Computing

Lesson 9: 2D Arrays and Lists

Programming Part 5: Strings and Lists

Rebecca Franks

Word play

Code snippet

Creates a 2D list and displays the item held in row 1 and column 2. The output will be 6

```
scores = [["Fred", "Wilma", "Dino"],
          [4, 5, 6]]

print(scores[1][2])
```

Task: Accessing lists within lists

Step 1

Here is a 2D list containing different types of animals. Each list (row) has a unique type of animal. Copy this code into your development environment to use with these tasks.

```
animals = [["Salmon", "Pollock", "Cod"],
           ["Parrot", "Duck", "Wren"],
           ["Camel", "Lion", "Tiger"]]
```

Step 2

Write a line of code that will print all of the birds.

Step 3

Write a line of code that will print all of the mammals.

Task: Accessing single items within 2 Dimensional lists

Step 1

Using the same 2 Dimensional list, write a line of code that will print `Wren` as output.

Step 2

Write a line of code that will print `Pollock` as output.

Step 3

Write a line of code that will print `Camel` as output.

Task: Select a random item

Step 1

Import the random integer function from the random library.

Step 2

Write some code that will select a random animal from the animals list.

Hint: You will need to generate two random numbers from within the list index range.

Step 3

Make sure that you test your code a number of times to ensure that you have selected the correct index range.

Password manager

Code snippet 1

Creates a 2D list and then changes 1 item in that list

```
animals = [["Salmon", "Pollock", "Cod"],
           ["Parrot", "Duck", "Wren"],
           ["Camel", "Lion", "Tiger"]]

animals[0][2] = "Plaice"
```

Code snippet 2

Creates a 2D list and appends an item to the list at location 0

```
animals = [["Salmon", "Pollock", "Cod"],
           ["Parrot", "Duck", "Wren"],
           ["Camel", "Lion", "Tiger"]]

animals[0].append("Trout")

print(animals)
```

Code snippet 3

Creates a list and continuously appends to the list until N is entered

```
words = []
words_left = True
while words_left:
    print("Enter a word")
    word = input()
    words.append(word)
    print("Would you like to enter another word? Y/N")
    answer = input().upper()
    if answer == "N":
        words_left = False

print(words)
```

Code snippet 4

Finds the index location of an entered item

```
1  print("Which word would you like to find?")
2  word = input()
3  location = (words[0].index(word))
```

Task: Populate the 2D list

Step 1

Use the code below as your starting point for the 2D passwords list. This creates a 2D list that holds three empty lists. These will be populated in the next step.

```
passwords = [[],
             [],
             []]
```

Step 2

The incomplete code below should add the item Raspbian to the first list in the 2D list. Complete it and test it out by printing the list underneath.

```
    passwords           ("Raspbian")
```

Task: Populate the 2D list

Step 3

Write 2 more lines of code that will:

Add the username pi to the list at location 1

Add the password raspberry to the list at location 2

Step 4

Test your code by printing the passwords list and checking if each list now has a new item. It should look like the output below:

```
[['Raspbian'], ['pi'], ['raspberry']]
```


Task: Iteratively populate the 2D list

Step 1

Incorporate your working code into a while loop. Your program should:

- Continue to ask the user if they would like to enter a new password.
- If they say yes it should prompt for the account, username and password
- If they say no then the loop should terminate
- After the loop has terminated it should print the entire 2D list

Tip: use the code snippet on the first page to help you with this.

Task: Access an account username and password

Step 1

Incorporate extra functionality to your program. After the user has entered all of their passwords it should no longer print them at the end.

It should:

- Ask the user which account they would like to reveal the password for
- Allow the user to enter the account name
- Search for the account name in the account list (list 0)
- Reveal the username and password for that account based on the found location

Use the input/output example table on the following slides to help you test your program.

Task: Access an account username and password

Example

Note: Given the input you see in this sample interaction, this is the output your program should produce.

The user is given a prompt.	`Enter the account`
The user enters the account name.	`raspbian`
The user is given a prompt.	`Enter the username`
The user enters the username.	`pi`
The user is given a prompt.	`Enter the password`
The user enters the password.	`raspberry`

Continues on next slide...

Task: Access an account username and password

The user is given a prompt	`Would you like to enter another account? Y/N`
The user enters a response.	`n`
The response was n so the loop terminates and gives the user a prompt.	`Which account would you like to see the password for?`
The user enters the account they wish to view.	`raspbian`

The program searches for the account and finds the index location. This is then used to provide output from the other lists.	`Account: raspbian`
	`Username: pi`
	`Password: raspberry`