Computer Science   Iterations

1. **basic `while` loop**

    ```python
            # establish condition that will have value of true
            # or false, and then...

            while <boolean condition> :
                # do these
                # statements
    ```

2. **`while` loop as a counter**

    ```python
            i = 0
            while i < finalValue:
                # statements
                i = i + 1
    ```

3. **basic `for` (counting) loop**

    ```python
            for i in range(finalValue):     # starts at 0, increments by 1, continues
                                            # while i < finalValue
                # do these
                # statements
    ```

4. **Advanced `for` loop**

    ```python
            for i in range(initialValue, finalValue, increment):
                # the body of this is repeated with i beginning at the initialValue
                # its value increased by the increment each time through the loop,
                # and continuing as long as i is less than the finalValue
                # for i in range(2,10,2):... would repeat four times (i = 2, 4, 6, 8)
    ```

5. **Sentinel loop looking for a signal to end the loop**

    ```python
            val = input("Enter a numeric value, or "q' to quit: ")
            while val.lower() != "q":
                numeric_val = float(val)  // converts val to a number
                if numeric_val > 1000000:
                    print("That's a really big number!")
                elif numeric_val < 0.0000001:
                    print("That's a really small number!")
                else:
                    print("That"s an interesting number?")
                val = input("Enter another numeric value, or 'q' to quit: ")
            print("Our loop is all done.")
    ```

6. **Error checking loop (using a `break` statement to exit the loop body)**

```
while True:            # infinite loop unless we break out of it!
    val = float(input("Enter a number greater than 0: "))
    if (input > 0):
        break
    else:
        print("Thank you.")
# program continues here
```

7. **Nested loops (using `for` as an example)**

```
for row in range(height):
    for col in range(width):
        # do something with
        # data at data[row][col]
    # do something at the end of each column?
# program continues here
```

EXERCISES

1. Write a `while` loop that prints the numbers from 1 to 20, as well as their squares, in this format:
```
1 squared = 1
2 squared = 4
3 squared = 9
.
.
.
```

2. Write a `for` loop that counts from 0 to 100 and prints out each number.

3. Write a `while` loop that asks the user to enter a series of positive numbers that will be added. The loop stops accepting input when the user enters a 0. Then print out the sum of those numbers.

4. Write a `for` loop that prints out the numbers 1, 4, 7, 10, 13, ... , 298, 301.

5. Write a `while` loop that prints out the numbers 0, 4, 8, 12, ... , 96, 100.

6. Write an infinite loop that has the user repeatedly enter passwords until he/she enters the correct password, a password of your choosing. Once the password is entered, `break` out of the infinite loop.

7. Write a loop that displays the Fibonacci sequence. The first two numbers in the Fibonacci sequence are `0` and `1`. Subsequent numbers are found by adding the previous two numbers, so the sequence begins `0, 1, 1, 2, 3, 5, 8, 13, ...`

8. Write a "prime finder" loop that determines whether a given number `n` is prime or not. Any integer `n > 2` is prime if no number between `2` and $\sqrt{n}$ (inclusive) evenly divides into `n`. The loop should return `true` if `n` is prime and `false` if `n` is not prime.