution

The while loop here will run while stock\_portfolio is less than 1m, which **will always be the case**, as it's not growing due to 0% interest



If your program is stuck in an infinite loop, you will need to **manually stop it** and modify your logic

