UNIVERSITY OF Hull

Lists

Wrestling with Python
Using Lists

University of Hull

The Cinema App So Far

Summary

- This is fine for this data set, but it's not very flexible

Lists

What we can do so far…

- Store data (using variables)
- Change data (using expressions)
- Make decisions (using conditions)
- Create loops (using while and for)
- There is not much more that we need to know how to do
  - But we do need to know how to create lists

Variables

- We have a reasonable idea of how to create a variable:

```
sales = 10
```

- This will create a variable which can hold a single integer value
- The variable has the identifier **`sales`**
- The variable holds the value **`10`**

Handling more data

- If we want to store more data, the simplest approach is to create more variables:

```
sales1 = 5
sales2 = 10
sales3 = 0
sales4 = 30
```

Manipulating data

- However, this makes the data hard to work with:

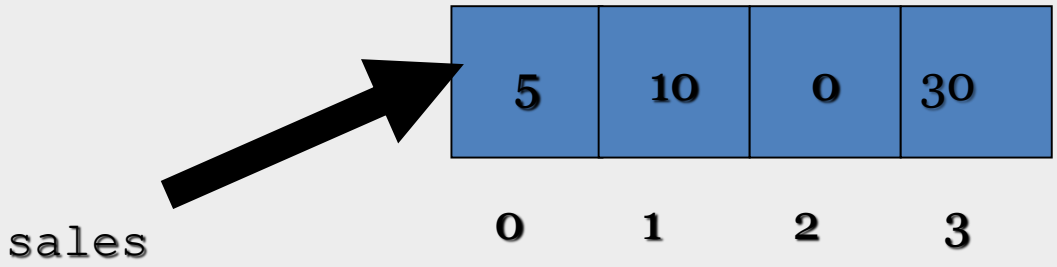
```
if (sales1 > sales2 and
    sales1 > sales3 and
    sales1 > sales4 )
{
   print (sales1)
}
```

Manipulating data

```
if(sales1 > sales2 and sales1 > sales3 and
sales1 > sales4):
     print(sales1)
elif(sales2 > sales1 and sales2 > sales3 and
sales2 > sales4):
     print(sales2)
elif(sales3 > sales1 and sales3 > sales2 and
sales3 > sales4):
     print(sales3)
else:
     print(sales4)
```

Lists

- An list lets us store more than one variables which we can *index* using a *subscript*



- **`sales`** is a variable which contains 4 other variables

Creating a list

- You can create an empty list

```
salesList = []
```

```
salesList = list()
```

- Or you can specify some data in the list

```
salesList = [5, 10, 0, 30]
```

Appending a list

- You can create an empty list

```
>>>salesList = [5, 10, 0, 30]
>>>salesList.append(6)
>>>salesList
[5, 10, 0, 30, 6]
```

- Or you can specify some data in the list

Accessing a list

- The value in the square brackets is called a *subscript*
- Note that the initial element has a subscript of `0`

```
>>> salesList = [5, 10, 0, 30]
>>> salesList [0]
5
>>> salesList [2]
0
```

Subscripts Etiquette

- Subscripts start at 0
- If you try to access an element which is not in the list (perhaps by using a subscript which is too large) your program will fail
- Subscripts should be checked as your program runs so that our programs never "fall off the end of a list"

Finding the length of a list

- You can find the length of a list like this:

```
>>> salesList = [5, 10, 0, 30]
>>> len(salesList)
4
```

Other things lists can do

- There are lots of other things that lists can do:
- insert, remove, clear, sort, reverse etc
- http://docs.python.org/3.3/tutorial/datastructures.html#more-on-lists

The power of lists

- Lists are great for storing lots of data

```
salesList = []
for i in range(4):
      salesString = input( “Enter sales” )
      salesList.append(int(salesString))
```

- This will read in 4 values, but it would work just the same if we had 1000 values to enter

Lists don't have a fixed size

- We don't even need to know how many values we will need to enter when we write our code

```
salesList = []
moreValues = True

While (moreValues):
      salesString = input("Enter sales")
      salesList.append(int(salesString))
      if(input( "More values? y/n" ) == "n"):
            moreValues = False
```

The power of subscripts

- Subscripts become very powerful when we discover that we can use a variable as a subscript:

```
maxSales = 0
for i in range(4):
    if (salesList[i] > maxSales):
        maxSales = salesList[i]
```

- This will find the largest sales value from the first 4 elements in the list

Why the length of a list is useful

- If we don't know how many elements the list has – no problem!

```
maxSales = 0
for i in range(len(salesList)):
    if (salesList[i] > maxSales):
        maxSales = salesList[i]
```

- This will find the largest sales value in the list

A list is iterable

```
maxSales = 0
for i in salesList:
    if (i > maxSales):
        maxSales = i
```

- As a list is an iterable type we can use it in to control a for loop
- i will be assigned each of the values in the list

Summary

- Lists are the last thing that we need to know how to write every program in the world
- The allow us to store huge amounts of data and search and sort it
- The key to the power of a list is the use of variables as subscripts