

University of Hull
Department of Computer Science

Playing with Python

Vsn. 1.0 Rob Miles 2014

Cricket Scores and Lists

Practical Break 1: Processing Cricket Player Scores

In this lab we are going to use a list to store the cricket scores.

Creating a New Program

We are going to make a new version of the cricket scores program. If you can find your old one you can use that as the basis of this work, otherwise you can start from scratch.



Before you go any further; perform the following:

1. Start the Idle environment and create a new program called cricketList or open your old cricket program.

Creating a list

For each batsman we are going to store their score. We are going to store each of the score a list we could call `scoreList`:

```
scoreList = []
```



Before you go any further; perform the following:

2. Add the above line to your program to create an empty list.

Every member of the team must bat, so 11 scores will be entered into the program. You will need to loop round 11 times. We can use a for loop for this.

```
for i in range(1,12):
```

```
    # go round this loop 11 times - once for each score value
```

The loop must read a score value from the user and then append it to the scores list.



Before you go any further; perform the following:

3. Add the loop above, and add the statements to the loop that will read in the score value and then add it to the list. You can refer back to previous programs that have read numbers from users if you are not sure how to do this. You will find the `input` and `int` methods worth examining.

Printing out the scores

We can use the for loop to work through the elements in the list and print them out.

```
for score in scoreList:
```

```
    print(score)
```



Before you go any further; perform the following:

4. Add the loop above which will print out the scores once they have been entered.
5. Test your program to make sure that it will work.

Score Data Processing

We can use this loop format to work through the data as many times as we like.



Before you go any further; perform the following:

6. Add another loop that will go through the list and find the highest score.

Practical Break 2: Starting Sorting

Computers are very good at sorting data. Now we can consider how they do it.

```
if scoreList[0] > scoreList [1]:
```

```
    temp=scoreList[0]
```

```
    scoreList[0] = scoreList[1]
```

```
    scoreList[1] = temp
```

The above code will make sure that the first two elements in the list are in the right order. So, once the code has run the list will probably be in the wrong order, but it will be a bit less wrong. Note how it uses a temp variable to hold the value during the swap.

```
if scoreList[i] > scoreList [i+1]:
```

```
    temp=scoreList[i]
```

```
    scoreList[i] = scoreList[i+1]
```

```
    scoreList[i+1] = temp
```

This is a development of the first code, in that it makes sure that two adjacent items are in the right order. If we had a loop that took the value of i from 0 to 9 (don't take i to 10 as this will cause problems – see if you can figure out why) then we could work through the array swapping numbers that were the wrong way round.

This would still not completely sort the array, but if we made lots of passes (10 would be guaranteed to work) through the data we would end up with the list completely sorted. This is called the Bubblesort method of sorting numbers.

We can use this loop format to work through the data as many times as we like.



Before you go any further; perform the following:

1. Add code to your cricket scores program to sort the scores into ascending order.