JJenkins_HW #2

August 20, 2018

1 JJenkins Assignment 2

1.1 Exercise 1 (6 Marks)

- Create a list with 10 string or name items and print the list
- Remove the last 3 items from the list
- Append 3 new items to the list
- Insert 1 new item between items 4 and 5
- Change item 2 to something different
- Take a slice from the list and print it

Remember to print each result. You can do all items in one Python cell if you want. Use remark (#) statements to label each part of your solution. If you need to separate the output for neatness, you can insert an empty print() statement or use whitespace.

Use the code cell below for your solution.

```
In [1]: print("Create a list with 10 string or name items and print the list")
        my_furniture = ['couch','chair','table','desk', 'stool','bench', 'bed', 'hammock', 'dr
        print("\nMy list of 10 strings is the following list of furniture", my_furniture)
        for piece in my_furniture:
            print(piece)

        print("\nRemove the last 3 items from the list")
        last_3 = range(-3, 0)
        for index in last_3:
            del my_furniture[index]
        print("\nThere are " + str(len(my_furniture)) + " items of furniture after I remove the
        print("My list of " + str(len(my_furniture)) + " items of furniture now includes the f
        for piece in my_furniture:
            print(piece)

        print("\nAppend 3 new items to the list")
        # adding 3 new items to my list
        my_furniture.append('coat rack')
        my_furniture.append('hutch')
        my_furniture.append('book case')
        print("\nMy appended list of strings is the following " + str(len(my_furniture))
              + " item list of furniture:" + "\n", my_furniture)
        for piece in my_furniture:
            print(piece)

        print("\nInsert 1 new item between items 4 and 5")
        my_furniture.insert(4, 'pool table')
        print("Inserting a new item between items 4 and 5, my " + str(len(my_furniture))
              + " item list is now the following:" "\n", my_furniture)
        for piece in my_furniture:
            print(piece)

        print("\nChange item 2 to something different")
        my_furniture[1] = 'curio cabinet'
        print("Changing item 2 to something different, my " + str(len(my_furniture))
              + " item list is now the following:" "\n", my_furniture)
        for piece in my_furniture:
            print(piece)

        print("\nTake a slice from the list and print it")
        first_piece = my_furniture[0]
        print(my_furniture[0])
        print("The first piece of furniture in my list is a " + first_piece + (".\n"))
        last_piece = my_furniture[-1]
        print(last_piece)
        print("The last piece of furniture in my list is a " + last_piece + (".\n"))
        print("The middle pieces of furniture in my list include: ")
        print(my_furniture[1:-1])
        print()
```

```
Create a list with 10 string or name items and print the list

My list of 10 strings is the following list of furniture ['couch', 'chair', 'table', 'desk', '
couch
chair
table
desk
stool
bench
bed
hammock
dresser
ottoman

Remove the last 3 items from the list

There are 7 items of furniture after I remove the last 3 items for the list.
My list of 7 items of furniture now includes the following:
 ['couch', 'chair', 'table', 'desk', 'stool', 'bench', 'bed']
couch
```

```
chair
table
desk
stool
bench
bed

Append 3 new items to the list

My appended list of strings is the following 10 item list of furniture:
 ['couch', 'chair', 'table', 'desk', 'stool', 'bench', 'bed', 'coat rack', 'hutch', 'book case
couch
chair
table
desk
stool
bench
bed
coat rack
hutch
book case

Insert 1 new item between items 4 and 5
Inserting a new item between items 4 and 5, my 11 item list is now the following:
 ['couch', 'chair', 'table', 'desk', 'pool table', 'stool', 'bench', 'bed', 'coat rack', 'hutcl
couch
chair
table
desk
pool table
stool
bench
bed
coat rack
hutch
book case

Change item 2 to something different
Changing item 2 to something different, my 11 item list is now the following:
 ['couch', 'curio cabinet', 'table', 'desk', 'pool table', 'stool', 'bench', 'bed', 'coat rack
couch
curio cabinet
table
desk
pool table
stool
bench
bed
```

```
coat rack
hutch
book case

Take a slice from the list and print it
couch
The first piece of furniture in my list is a couch.

book case
The last piece of furniture in my list is a book case.

The middle pieces of furniture in my list include:
['curio cabinet', 'table', 'desk', 'pool table', 'stool', 'bench', 'bed', 'coat rack', 'hutch']
```

1.2 Exercise 2 (3 Marks)

Write a program that calculates and prints the first 50 powers of 2 minus 1. Instead of using n**2, you will need the following function (which we used earlier in the course):

2**n - 1

Hint: Use the squares program in Notebook 2 as a model.

Use the code cell below for your solution.

In [2]:
```
# setting my n to zero and creating a variable for my exponent
n = 0
exponent = n-1

# using a while loop to to loop through my counts of n from 0 to 50
while n < 51:
    # calculating 2 to the power of (n-1) for each count of n from 0 to 50
    pwrs_of_2minus1 = 2**exponent
    # printing out the value of 2 to the power of (n-1) for each count of n
    print("2 to the power of (" + str(n) + " minus 1), also written as 2^(" + str(n) +
          + "2^(" + str(exponent) + ") and has a value of " + str(pwrs_of_2minus1) +".
    # incrementing my count for this loop iteration to be passed to the next iteration
    n = n + 1
    # incrementing my exponent to be passed to the next iteration
    exponent = n -1
```

```
2 to the power of (0 minus 1), also written as 2^(0-1) is equal to 2^(-1) and has a value of 0
2 to the power of (1 minus 1), also written as 2^(1-1) is equal to 2^(0) and has a value of 1.
2 to the power of (2 minus 1), also written as 2^(2-1) is equal to 2^(1) and has a value of 2.
2 to the power of (3 minus 1), also written as 2^(3-1) is equal to 2^(2) and has a value of 4.
2 to the power of (4 minus 1), also written as 2^(4-1) is equal to 2^(3) and has a value of 8.
2 to the power of (5 minus 1), also written as 2^(5-1) is equal to 2^(4) and has a value of 16
2 to the power of (6 minus 1), also written as 2^(6-1) is equal to 2^(5) and has a value of 32
2 to the power of (7 minus 1), also written as 2^(7-1) is equal to 2^(6) and has a value of 64
```

```
2 to the power of (8 minus 1), also written as 2^(8-1) is equal to 2^(7) and has a value of 12
2 to the power of (9 minus 1), also written as 2^(9-1) is equal to 2^(8) and has a value of 25
2 to the power of (10 minus 1), also written as 2^(10-1) is equal to 2^(9) and has a value of
2 to the power of (11 minus 1), also written as 2^(11-1) is equal to 2^(10) and has a value of
2 to the power of (12 minus 1), also written as 2^(12-1) is equal to 2^(11) and has a value of
2 to the power of (13 minus 1), also written as 2^(13-1) is equal to 2^(12) and has a value of
2 to the power of (14 minus 1), also written as 2^(14-1) is equal to 2^(13) and has a value of
2 to the power of (15 minus 1), also written as 2^(15-1) is equal to 2^(14) and has a value of
2 to the power of (16 minus 1), also written as 2^(16-1) is equal to 2^(15) and has a value of
2 to the power of (17 minus 1), also written as 2^(17-1) is equal to 2^(16) and has a value of
2 to the power of (18 minus 1), also written as 2^(18-1) is equal to 2^(17) and has a value of
2 to the power of (19 minus 1), also written as 2^(19-1) is equal to 2^(18) and has a value of
2 to the power of (20 minus 1), also written as 2^(20-1) is equal to 2^(19) and has a value of
2 to the power of (21 minus 1), also written as 2^(21-1) is equal to 2^(20) and has a value of
2 to the power of (22 minus 1), also written as 2^(22-1) is equal to 2^(21) and has a value of
2 to the power of (23 minus 1), also written as 2^(23-1) is equal to 2^(22) and has a value of
2 to the power of (24 minus 1), also written as 2^(24-1) is equal to 2^(23) and has a value of
2 to the power of (25 minus 1), also written as 2^(25-1) is equal to 2^(24) and has a value of
2 to the power of (26 minus 1), also written as 2^(26-1) is equal to 2^(25) and has a value of
2 to the power of (27 minus 1), also written as 2^(27-1) is equal to 2^(26) and has a value of
2 to the power of (28 minus 1), also written as 2^(28-1) is equal to 2^(27) and has a value of
2 to the power of (29 minus 1), also written as 2^(29-1) is equal to 2^(28) and has a value of
2 to the power of (30 minus 1), also written as 2^(30-1) is equal to 2^(29) and has a value of
2 to the power of (31 minus 1), also written as 2^(31-1) is equal to 2^(30) and has a value of
2 to the power of (32 minus 1), also written as 2^(32-1) is equal to 2^(31) and has a value of
2 to the power of (33 minus 1), also written as 2^(33-1) is equal to 2^(32) and has a value of
2 to the power of (34 minus 1), also written as 2^(34-1) is equal to 2^(33) and has a value of
2 to the power of (35 minus 1), also written as 2^(35-1) is equal to 2^(34) and has a value of
2 to the power of (36 minus 1), also written as 2^(36-1) is equal to 2^(35) and has a value of
2 to the power of (37 minus 1), also written as 2^(37-1) is equal to 2^(36) and has a value of
2 to the power of (38 minus 1), also written as 2^(38-1) is equal to 2^(37) and has a value of
2 to the power of (39 minus 1), also written as 2^(39-1) is equal to 2^(38) and has a value of
2 to the power of (40 minus 1), also written as 2^(40-1) is equal to 2^(39) and has a value of
2 to the power of (41 minus 1), also written as 2^(41-1) is equal to 2^(40) and has a value of
2 to the power of (42 minus 1), also written as 2^(42-1) is equal to 2^(41) and has a value of
2 to the power of (43 minus 1), also written as 2^(43-1) is equal to 2^(42) and has a value of
2 to the power of (44 minus 1), also written as 2^(44-1) is equal to 2^(43) and has a value of
2 to the power of (45 minus 1), also written as 2^(45-1) is equal to 2^(44) and has a value of
2 to the power of (46 minus 1), also written as 2^(46-1) is equal to 2^(45) and has a value of
2 to the power of (47 minus 1), also written as 2^(47-1) is equal to 2^(46) and has a value of
2 to the power of (48 minus 1), also written as 2^(48-1) is equal to 2^(47) and has a value of
2 to the power of (49 minus 1), also written as 2^(49-1) is equal to 2^(48) and has a value of
2 to the power of (50 minus 1), also written as 2^(50-1) is equal to 2^(49) and has a value of
```

1.3 Exercise 3 (6 Marks)

Write a program that finds the sum of the first 1000 even integers. Not the evens from 1 to 1000, but the first **1000** even integers. Also calculate the average of the 1000 even integers. Print the length of the list, the sum of the list, the average of the list and the last 5 even numbers in the list.

```
In [3]: # creating an empty list for my even integers and initializing my count of even intege
        even_integers = []
        even_integers_count = 0

        # using a while loop that will run until my list of even integers contains 1000 items
        while len(even_integers) <= 999:
            # using an if statement to check whether the current integer is even or not
            if even_integers_count % 2 == 0:
                # appending even integers to my list
                even_integers.append(even_integers_count)
            # incrementing my while loop to the next integer
            even_integers_count = even_integers_count + 1


        # printing the length of the list of even integers
        print("The length of the list of even integers is {0:,g}".format(len(even_integers)) +

        # calculating the sum of my list of even integers and printing out the sum.
        sum_even_integers = sum(even_integers)
        print("The sum of the first {0:,g}".format(len(even_integers))
              + " even integers is equal to {0:,g}".format(sum_even_integers) + ".\n")

        # calculating the average of my list of even integers and printing out the average.
        avg_even_integers = sum_even_integers / float(len(even_integers))
        print("The average of the first {0:,g}".format(len(even_integers))
              + " even integers is equal to {0:,g}".format(avg_even_integers) + ".\n")

        # printing out the last 5 numbers in the list
        print("The last five numbers in the list are: ", even_integers[-5:])
```

```
The length of the list of even integers is 1,000.

The sum of the first 1,000 even integers is equal to 999,000.

The average of the first 1,000 even integers is equal to 999.

The last five numbers in the list are:  [1990, 1992, 1994, 1996, 1998]
```