python_selecting_actions_elif_workbook_answers

September 17, 2020

1 PYTHON SELECTING ACTIONS ELIF STATEMENTS WORKBOOK ANSWERS

Remember, there are different ways to write code to get the same answer, so your answer can be correct and different to the answer example!

If you feel stuck and want some in person help, then have a look at the events page to join in a workshop https://swamphen.co.uk/events.

```
In [ ]: # length of time these singles were in the charts
        crazy = 14
        save = 6
        play = 8
        another = 10
        flash = 13
```

```
In [ ]: # put these values into a tuple
        time = (crazy, save, play, another, flash)
```

```
In [ ]: # what is the maximum value in this tuple?
        max(time)
```

```
In [ ]: # using your newly created tuple calculate if
        # play the game the single that spent the least or most amount of time in the charts?
        if play == min(time):
            print('play the game spent the least amount of time in the charts')
        elif play == max (time):
            print('play the game spent the most amount of time in the charts')
        else:
            print('play the game did not spend the most or least amount of \
            time in the charts')
```

```
In [ ]: # do the same calculation, but this time with the song save me
        if save == min(time):
            print('save me spent the least amount of time in the charts')
        elif save == max (time):
            print('save me spent the most amount of time in the charts')
        else:
            print('save me did not spend the most or least amount of time \
            in the charts')
```

```
In [ ]: # write an if else statement to work out if the time spent in the charts is divisible
        if play % 4 == 0:
            print('time in charts is divisible by 4')
        else:
            print('time in charts is not divisible by 4')
```

```
In [ ]: # sort the singles into 3 categories
        # 6 and under, 8-12, 13 + weeks
        low = []
        middle = []
        high = []
```

```
In [ ]: single = flash

        if single <= 6:
            low.append(single)
        elif single <= 12:
            middle.append(single)
        else:
            high.append(single)

        print(low, middle, high)
```