INTRODUCTION TO PYTHON ALGORITHMS USING PYTHON

CHARIS CODING CLUB

WWW.CHARISCODINGCLUB.COM

COURSE OUTLINE

• LESSON1:	INTRODUCTION TO ALGORITHMS, REFRESHER, GIT
• LESSON2:	TACKLING ALGORITHMIC PROBLEMS – EUREKA! CHALLENGE
• LESSON3:	STRING METHOD AND STRING CHALLENGE
• LESSON4:	NEW PYTHON METHODS FOR SCRIPTING
• LESSON5:	CHALLENGE 3 AND 4
• LESSON6:	FUNCTIONS AND CLASSES
• LESSON7:	CHALLENGE 5 AND 6
• LESSON8:	SIMPLE GAME
• LESSON9:	PROJECT PART 1
• LESSON10:	PROJECT PART 2

LESSON 4:
NEW PYTHON METHODS FOR SCRIPTING

OBJECTIVES:

PROVIDE SOLUTIONS TO THE PREVIOUS TWO EXERCISES

COMPARING METHODOLOGIES

NEW METHODS

SOLUTIONS TO OUR PREVIOUS TASKS

- NOW, LET’S START BY SOLVING OUR FIRST TWO CHALLENGES AND SEEING HOW WE HAVE USED ALGORITHMS TO SOLVE PROBLEMS.

THE EUREKA! CHALLENGE

The volume of an object is equal to the volume of the water displaced when this object is submerged.

In the Eureka challenge, our task was to find which material a piece of metal (in this case, a crown) was made of.

UNDERSTANDING THE REQUIREMENTS:

FORTUNATELY, THE REQUIREMENTS WERE CLEARLY STATED. WHICH WERE:

1. Ask the user to enter the mass of an object. (in Kg)
2. Ask the user to enter the volume of an object. (in m^3)
3. Calculate and output the density of this object (mass/volume)
4. Compare this density with the densities of different metals such as pure gold, silver and bronze and output the corresponding metal if there is a match.

The first and second requirements were to ask the user to input the mass and density of the object.
Luckily, this was given to us in the code snippet we copied.

Computing The Density

- The first thing we do is to compute the density which is mass divided by volume. Note that we have converted inputs from strings to floats

```
# Eureka! Archimedes and King Hiero's Crown

mass = float (input("Input the mass of the crown in kg: "))
volume = float (input("Input the volume of the crown in cubic meters:"))

# Compute the density of the imputed values

density = mass / volume
```

Now we have the computed density and can continue with the rest of our code.

Third Requirement: Comparing Densities

- The next thing we do is to compare the densities, let’s start with Aluminum. Since the test is a range, we have to take care boundary conditions.

```
# Compute the density of the imputed values

density = mass / volume

# Test between boundary conditions

if density >= 24000 and density <= 27000:
    print('Density is '  + str(density) + 'kg/m3. This metal is Aluminum')
```

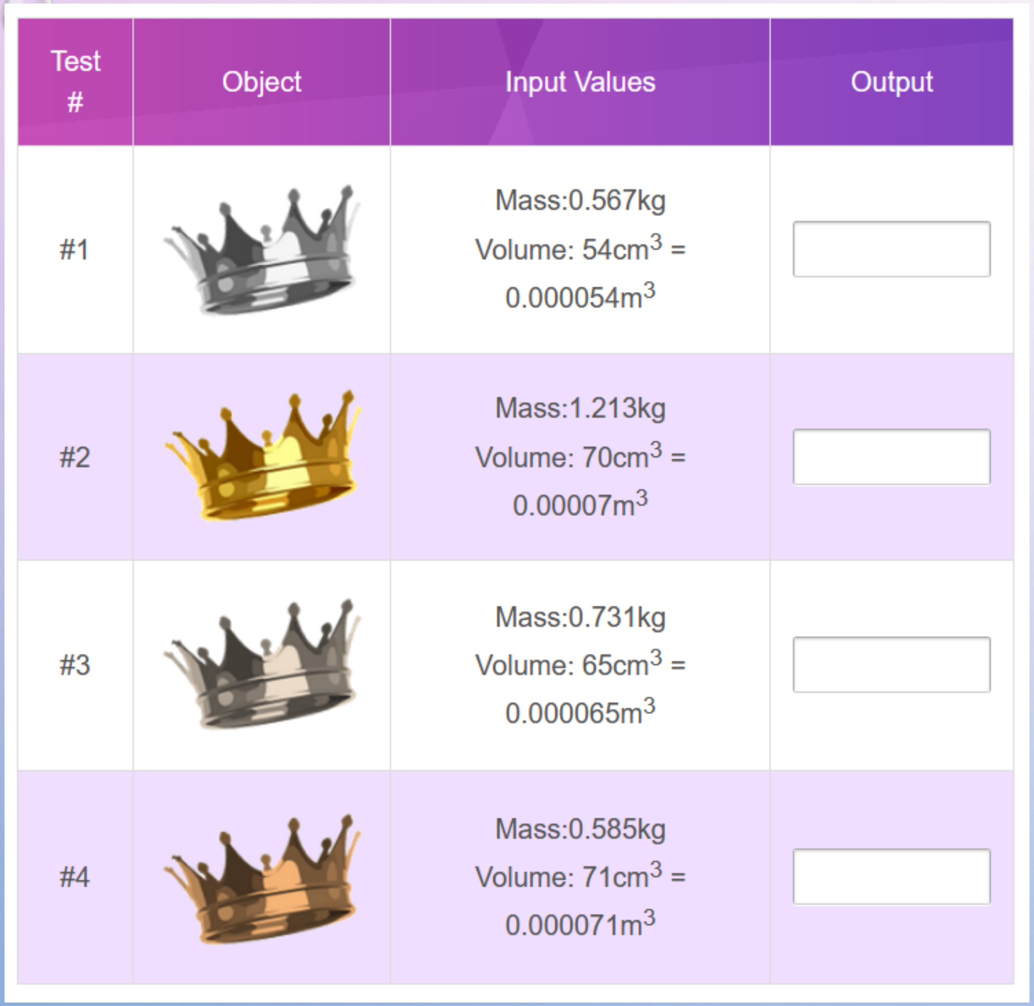
Third Requirement: Comparing Densities (Contd.)

We then have to make the comparison for all the metals in the given.
A good way to do this is using the elif control flow we learned before.

We then have to make the comparison for all the metals in the given.
A good way to do this is using the elif control flow we learned before as shown in the next slide:

```python
# Test between boundary conditions

if density >= 24000 and density <= 27000:
    print('Density is '  + str(density) + 'kg/m3. This metal is Aluminum')

elif density >= 8100 and density <= 8300:
    print('Density is '  + str(density) + 'kg/m3. This metal is Bronze')

elif density >= 10400 and density <= 10600:
    print('Density is '  + str(density) + 'kg/m3. This metal is Silver')

elif density >= 11200 and density <= 11400:
    print('Density is '  + str(density) + 'kg/m3. This metal is Lead')

elif density >= 17100 and density <= 17500:
    print('Density is '  + str(density) + 'kg/m3. This metal is Gold')

elif density >= 21000 and density <= 21500:
    print('Density is '  + str(density) + 'kg/m3. This metal is Platinum')
```

NOW WE CAN TEST CASES BY IMPUTING THE TEST CASES GIVEN



Test #	Object	Input Values	Output
#1		Mass:0.567kg Volume: 54cm^3 = 0.000054m^3	
#2		Mass:1.213kg Volume: 70cm^3 = 0.00007m^3	
#3		Mass:0.731kg Volume: 65cm^3 = 0.000065m^3	
#4		Mass:0.585kg Volume: 71cm^3 = 0.000071m^3	

Test them with your own code and state what you get for the following cases:

*1:
*2:
*3:
*4:

MY ANSWERS:

* Silver

* Gold

* Lead

* Bronze

Full code in the next slide

```python
# Eureka! Archimedes and King Hiero's Crown
mass = float (input("Input the mass of the crown in kg: "))
volume = float (input("Input the volume of the crown in cubic meters:"))
# Compute the density of the imputed values
density = mass / volume
# Test between boundary conditions
if density >= 24000 and density <= 27000:
    print('Density is '  + str(density) + 'kg/m3. This metal is Aluminum')
elif density >= 8100 and density <= 8300:
    print('Density is '  + str(density) + 'kg/m3. This metal is Bronze')
elif density >= 10400 and density <= 10600:
    print('Density is '  + str(density) + 'kg/m3. This metal is Silver')
elif density >= 11200 and density <= 11400:
    print('Density is '  + str(density) + 'kg/m3. This metal is Lead')
elif density >= 17100 and density <= 17500:
    print('Density is '  + str(density) + 'kg/m3. This metal is Gold')
elif density >= 21000 and density <= 21500:
    print('Density is '  + str(density) + 'kg/m3. This metal is Platinum')
else:
    print('Unknown material or invalid input.')
```

USERNAME VALIDATION CHALLENGE

07jFox_S
09kJohnson_S
11rTaylor_S
00pJones_T
00jOliver_A

In the validation challenge, our task was to perform validation so that only valid usernames pass such as the ones on the left side of this screen.

ONCE AGAIN, THE REQUIREMENTS ARE CLEARLY STATED:

- If the username is less than 6 characters long the program should ask the user to enter a valid username.
- If the username does not contain the character "_" it should also ask the user to enter a valid username.
- If the username is valid, the program should decide if the user is a member of staff or a student.
- If they are a student the programme should find out their year group.
- The program should also display the initial of the student as well as their last name.
- Finally the program should display whether the user is a Student, a Teacher or an Admin member of staff.

FROM THE REQUIREMENT, WE CAN BREAK DOWN THE TASK INTO TWO PARTS

- PART A: Deals with evaluating whether the username is valid or not.
- PART B: The part that outputs the information of the user

- Because Part A deals with conditions that must all pass, it’s a good candidate for if and elif statements.
- Part B, on the other hand, outputs statements independently and is therefore a good candidate for separate if statements.

Reading the input and starting control flow

Since we're reading the input, we would use the input function to receive the username.

```
# Getting username input

username = input("Enter a valid username: ")
```

Now we can proceed to check whether the input is valid by checking for the various conditions.

1. Checking for length

After getting the username input, we can check for length using the len() function. Using it in the code to check for appropriate length, we can write the following:

```
# Getting username input

username = input("Enter a valid username: ")

# Testing the conditions

if len(username) < 6:
    print('Your username must not be less than 6 characters. Enter a valid username')
```

2. Checking whether “_” is in the username

To check whether _ is in the username we use the *in* keyword which we learned in the previous class:

```
elif '_' not in username:
    print('The code at the end of your username must include underscore. Enter a valid username')
```

The keyword *not* is used to check the opposite of the condition described using *in*.

3. Checking whether the first two letters are numbers

To check whether the first two letters are numbers, we first slice the list and return the first two numbers. Remember that the result of trying to cast an incompatible type will product a null result.

```
elif int(username[:2]) is null:
    print('First two letters must be integer representing year. Enter a 
valid username.')
```

The keyword *not* is used to check the opposite of the condition described using *in*.

3. Checking whether the first two letters are numbers

To check whether the first two letters are numbers, we first slice the list and return the first two numbers. We then use isnumeric() method to check if it’s a number.

```
elif not username[:2].isnumeric():
    print('First two letters must be a number representing year. Enter a valid username.')
```

4. Checking whether the name and surname are numbers

Next we want to confirm that the username and surname are numbers. We already know that the first two characters are for the year, while the last two characters are an underscore and a letter.
Therefore, the initial and last name are contained in between. So we slice those two and check whether they are alphabets through the method we learned in the previous lesson.

```
elif not username[2:-2].isalpha():
    print('Username must have initial and lastname after year digits.
Enter a valid username’)
```

5. Checking a valid ending code

To check if the ending is valid, since we have already checked for the underscore "_" earlier, we need to check if the last letter is S or T or A. We do this with the following code:

```
elif username[-1] != 'S' or  username[-1] != 'T' or  username[-1] != 'A':
    print('Enter a valid code at the ending')
```

END OF PART A; NOW PART B:

Now we need to output the person's year group, the person's initial, surname, as well as whether the person is a member of staff.

For this to be possible, we first have to add a Boolean variable which will hold true or false. Let's call the variable valid.

We only want the code to execute if valid is true.

6. Add “valid” Boolean variable and use it to modify code

We add the variable valid and initialize it to true. If any of the conditions we did before fails, we put valid to false.

We do this for all the conditions we’ve written so far. The code below continues in the next slide:

```
username = input("Enter a valid username: ")

valid = True

# Testing the conditions

if len(username) < 6:
    print('Your username must not be less than 6 characters. Enter a
valid username')
    valid = False
```

```python
elif '_' not in username:
    print('The code at the end of your username must include underscore. Enter a valid username')
    valid = False

elif not username[:2].isnumeric():
    print('First two letters must be a number representing year. Enter a valid username.')
    valid = False

elif not username[2:-2].isalpha():
    print('Username must have initial and lastname after year digits. Enter a valid username')
    valid = False

elif username[-1] != 'S' or  username[-1] != 'T' or  username[-1] != 'A':
    print('Enter a valid code at the ending')
    valid = False
```

7. Writing the required outputs

Were almost done, all we need to do now, is add the valid condition and then inside it, put an if block to output the required information

```
if valid:
    if username[:2] != '00':
        print('Your year group is', username[:2])
```

Inside the if block which will run if the username remains valid, the first test we have done is to output the year group. This is only true for students whose first two letters is not ‘00’.

The first line will therefore only execute for students who have year at the beginning.

7b. Writing the required outputs

Still inside the if block, we then output the other conditions that will say if the person is a student, admin or teacher from the code. The result is shown below:

```
...
    if username[-2:] == '_S':
        print('Your are a student.\nYour surname is', username[3:-2],
'\nYour initial is', username[2])

    if username[-2:] == '_A':
        print('Your are an Admin member of staff.\nYour surname is',
username[3:-2], '\nYour initial is', username[2])

    if username[-2:] == '_T':
        print('Your are a student.\nYour surname is', username[3:-2],
'\nYour initial is', username[2])
```

END OF TASK! ☺

- Here's the full code in this slide and the next two slides:

```
# Get username input

username = input("Enter a valid username: ")

valid = True

# Testing the conditions

if len(username) < 6:
    print('Your username must not be less than 6 characters. Enter a valid 
username')
    valid = False
```

Full code contd:

```python
elif '_' not in username:
    print('The code at the end of your username must include underscore. Enter a valid username')
    valid = False

elif not username[:2].isnumeric():
    print('First two letters must be a number representing year. Enter a valid username.')
    valid = False

elif not username[2:-2].isalpha():
    print('Username must have initial and lastname after year digits. Enter a valid username')
    valid = False

elif username[-1] != 'S' or  username[-1] != 'T' or  username[-1] != 'A':
    print('Enter a valid code at the ending')
    valid = False
```

Full code continued:

```
if valid:
    if username[:2] != '00':
        print('Your year group is', username[:2])

    if username[-2:] == '_S':
        print('Your are a student.\nYour surname is', username[3:-2],
'\nYour initial is', username[2])

    if username[-2:] == '_A':
        print('Your are an Admin member of staff.\nYour surname is',
username[3:-2], '\nYour initial is', username[2])

    if username[-2:] == '_T':
        print('Your are a student.\nYour surname is', username[3:-2],
'\nYour initial is', username[2])
```

DONE FOR THE DAY…..

ALMOST!

CLASSWORK: MODIFYING THE SOLUTION

Now, a new system is being introduced that modifies the old one.

- Year Group Using two digits (e.g. “07” for year 7, “11” for year 11, “00” for staff members)
- 1 character for their initial (first letter of their firstname)
- The user’s lastname
- A 2-digit code: “_S” for students, “_T” for teachers, “_A” for admin staff.

The symbols above are same as before, but the arrangement is now different.

CLASSWORK: MODIFYING THE SOLUTION

In the new system the arrangement is different.

Last four digits contain the code and year now, so the following are the valid usernames

Rearranging the code

Invalid	Valid
07jFox_S	jFox_S07
09kJohnson_S	kJohnson_S09
11rTaylor_S	rTaylor_S11
00pJones_T	pJones_T00
00jOliver_A	jOliver_A00

ASSIGNMENT FINISH UP YOUR CODE

- TEST IT AND MAKE SURE ALL TEST CASES PASS AND OTHERS FAIL

QUESTION AND ANSWER TIME!

- ANY QUESTION?

GOOD LUCK WITH THE ASSIGNMENT!

SEND IN YOUR ANSWERS ON OR BEFORE **THURSDAY** NEXT WEEK