From lecture 4

Use the len() function to calculate how many characters are in a string
Obtain a single character from a string
Slice strings
Concatenate strings

```python
words = " Prince Charming ">
length = len(words)

letter1 = words[3]
letter2 = words[-5]
letter3 = words[len(words) - 2]

letters1 = words[3:6]
letters2 = words[:6]
letters3 = words[6:]
letters4 = words[-3:]

word = letter1 + letter2
word = word + " " + letter3

print(letters1, letters2, letters3, letters4, word)
```
String methods – find()

The **find**() method is used to look for the position (index number) of the first occurrence (from the left) of some characters. If the characters are found, the find() method returns the index number, otherwise the find() method returns -1.

For example,

```python
greeting = "Hello World"
position1 = greeting.find(" ")
position2 = greeting.find("z")
position3 = greeting.find("orl")
print(position1, position2, position3)
```

```
5 -1 7
```
The `rfind()` method is used to look for the index position of the last occurrence (from the right) of some characters. If the characters are found, the `rfind()` method returns the index number, otherwise the `rfind()` method returns -1.

```
greeting = "Hello World"
position1 = greeting.find("o")
position2 = greeting.rfind("o")
position3 = greeting.rfind("orl")
position4 = greeting.rfind("lro")
print(position1, position2, position3, position4)
```

Output:
```
4 7 7 -1
```
Exercise

Complete the following program so that it prints the initial from the first name followed by a full stop, a space and followed by the surname. Assume the full name is always two names separated by a single space.

```python
full_name = "Wystan Auden"
first_letter = full_name[0]
space_index = full_name.find(" ")
last_name = full_name[space_index + 1:]
initialled_name = first_letter + ". " + last_name
print(initialled_name)
```

W. Auden

Funeral Blues: Stop all the clocks, cut off the telephone/ Prevent the dog from barking ...
Common Python inbuilt functions

`min()` is an **inbuilt function** which can be used to find the smallest number from a comma separated set of numbers and `max()` is the **inbuilt function** which can be used to find the largest number from a comma separated set of numbers, e.g.,

```python
num1 = 32
num2 = 16
smallest = min(num1, num2)
print(smallest)

smallest = min(32.7, 56.4, 3, -1.1, 56.99, -1.2)
print(smallest)

largest = max(num1, num2)
print(largest)

largest = max(32.7, 56.4, 3, -1.1, 56.99, -1.2)
print(largest)
```
Common Python inbuilt functions

The **inbuilt function** `abs()`, is used to get the absolute value (the magnitude) of a number (int or float), e.g.,

```python
num1 = 32
num2 = -32
num3 = abs(16 - 23)

print("1.", abs(num1))
print("2.", abs(num2))
print("3.", num3)
print("4.", abs(-16.78))
```

1. 32
2. 32
3. 7
4. 16.78
Complete the following program so that it prints the total tax and the net pay rounded to a whole number. The first $14000 is not taxed. The next amount up to $38500 is taxed at 24% and the rest is taxed at 34%.

```
salary = 54000
no_tax_boundary = 14000
rate1_boundary = 38500
rate1 = 0.24
rate2 = 0.34
rate2_amount = salary - rate1_boundary
rate1_amount = rate1_boundary - no_tax_boundary
rate2_tax = round(rate2_amount * rate2)
rate1_tax = round(rate1_amount * rate1)
total_tax = rate2_tax + rate1_tax
net_pay = salary - total_tax
print("Salary: $",salary,sep="")
print("Amount to be taxed at: ",round(rate1*100),"%: $",rate1_amount,sep="")
print("Tax at rate1: $",rate1_tax,sep="")
print("Amount to be taxed at: ",round(rate2*100),"%: $",rate2_amount,sep="")
print("Tax at rate2: $",rate2_tax,sep="")
print("=============================")
print("Total tax: $",total_tax, sep = "")
print()
print("Net pay: $", net_pay, sep = "")
print("=================================")
```

**Salary:** $54000  
**Amount to be taxed at:** 24%: $24500  
**Tax at rate1:** $5880  
**Amount to be taxed at:** 34%: $15500  
**Tax at rate2:** $5270  
**Total tax:** $11150  
**Net pay:** $42850  
**================================**