University of HUDDERSFIELD
Inspiring tomorrow's professionals

Software Design & Development CFS2160

Week 2 – Data Types

Session Plan

University of HUDDERSFIELD
Inspiring tomorrow's professionals

1. Reflect on last weeks session.
2. More on time management.
3. Strings and other things (data types).
4. Do a bit of coding.
5. Naming conventions.
6. Work on unfinished tutorial work.
7. Finish with questions?

Last Week

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Last week we talked about time management and took a gentle step into the world of programming in Python.

You should have now created the accounts as instructed by Tony, have a basic understanding of GitHub and written some simple code in Python anywhere.

You will required to use GitHub throughout the year.

If you have yet to do the above, please make the effort to do **AS SOON AS POSSIBLE**! Use your Guided Independent Study time.

Don't leave things until the last minute, time goes by very quickly at university and a lot of deadlines approach fast.

Remember

University of HUDDERSFIELD
Inspiring tomorrow's professionals

It's fine to not know something, that is why we study and practice. No one knows everything and the sooner we discuss things we don't know or understand, the sooner we can learn the solutions.

"Do today what you can do today, you will have something new to do tomorrow."

Data Types

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Most, if not all programming languages use the concept of Data Types. Simply put a Data Type describes the type of a piece of data.

Common data types found every day in programming are:

- Integer / int (a whole number, e.g. 456)
- String (a piece of text which is surrounded with speech marks, e.g. “Hello. World”)
- Boolean (True or False, it can only be true or false, not both, e.g. True)
- Float (a number with a decimal point, e.g. 3.14159265359)

Python Data Types

University of HUDDERSFIELD
Inspiring tomorrow's professionals

In programming, data is stored in a variable, such as this *steve_is_awesome = True*.

Once a variable is declared you can change is value and sometimes even its data type if needed. You may think *steve_is_awesome = False* is more correct, therefore you can change its value ☹.

Python is designed to automatically deal with data types, when declaring a variable it assumes and assigns the correct data type to it so you don't have to.

Read more on data types at the following website.

https://www.tutorialspoint.com/python3/python_variable_types.htm

Try it Out

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Open the PyCharm IDE, create a new python file (with a sensible name in a suitable folder) and code something like the following, use your own information.

```
name = "Steve"
email = "s.mcguire@hud.ac.uk"
age = 40
print(name)
print(email)
print(age)
print(type(name)) # type method gets the type of the data in the brackets
print(type(email))# print method prints whatever is in the brackets
print(type(age))
```

The print statement we saw last week, the type() function is where we ask Python what type a piece of data is, the code then just prints what type of data it is.

Try it Out - Result

University of HUDDERSFIELD
Inspiring tomorrow's professionals

The result will be similar to this

"C:\Program Files\Statistics\Python3\pythonw.exe" "K:/GTA Work/Programming/Repository/Python/Week 2/strings_and_things.py"

Steve
s.mcguire@hud.ac.uk
40
<class 'str'>
<class 'str'>
<class 'int'>

Don't forget to save the work with a useful file name and folder structure, don't complicate things, nice and simple will do fine. You can see my folder structure highlighted in blue above.

Using Operators

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Naturally, we can use operators to do things to our variable, they would be useless without them.

For numbers we can use all the usual mathematical operators.

For strings we can concatenate (join them together) using the + symbol.

```
number_1 = 2
number_2 = 10
name = "Steve"
email = "s.mcguire@hud.ac.uk"

print(number_1 + 2) #prints - 4
print(number_1 * number_2) #prints - 20
print("Name: " + name + ", Email: " + email) #prints - Name: Steve, Email: s.mcguire@hud.ac.uk
```

Python Naming Convention

University of HUDDERSFIELD
Inspiring tomorrow's professionals

There is a correct way to name your files and variables in Python.

Only use lowercase letters and use underscores for spaces.

e.g.

My File – This couldn't be more wrong, don't do it, please don't do it, **ever**!

week_1_strings – this is perfect, it tells us what the file contains and when we worked on it.

You will see I used this convention when naming steve_is_awesome earlier.

Get into the habit of naming your files and variables correctly, it will save you time in future.

Finally

University of HUDDERSFIELD
Inspiring tomorrow's professionals

1. Any questions on Python?
2. Continue working on unfinished programming tutorial work.
3. Ensure all your work has been committed and pushed to GitHub, this is vitally important for your future work.
4. Do not leave your work on the C:\ drive of the machine you have been working on.