VIDEO 12 : LISTS 2

In this video we'll cover List Comprehensions, Multidimensional Lists and you'll have to solve another problem.

List Comprehensions

You can construct lists in interesting ways using list comprehensions. You can do this by performing an operation on each item in the list.

CODE

```
import math
# Create a list of even values
even_list = [i*2 for i in range(10)]

for k in even_list:
    print(k, end=", ")
print()

# List of lists containing values to the power of
# 2, 3, 4
num_list = [1,2,3,4,5]

list_of_values = [[math.pow(m, 2), math.pow(m, 3), math.pow(m, 4)]
                  for m in num_list]

for k in list_of_values:
    print(k)
print()

# Create a 10 x 10 list
multi_d_list = [[0] * 10 for i in range(10)]

# Change a value in the multidimensional list
multi_d_list[0][1] = 10

# Get the 2nd item in the 1st list
# It may help to think of it as the 2nd item in the 1st row
print(multi_d_list[0][1])

# Get the 2nd item in the 2nd list
print(multi_d_list[1][1])
```

Multidimensional Lists

Multidimensional list are tables of data that spans across rows and columns. Here I'll show how indexes work with a multidimensional list.

CODE

```
# Create the multidimensional list 10 x 10
list_table = [[0] * 10 for i in range(10)]
```

```
# Fill the list with values
for i in range(10):

    for j in range(10):
        list_table[i][j] = "{} : {}".format(i, j)

for i in range(10):

    for j in range(10):
        print(list_table[i][j], end=" || ")
    print()
```

Python Problem for you to Solve

With 2 for loops fill the cells in a multidimensional list with a multiplication table using values 1 - 9. This is your goal.

1, 2, 3, 4, 5, 6, 7, 8, 9,
2, 4, 6, 8, 10, 12, 14, 16, 18,
3, 6, 9, 12, 15, 18, 21, 24, 27,
4, 8, 12, 16, 20, 24, 28, 32, 36,
5, 10, 15, 20, 25, 30, 35, 40, 45,
6, 12, 18, 24, 30, 36, 42, 48, 54,
7, 14, 21, 28, 35, 42, 49, 56, 63,
8, 16, 24, 32, 40, 48, 56, 64, 72,
9, 18, 27, 36, 45, 54, 63, 72, 81

Solution

CODE

```
# Create the multidimensional list
mult_table = [[0] * 10 for i in range(10)]

# This will increment for each row
for i in range(1, 10):

    # This will increment for each item in the row
    for j in range(1, 10):

        # Assign the value to the cell
        mult_table[i][j] = i * j

# Output the data in the same way you assigned it
for i in range(1, 10):

    for j in range(1, 10):
        print(mult_table[i][j], end=", ")

    print()
```

That’s it for this video. In the next video we’ll cover Dictionaries.