VIDEO 4 : For & Range

Another thing every programming language must do is to provide a way to perform the same action multiple times. The keyword for is one one way in which you can loop through code while executing repeatedly.

For as long as you have more hamburger to eat, keep eating that hamburger.

For loops come in many forms.

You can loop through a list of values. That list is surrounded with square brackets. Each time through the list the next value will be assigned to the variable I in this example.

CODE

```
for i in [2,4,6,8,10]:
    print("i = ", i)
```

We can also have the keyword range define our list for us. range(10) will create a list starting at 0 and go up to, but not include 10, the number passed into it.

We can now combine range with a for loop to print the numbers 0 through 4 like so.

CODE

```
for i in range(5):
    print("i = ", i)
```

We can also define the starting and ending values with range. This time we'll print the values 2 through 4.

CODE

```
for i in range(2, 5):
    print("i = ", i)
```

Now before you solve another problem I want to teach you how to test if a number is odd or even. If you divide any even number by 2 it will not have a remainder. The modulus operator provides the remainder of a division as I covered previously.

So if I % 2 == 0 we know that I is an even value.

CODE

```
I = 6
print("Is 6 even :", ((i % 2) == 0))
```

Python Problem for you to Solve

Use the knowledge you have gained to print out all odd numbers from 1 to 20. Feel free to look at everything on this page to help.

Solution

CODE

```
# Use for to loop through the list from 1 to 21
for i in range(1, 21):

# Use modulus to check that the result is NOT EQUAL to 0
# Print the odds

    if ((i % 2) != 0):
        print("i = ", i)
```

More About Floats

As you recall, floats are values that have decimal values. We can convert string input into a float like this

CODE

```
your_float = input(“Enter a float :”)
your_float = float(your_float)
```

We can also use format with print to define the number of decimals displayed in output. If you wanted to show 2 decimal values you would use {:.2f} like in this example.

CODE

```
print("Rounded to 2 decimals : {:.2f}”.format(your_float))
```

2nd Python Problem for you to Solve

In this problem you will calculate how much money a person will have after investing for 10 years. Compounding interest is the act of reinvesting each years interest payment and then receiving interest on the initial value as well as on interest payments.

Your program will :

1. Have the user enter their investment amount and expected interest
2. Each year their investment will increase by their investment + their investment * the interest rate
3. Print out their earnings after a 10 year period

Give it a try using everything already covered above.

SOLUTION

```
# Ask for money invested + the interest rate
money = input("How much to invest: ")
interest_rate = input("Interest Rate: ")

# Convert value to a float
money = float(money)
```

```
# Convert value to a float and round the percentage rate by 2 digits
interest_rate = float(interest_rate) * .01

# Cycle through 10 years using for and range from 0 to 9
for i in range(10):

# Add the current money in the account + interest earned that year
money = money + (money * interest_rate)

# Output the results
print("Investment after 10 years: {:.2f}".format(money))
```

Whether you got the problem right doesn't matter. The goal was to get you to think in new ways and to understand the final code.

Order of Operations

When a computer tries to solve a calculation it follows certain rules. It will for example multiply values next to each other before adding them, no matter what the order is. These rules are known as the Order of Operations. Calculations will occur in this order :

1. exponentiation and root extraction
2. multiplication and division
3. addition and subtraction

Here is an example to help you understand.

CODE

```
print("2 + (3 * 4) = ", (2 + (3 * 4)))
print("(2 + 3) * 4 = ", ((2 + 3) * 4)))
```

That's it for this video. In the next video I'll cover While Loops, Random Values, Break, Continue, and I'll provide more problems for you to solve.