Chapter 4.2: Data Types & Data Structures
Arrays
1 of 5

Objectives

- Understand the term Array
- Be able to explain the advantages of an Array
- Be able to create and use an Array in your code

Key Terms

- Array
- Append
- Index
- Static
- Dynamic

<table>
<tr><th colspan="6">Computer Progression Pathways Codes</th></tr>
<tr><th>Algorithms</th><th>Programming & Development</th><th>Data & Data Representation</th><th>Hardware & Processing</th><th>Communication & Networks</th><th>Information Technology</th></tr>
<tr><td>A1</td><td>P1</td><td>D1</td><td>H1</td><td>C1</td><td>I1</td></tr>
<tr><td>A2</td><td>P2</td><td rowspan="2">D2</td><td rowspan="2">H2</td><td>C2</td><td>I2</td></tr>
<tr><td rowspan="2">A3</td><td>P3</td><td rowspan="2">C3</td><td>I3</td></tr>
<tr><td>P4</td><td>I4</td></tr>
<tr><td>A4</td><td>P5</td><td>D3</td><td>H3</td><td>C4</td><td>I5</td></tr>
<tr><td>A5</td><td>P6</td><td>D4</td><td>H4</td><td rowspan="4">C5</td><td>I6</td></tr>
<tr><td>A6</td><td rowspan="2">P7</td><td rowspan="2">D5</td><td rowspan="2">H5</td><td>I7</td></tr>
<tr><td rowspan="2">A7</td><td>I8</td></tr>
<tr><td>I9</td></tr>
<tr><td>A8</td><td>P8</td><td>D6</td><td>H6</td><td>C6</td><td>I10</td></tr>
<tr><td>A9</td><td>P9</td><td>D7</td><td rowspan="2">H7</td><td>C7</td><td>I11</td></tr>
<tr><td>A10</td><td>P10</td><td>D8</td><td>C8</td><td>I12</td></tr>
<tr><td>A11</td><td>P11</td><td>D9</td><td>H8</td><td>C9</td><td>I13</td></tr>
<tr><td>A12</td><td>P12</td><td rowspan="3">D10</td><td>H9</td><td>C10</td><td>I14:</td></tr>
<tr><td rowspan="2">A13</td><td>P13</td><td rowspan="2">H10</td><td rowspan="2">C11</td><td>I15</td></tr>
<tr><td>P14</td><td>I16</td></tr>
<tr><td>A14</td><td>P15</td><td>D11</td><td>H11</td><td>C12.</td><td>I17</td></tr>
<tr><td>A15</td><td>P16</td><td>D12</td><td>H12</td><td>C13</td><td>I18</td></tr>
<tr><td>A16</td><td>P17</td><td>D13</td><td rowspan="4">H13</td><td rowspan="4">C14</td><td rowspan="4">I19</td></tr>
<tr><td rowspan="3">A17</td><td rowspan="3">P18</td><td>D14</td></tr>
<tr><td>D15</td></tr>
<tr><td>D16</td></tr>
<tr><td>A18</td><td>P19</td><td>D17</td><td>H14</td><td>C15</td><td>I20</td></tr>
<tr><td>A19</td><td>P20</td><td>D18</td><td rowspan="4">H15</td><td rowspan="4">C16</td><td>I21</td></tr>
<tr><td rowspan="3">A20</td><td>P21</td><td>D19</td><td>I22</td></tr>
<tr><td>P21</td><td rowspan="3">D20</td><td rowspan="3">I23</td></tr>
<tr><td>P22</td></tr>
<tr><td>P23</td></tr>
<tr><td>A21</td><td>P24</td><td>D21</td><td rowspan="4">H16</td><td>C17</td><td>I24</td></tr>
<tr><td>A22</td><td>P25</td><td>D22</td><td>C18</td><td>I25</td></tr>
<tr><td>A23</td><td>P26</td><td rowspan="3">D23</td><td rowspan="3">C19</td><td>I26</td></tr>
<tr><td>A24</td><td rowspan="2">P27</td><td>I27</td></tr>
<tr><td>A25</td><td>I28</td></tr>
<tr><td>A26</td><td>P28</td><td>D24</td><td>H17:</td><td rowspan="3">C20</td><td rowspan="3">I29</td></tr>
<tr><td rowspan="2">A27</td><td>P29</td><td>D25</td><td>H18</td></tr>
<tr><td>P30</td><td>D26</td><td>H19</td></tr>
</table>

Chapter 4.2: Data Types & Data Structures
Arrays
2 of 5

Introduction
So far we have created variables and assigned a value to that variable. We have done stuff like:

```
firstName = Dylan (string)
DOB = 19/10/1974 (string)
Male = True (boolean)
```

Variables are useful to store one item of data, but what if I wanted to store the details of lots of people? I could use an **ARRAY**! There are other ways as well. But in this section we are going to look at ARRAYS.

So if I were just using variables and wanted to store three different first names I would need to do something like this:

```
firstNameOne = Dylan
firstNameTwo = Sally
firstNameThree = Sofia
```

This would work fine, but what if I wanted to use the program later and forgot how many names I were using or I did not know how many names I was starting off with. Well there is a solution and that solution is an Array.

Arrays
An array is a data structure that groups all related data under one identifier (name). In the example above we have used three identifiers (firstNameOne, firstNameTwo and firstNameThree).

A **static** array is a fixed size which means when you create the array you must specify the number of items it will hold and if you want to store more data it will not let you as it will be full, the size of a static array must be stated when you create it in your code. A **dynamic** array will adjust to let you add more items if required.

Example Static Array

```
Friends [5]
```

This will create a static array that will allow you to store five friends under the identifier name Friends.

Index

The index is the number of the item. If you wanted to print the third item in the array you could write.

```
print(Friends[3])
```

Pseudocode Example

```
array answers[10]
for index = 0-9
    response =  input("Please enter answer" +
    index +1)

    answers[index] = response

next index
print ("Test finished.")
```

Task

Code a quiz in Python your code must ask the user 10 questions and store their answers in an Array. Screen shot your COMMENTED code and stick in the box below. Make sure you crop and resize your image. Look down

Chapter 4.2: Data Types & Data Structures
Arrays
4 of 5

```
Question 6
Which of these is a data type?

a.... Statement
b.... Real
c.... Selection

Make your choice >>>>   b
Correct!
Question 7
What is a constant?

a.... A name assigned to a value that can be changed whilst the program is running
b.... A variable that doesn't change
c.... A name assigned to a value that can't be changed

Make your choice >>>>   c
Correct!
Question 8
What are true and false represented by in Boolean?

a.... T and F
b.... y and n
c.... 0 and 1

Make your choice >>>>   c
Correct!
Question 9
How should you name a variable with multiple words?

a.... Camel case.
b.... Underscores
c.... Just use the space bar...

Make your choice >>>>   a
Correct!
Question 10, Final Question
What is the difference between = and ==

a.... One of them can assign items to things whilst the other checks if two values are equal
b.... One of them doesn't do anything
c.... They both check if two values are equal. There's no difference between = and ==

Make your choice >>>>   a
Correct!
```

Link at bottom

```
    print("Question 9")
    print("How should you name a variable with multiple words?")
    print()
    print("a.... Camel case. ")
    print("b.... Underscores")
    print("c.... Just use the space bar...")
    print()
    answer = input("Make your choice >>>>  ")
    if answer == "a" or answer == "b":
        print("Correct!")
        result = result + 1
    else:
        print("Incorrect")
    quizAnswers.append (answer)

    print("Question 10, Final Question")
    print("What is the difference between = and ==")
    print()
    print("a.... One of them can assign items to things whilst the other checks if two values are equal")
    print("b.... One of them doesn't do anything")
    print("c.... They both check if two values are equal. There's no difference between = and ==")
    print()
    answer = input("Make your choice >>>>  ")
    if answer == "a":
        print("Correct!")
        result = result + 1
    else:
        print("Incorrect")
    quizAnswers.append (answer)

    print("Hope you enjoyed the quiz. You scored", result, "out of 10")
    print("Here are for no reason your answers", quizAnswers)

    with open("quiz.txt", mode="w",encoding="utf-8") as my_file:
        for elements in quizAnswers:
            my_file.write(elements+"\n")

quiz()
```

https://repl.it/@ManrajLally/Quiz-Chapter-4 Go here for it all

So on my quiz I made a pretty basic one but also created ‘quizAnswers’. At the end of each question, the answer that was input was added to quizAnswers using append. At the end I printed quizAnswers to check it worked. At the bottom I also made it so

quizAnswers was saved to a text file. For fun I added a variable that increased if you got the answer correct and I printed it at the end so the user could see their score out of 10.

Resources

BBC - http://www.bbc.co.uk/education/guides/zy9thyc/revision
Python School - https://pythonschool.net/basics/lists-functions-and-files/