Cyber Academy

Module 01

Session 2 –Python

10 November 2021

Taught by Michelle Sheppard

TTA

TECH TALENT ACADEMY

Session overview

- Resources /News
- Recap over last week
- HLTs
- Employability
- GitHub
- This week
  - Functions\procedures
  - Selection
  - Iteration
  - Cyber and Python

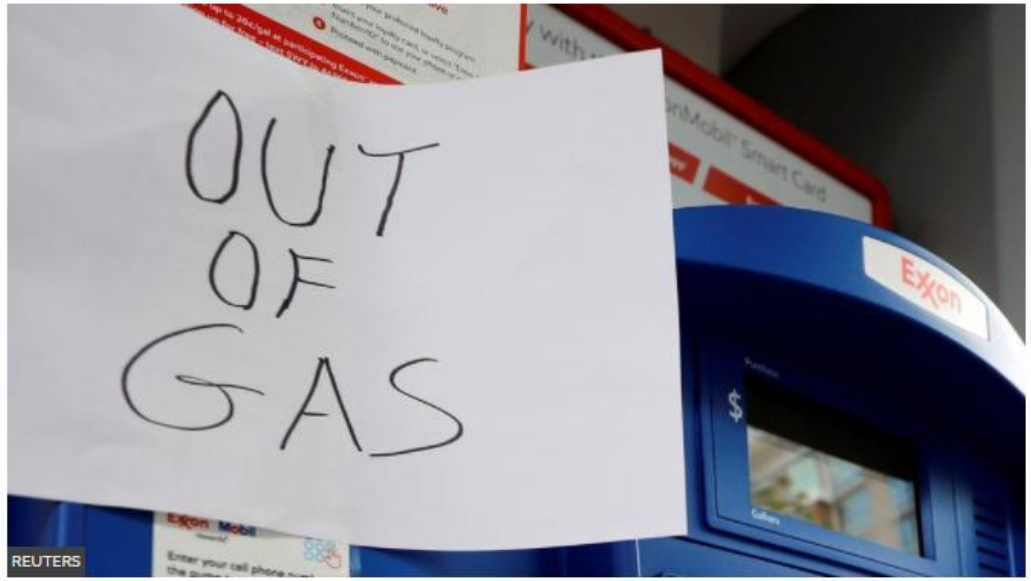
TTA

News

- IOT
- Ransomware

US offers $10m bounty for Colonial Pipeline hackers

4 days ago



The United States government has offered a bounty of up to $10million (£7.4m) for information about the hacking group known as DarkSide.

In May, a DarkSide ransomware attack shut down a vital 5,500-mile-long fuel pipeline on the east coast of the US.

The pipeline carries 45% of the fuel used on the east coast.

The bounty is offered for information which can lead to the "identification or location of any individuals" in a leadership position with DarkSide.

- What is ransomware?

A separate $5m reward has been offered for information leading to the arrest of anybody "conspiring to participate" in a DarkSide ransomware attack.

The cyber-attack caused fuel shortages after the Colonial Pipeline company shut down its operations for several days.

TTA

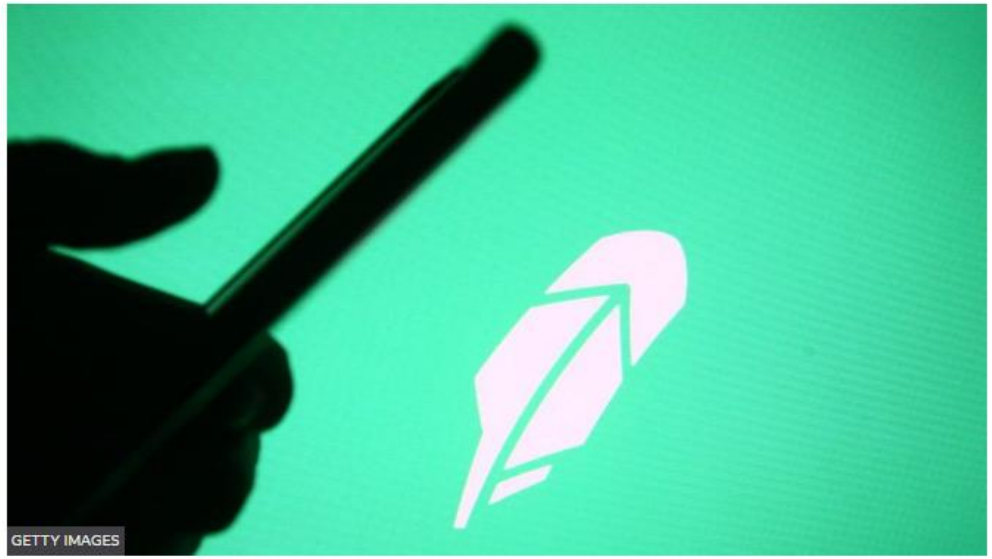
Cyber News

- Social engineering

https://www.bbc.co.uk/news/technology-59209494

Robinhood trading app hit by data breach affecting seven million

1 hour ago



US share-trading app Robinhood has been hit by a security breach that has exposed the names or email addresses of more than seven million people.

The company says the breach affected "a limited amount of personal information for a portion of our customers".

And it does not believe the most sensitive information it gathers - US social security numbers and financial information - was revealed.

Robinhood said it had rejected a demand for payment and reported the attack.

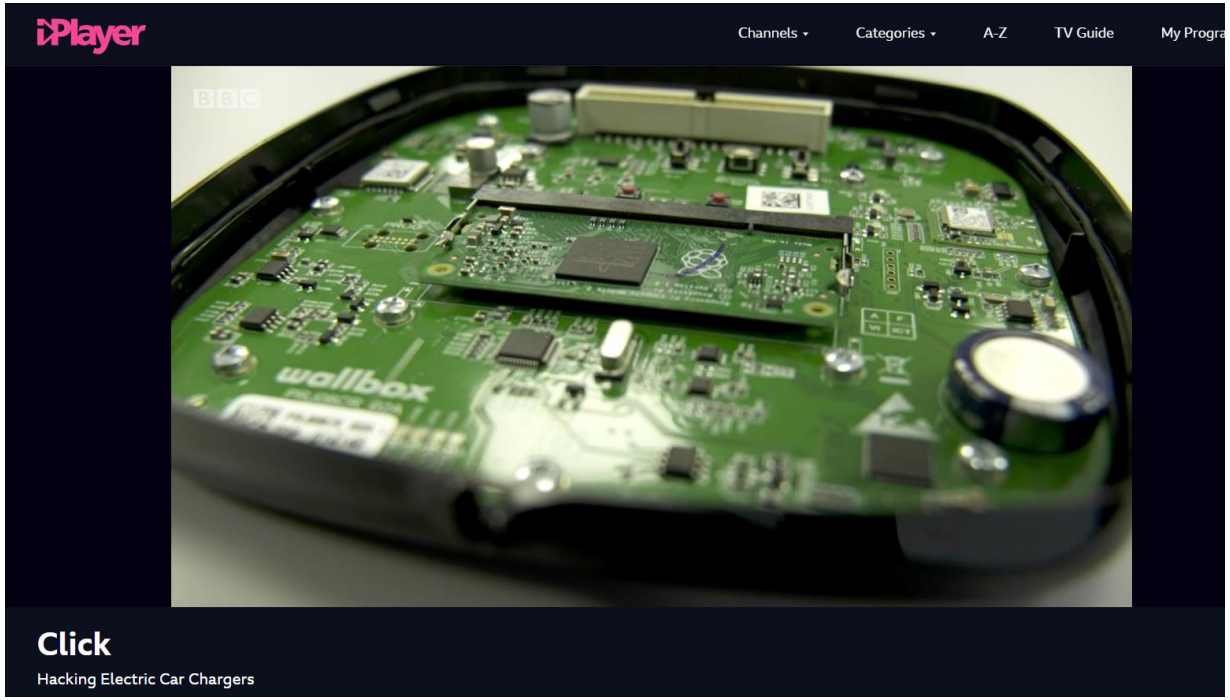
Such ransom demands are not uncommon in cyber-attacks and usually amount to a promise not to sell on the compromised data or leak it for free online. The company did not say what terms were involved in its case.

Instead of complying with what it called "extortion", Robinhood said it had notified law-enforcement authorities and hired an external cyber-security firm to help deal with the incident.

TTA

Cyber NEWS

- NEWS



BBC iPlayer - Click - Hacking Electric Car Chargers

TTA

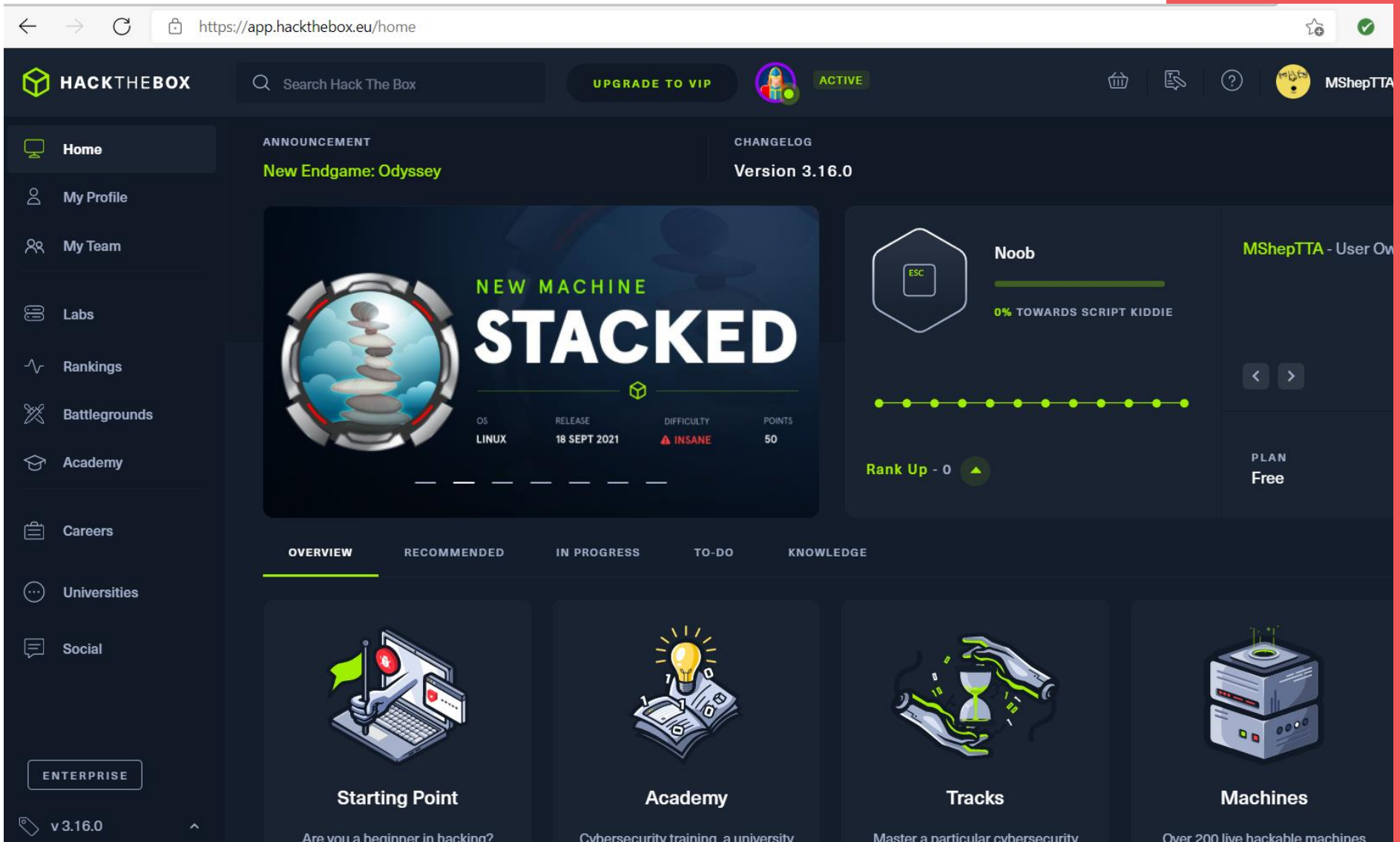
NEWS

- NEWS

TTA

Resources

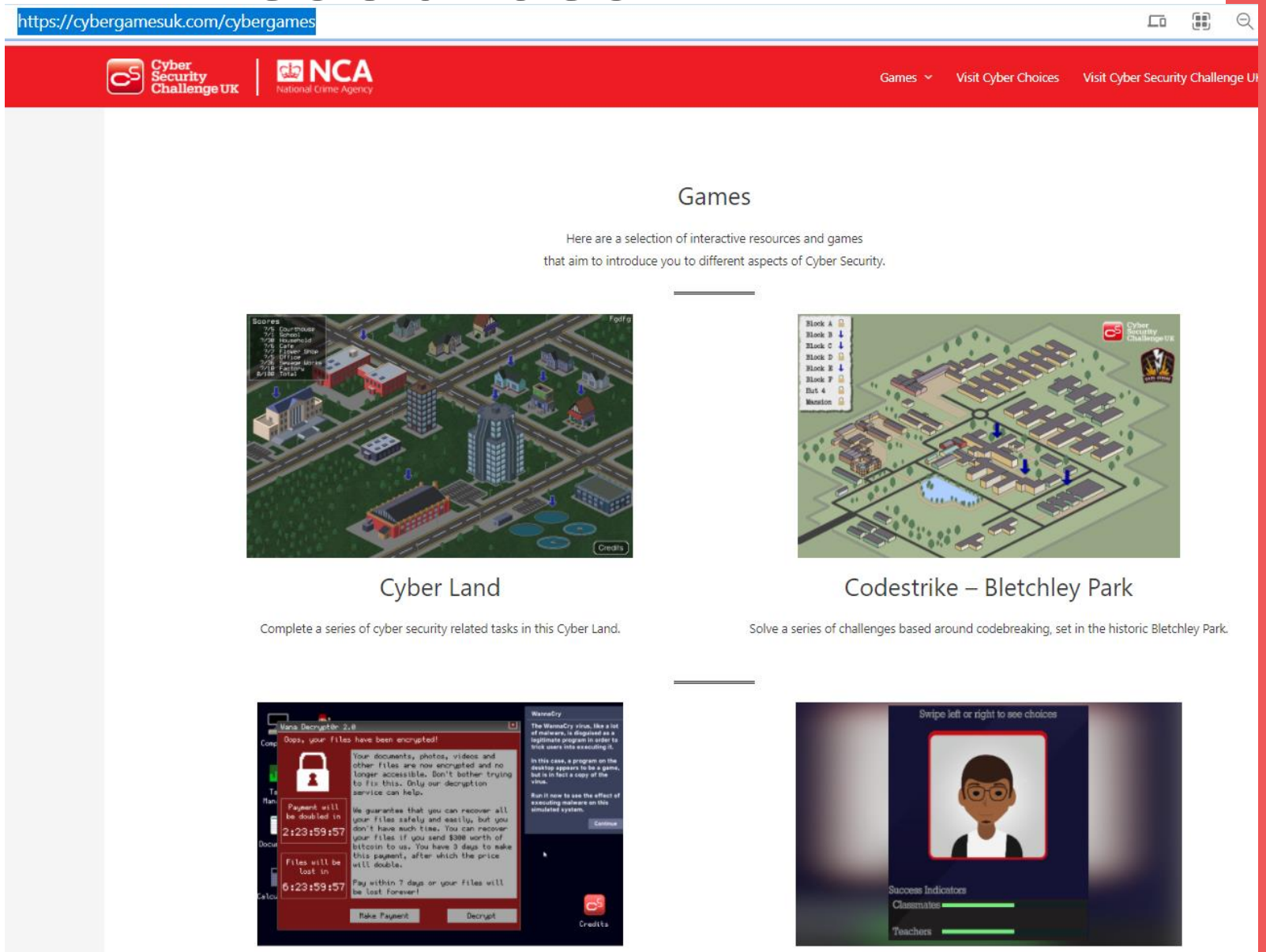
- Hack the Box



TTA

Resources

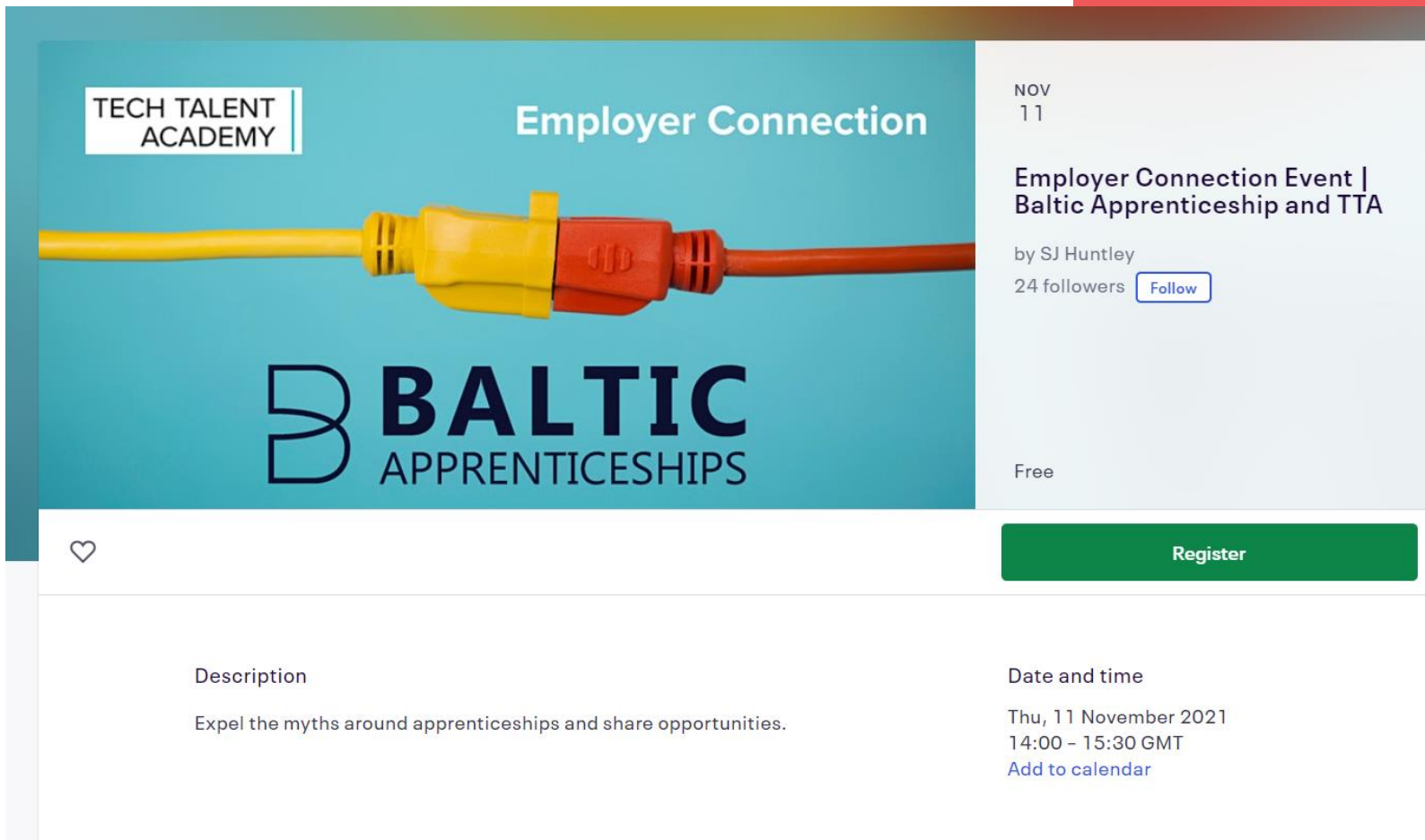
- Cyberland



TTA

Employability

- Apprenticeship
- CV workshop
- LinkedIn profile
- paul@techtalent.academy



TTA

Session overview

- Recap over last week
- Intro to Python
- Suggested solutions
- HLTs

TTA

Home Learning Task

TTA

1.

Write a program that allows user to enter their favourite starter, main course, dessert and drink.
Output a message which says – “Your favourite meal is ……with a glass of….”

2.

Write a program that

Inputs a students mark in an exam

If the mark is:

Greater than or = 90 display Grade A*

Greater than or = 80 display grade A

Greater than or = 70 display grade B

Greater than or = 60 display grade C

Otherwise display – “Have another go!”

Session overview HLT 1

food HLT.py - C:/Users/MichelleSheppard/OneDrive - TechTalent Academy/Documents - TechTalent Academy/DfE/Courses/Cyber WM Sept 14 Michelle/Sessions/Session 1 - python/food HLT.py (3.9.5)

File Edit Format Run Options Window Help

```
#Sept 2021
#Mshep TTA
#Fave food

#Write a program that allows user to enter their favourite st artermain course, dessert and drink.
#Output a message which says - "Your favourite meal is ……..with a glass of…."

print("\n\n\tWelcome to the food Guide")

starter=input("\nWhat is your favourite starter? ")
maincourse=input("What is your favourite main course? ")
dessert=input("What is your favourite dessert? ")
drink=input("What is your favourite drink? ")

print ("\n\tYour favourite meal would be: \n"+starter,"followed by "+maincourse+" with a tasty dessert of "+dessert)
print("all washed down with a large glass of "+drink)

## look at the difference when you use the seperator , or +
#use of control characters \n \t
```

TTA

HLT 2 – grade calculator

```python

#TTA MShep
#Grade calculator
#Oct 2021
print ("\n\tHello and welcome to the grade calculator\n\n")
mark = int(input("Please enter your grade:  "))
if mark >=90:
  print("\n well done you got a Grade A*")
elif  mark >=80:
  print("\n well done you got a Grade A")
elif  mark >=70:
  print("\n well done you got a Grade B")
elif  mark >=60:
  print("\nWell done you got a Grade C")
else:
  print("\nHard luck - have another go")

```

```
    Hello and welcome to the grade calculator


Please enter your grade:  34

Hard luck - have another go
>
```

TTA

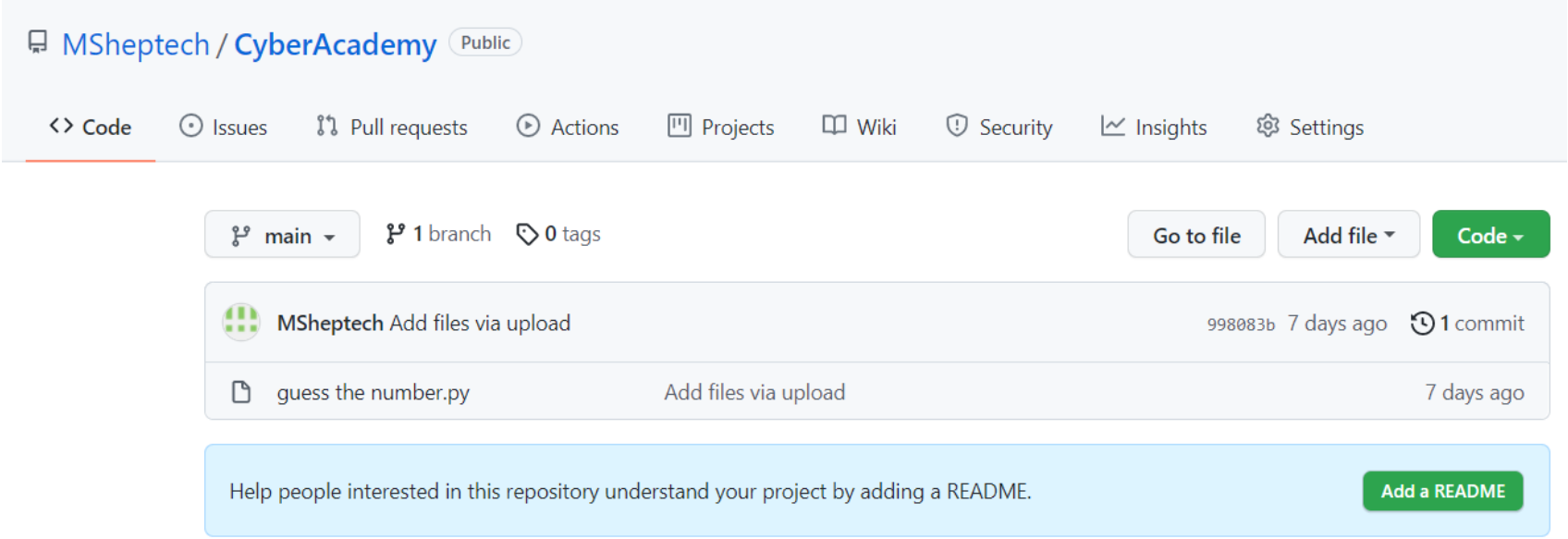
HLTs

- Submission
  - Link to repl.it
  - Email solution
  - Link to github

TTA

Session overview

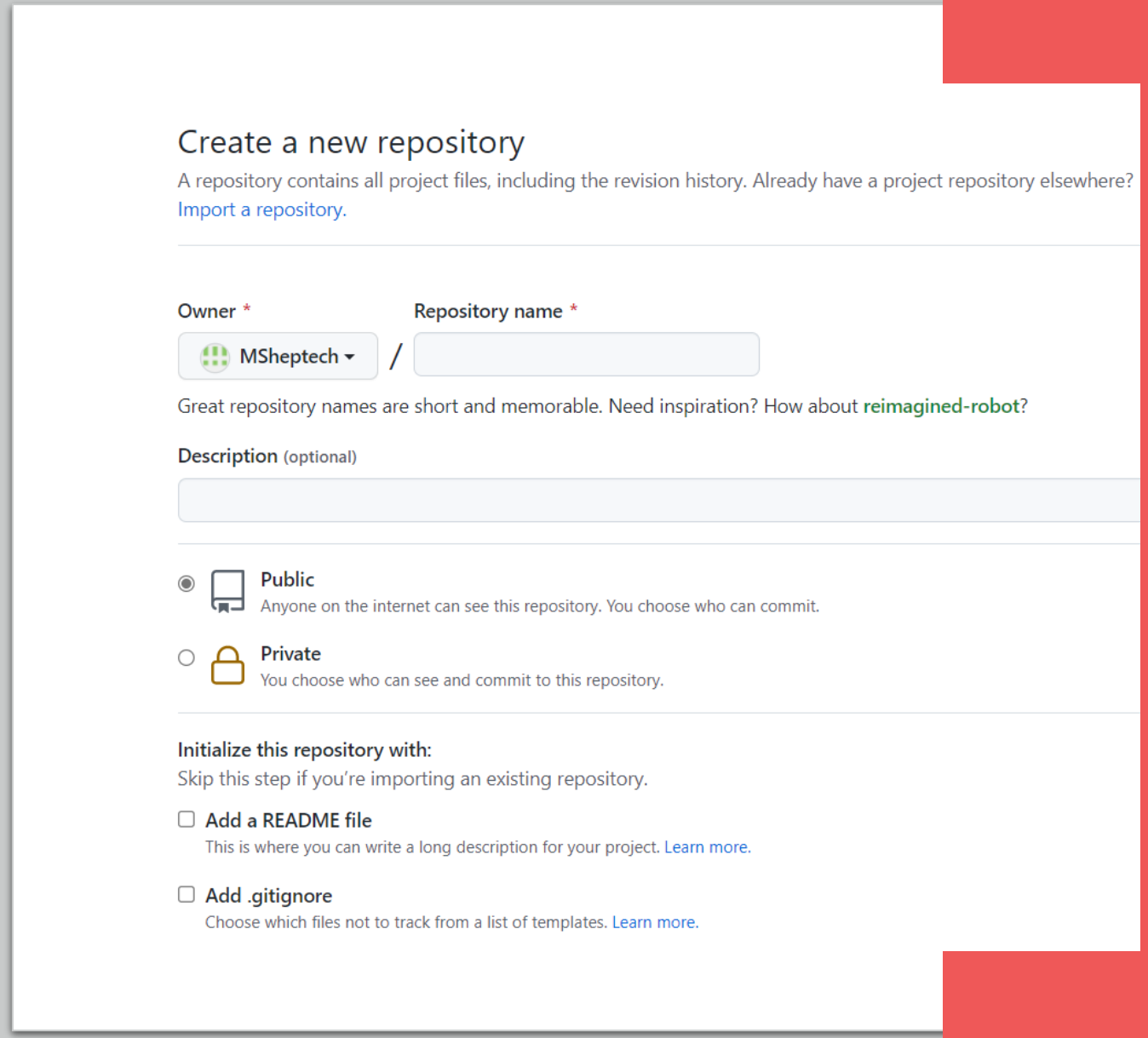
- Github



TTA

Session overview

- Github – create a new repository



TTA

Task - guess the number game

- Import random
- If statements

TTA

Task - guess the number game

Guess the number

- Create a program which the computer thinks of a number between 1 and 10 (random)
- Ask the user to input what they think the number is
- IF correct then congratulate them
- Else tell them they were wrong

How:

- Need to import a library – random
- Use the == or !=
- Print out the result

TTA

Like so …random

- If you had a go at the ‘guess the number game’
- You would have used a library called random
- import random
- This will give us access to other inbuilt commands
- Using repl.it or python
- Import random
- Num=random.randint(1,20) #integer#
- Print (Num)

```
>>> import random
>>> num=random.randint(1,30)
>>> print(num)
9
>>> |
```

TTA

Practical

```
import random
```

We re-using an existing piece of code to give us a random number

```
myName input("Hello! What is your name?")
number = random.randint(1, 10)
```

Generates a random number between 1 and 10 and stores it in the variable number

```
print("Well, “ +  myName +   " I am thinking of a number between 1 and 10.")
guess = int(input("Take a guess."))
```

Converts the input from text to an integer

```
if guess == number:
    print("Good job, " + myName + "! You guessed my number")
```

The == is *equal to*

Must be indented

TTA

Practical

```
#MShep TTA
#Sept 2021
#random guess
import random # imports the random library

print ("\n\tHello and welcome to the Guess your number game \n\n")

myName = input("Hello! What is your name? ")
number = random.randint(1,10) # random generation between 1 and 10

print("Well, " + myName +  " I am thinking of a number between 1 and 10.")

guess = int(input("Take a guess:") )

if guess == number:
    print("Good job, " + myName + " You guessed my number")
else:
   print("Wrong, better luck next time")
```

TTA

How can we improve this?
how about using iteration ie a loop
how about any validation - is it a number is it 10 or less

Like so …random

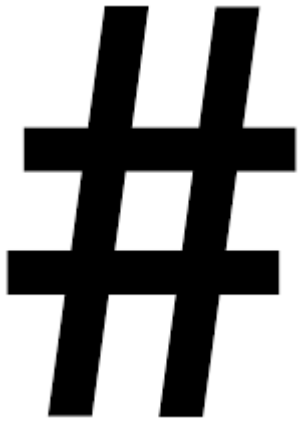
Spend 15 minutes on this.

Create this code
Use repl.it or python
You can copy the code or link to it on github
Remember if copying there may be problems with the spaces and “

```
#MShep TTA
#Sept 2021
#random guess
import random # imports the random library

print ("\n\tHello and welcome to the Guess your number game \n\n")

myName = input("Hello! What is your name? ")
number = random.randint(1,10) # random generation between 1 and 10

print("Well, " + myName +  " I am thinking of a number between 1 and 10.")

guess = int(input("Take a guess:") )

if guess == number:
    print("Good job, " + myName + " You guessed my number")
else:
   print("Wrong, better luck next time")


#How can we improve this?
#think about using iteration ie a loop
#how about any validation - is it a number is it 10 or less
```

TTA

Comments

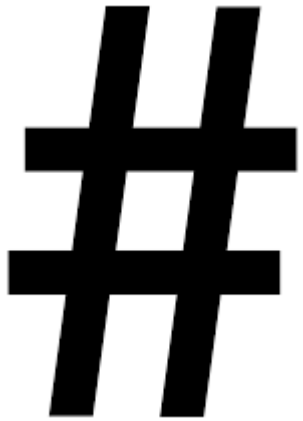


Commenting is important in coding.

It describes what is going on

It acts as a reminder, when you go back and look at your code.

The idea is for someone who has never seen the code to be able to read it, understand it and change it



TTA

Comments - Standard

File Edit Format Run Options Windows Help

```
def user_name ():
    # Function that asks the user to enter a name
    #            and then displays a greeting
    # User name is stored in variable user

    user = input ("Please enter your name ")
    print ("Hello " + user + " a pleasure to meet you")

    # end of function
```

TTA

Session Content

WHAT IS A FUNCTION?

WHAT IS THE USE OF A FUNCTION?

USING FUNCTIONS

USING COMMENTS

TTA

Calling a Function

Functions allow us to split our coding problem into smaller chunks – which make it easier to understand

They can be reused – the print function for example

Allows many people to work on the same project at once

Allows a project to be tested as each part is completed.

TTA

Functions

```
def NAME( LIST OF PARAMETERS ):
    STATEMENTS
```

In the context of programming, a **function** is a named sequence of statements that performs a desired operation.

You have already used several functions

input *print* *int* *randint*

TTA

Function Example

```
def main ():
    print ("Hello")
```

The syntax of a function. Main is the name of the function that you call in your program.

```
main()
```

In your program – this is how you call the function called main.

TTA

Practical

Write a Function that will ask your name and print it out.

Expand your ‘guess the number’ game

Add a main() procedure that holds all the game code

Can you ask the user if they would like another go? If “yes” then run the game again

TTA

Answer to Task 1 (Blue)

You must call the procedure name – ie username()

*username.py - C:/Users/MichelleSheppard/OneDrive - TechTalent Academy/Documents - TechTalent Academy/DfE/..

File Edit Format Run Options Window Help

```
#Sept 2021
#Mshep TTA
#Function

def username():
    name=input("Hello what is your name? : ")
    print("Hello",name)

username()
```

TTA

Practical Numbers Function

Write a program that will ask the user for four integer numbers and then add these numbers together before displaying the answer

```
def numbers():

    #Function to ask a user name
    #        then ask for 4 seperate numbers in 4 variables
    #It should then display the total of the 4 numbers

    #Ask the user for their name
    user = input("What is your name?")


    #Ask for the 4 varibles and numbers as whole numbers
    num_1 = int(input("Enter your first number: "))
```

TTA

Answer

```
def user_numbers ():
    # Function that asks the user to enter a name,
    #   and 4 separte numbers and displays their sum

    #   It stores the 4 numbers in variables num_1 ... num_4

    # Get the users name
    user = input ("Please enter your name ")

    # Ask the user for their numbers and convert to integer
    num_1 = int(input ("Enter your first number "))
    num_2 = int(input ("Enter your first number "))
    num_3 = int(input ("Enter your first number "))
    num_4 = int(input ("Enter your first number "))

    # add them up
    sum_num = num_1 + num_2 + num_3 + num_4

    # Display the sum
    print ("the sum of these number is " + str(sum_num))

    # End of function

user_numbers()
```

```
>>>
Please enter your name Shaun
Enter your first number 2
Enter your first number 32323
Enter your first number 35343535
Enter your first number 435
the sum of these number is 35376295
>>>
```

TTA

Parameter/argument Passing

Sometimes we need to pass values to the function. These are known as parameters or arguments.

These need to be declared within the brackets () when creating the function

Important!

- The order in which the parameters are declared is important for the order the values are passed to them.
- You can then use these arguments within the function.
- Allows for code reusability
- DRY (Don’t Repeat Yourself)

TTA

Parameter/argument Passing

Sometimes we need to pass values to the function. These are known as parameters or arguments.

These need to be declared within the brackets () when creating the function

```
def my_function(fname):
    print(fname +" Welcome")

my_function("Bob")
my_function("Harry")
my_function("Naomi")
```

TTA

Parameter/argument Passing

```
#This is a procedure called output
#It will print a given number and its square

def output(number):
    print(number,"squared =",number*number)


output(4)
output(45)
```

TTA

Multiple Parameters

A parameter is the variable (or variables) that get passed to the procedure. You can require more than one parameter.
You can pass as many parameters as you like to a procedure as long as you set it up correctly.

This will pass two numbers and add them together

```
def output_m(number1,number2):
    print(number1,"+",number2,"=",number1+number2)

output_m(3,4)
```

TTA

Practice Tasks

Complete short practice tasks – 20 minutes

Practice

TTA

Practice – using functions and procedures

- **Task 1:**
- **See if you can create a procedure that passes a number and cubes it (**3)**

- **Task 2:**
- **See if you can take two numbers and give the first number to the power of the second number eg 3**2 = 9**

TTA

```
#This is a procedure called cubey
#It will print a given number and its cube

def cubey(number):
    print(number,"cubed =",number**3)
```

Suggested answers

- **Task 1:**
- **See if you can create a procedure that passes a number and cubes it**
-

- **Task 2:**
- **See if you can take two numbers and give the first number to the power of the second number eg 3**2 = 9**

```
#this is a proc called pow
#takes two numbers and give the first number to the power of the second number

def pow(num,power):
    print (num,"to the power of ",power,"= ",num**power)
```

TTA

Practice

Task 3

Write a Calculator program which will

Ask for two numbers.

Then offers a menu to the user giving them a choi
of operator

e.g. – Enter “a” if you want to add

“b” if you want to subtract

Etc.

Include +, -, /, *, **(to the power of) and

square.

HINT – you will need to use an IF statement

This can be done by using procedures for each calculation eg add() sub() etc

TTA

Example solution

```
#Sept 2021
#MShep TTA
#Calculator



def calc():
    global number1# global enables the variable to be passed between procedures
    global number2

    print ("*****Welcome to the calculator*****")
    number1=int(input("enter the first number"))
    number2 = int(input("Enter the second number"))

    calcu=input (" what would you like to do\n\
a. Addition\n\
b.Subtraction\n")

    if calcu=="a":
        add()
    else:
        sub()

def add():
    print(number1,"+",number2,"=",number1+number2)
    calc()
def sub():
    print(number1,"-",number2,"=",number1-number2)
    calc()


calc()
```

Task 3

Write a Calculator program which will

Ask for two numbers.

Then offers a menu to the user giving them a choice of either subtraction or addition

e.g. – Enter “a” if you want to add

“b” if you want to subtract

Etc.

.

HINT – you will need to use an IF statement

This can be done by using procedures for each calculation eg add() sub() etc

Session Content

IF STATEMENT

IF ELSE STATEMENT

ELIF

TTA

Recap on the *if* Statement

We looked at basic if statement earlier

```
if <expression>:
    # write statement here
```

if statements are one of the building blocks of programming.

TTA

if Statements

File Edit Format Run Options Windows Help

```
def our_if ():
    user_input = input ("Enter One ")

    if user_input == "One":
        print ("Well done - you can read")


our_if()
```

TTA

if…else … statements

File Edit Format Run Options Windows Help

```
def our_if ():
    user_input = input ("Enter One ")

    if user_input == "One":
        print ("Well done - you can read")

    else :
        print ("Are you sure you should be in here")


our_if()
```

TTA

Indenting

File Edit Format Run Options Windows Help

```
def our_if ():
    user_input = input ("Enter One ")

    if user_input == "One":
        print ("Well done - you can read")

    else :
        print ("Are you sure you should be in here")


our_if()
```

TTA

Python Multi Selection

```
if test_1:
    statement 1
    statement 2
elif test_2:
    statement 3
    statement 4
elif test_3:
    statement 5
    statement 6
else :
    statement 7
    statement 8
```

If statement as you already know

Elif is new. It is executed if test_1 is false

Else statement as you already know

TTA

Practice

Add These functions to the program you have just created

- Use the calculator program
- Include other calculations
- Include /, *, **(to the power of) and square
- Give them the option to Quit the calculator

TTA

Logical Operators

Operator	Description
and	Returns true if both conditions are met
or	Returns true if either or both conditions are met
not	A true expression becomes false and vice versa

TTA

How to use logic operators - and

```
age= int(input("What is your age?"))

if (age > 17 and age < 60):
    print("You can learn to drive")
else:
    print("You cannot learn to drive")
```

TTA

How to use logic operators - or

```
number=int(input("Enter a number smaller than
10"))

if (number==1 or number==3 or number==5):
    print("You entered an ODD number")
else:
    print("You entered an even number")
```

TTA

Boolean variables and NOT operator

```
if (shoePrice < 9.99):
    cheapShoe= True
else:
    cheapShoe= False

if (not(cheapShoe)):
    print("The shoe is expensive")
else:
    print("This shoe is cheap")
```

This is how we assign a Boolean variable

TTA

Practice – using functions and procedures and the IF statement

- **Task 1:**
- **See if you can create a procedure that checks if someone is able to drive a car in the UK**
- **Ie is the user >=17 and they have a provisional driving licence**
- **Task 2:**
- **Create a procedure that checks if someone is old enough to have an alcoholic drink**
- **Ie is the person >=18 or is the person eating food**

TTA

licence.py - C:/Users/MichelleSheppard/OneDrive - TechTalent Academy/Documents - TechTalent Academy/DfE/Cour...

File Edit Format Run Options Window Help

```
#MShep TTA
#SEpt 2021
#procedure that checks if someone is able to drive a car in the UK

def drive():
    age=int(input("How old are you? "))
    licence=input("Do you hold a driving licence")
    if age >=17 and licence=="Y":
        print("You are able to learn to drive in the UK")
    else:
        print(" you are either too young or you need to apply for a licence")
drive()
```

Suggested answers

```
def drink():
    age=int(input("How old are you? "))
    food=input("Are you eating food in the pub?")
    if age >=18 and food=="N" or food=="Y":
        print("You are able to legally buy an alcoholic drink in the UK")
    elif age >=16 and food =="Y":
        print(" you can drink but someone over 18 must buy it for you")
    else:
        print("You are too young to drink in a pub")
drink()
```

- **Task 1:**
- **See if you can create a procedure that checks if someone is able to drive a car in the UK**
- **Ie is the user >=17 and they have a provisional driving licence**
- **Task 2:**
- **Create a procedure that checks if someone is old enough to have an alcoholic drink**

Ie is the person >=18 or is the person eating food

Session Content

LOOPS

LOOPY PROGRAM

FLOWCHART/PSEUDO CODE

CREATING PROGRAM USING THE LOOP

TTA

Loops – WHILE and FOR

- A WHILE loop is used in programming to repeatedly execute parts of a program if a condition is true.
- A FOR loop will continuously run until the variable reaches its value. For example, we our range was 1 to 6 so it repeated for 5 times!

```
i = 1
while i < 6:
  print(i)
  i = i + 1
```

```
sales = ["May", "June",
"July"]
for x in sales:
    print(x)
```

Iteration is another way of saying “loop” (iteration literally means to do something again). Loops are absolutely vital in programming in order to avoid having to write sequences of code out again and again.

Often known as iterative because it repeats.

TTA

Loops

What do you think this code will display?

```
#Sept 2021
#MShep TTA
#while loop


number = 1
while number < 10:
    print("This is turn",number)
    number = number + 1
print("The loop is now finished")
```

TTA

Loops

loopy while.py - C:/Users/MichelleSheppard/OneDrive - TechTa

File Edit Format Run Options Window Help

```python
#Sept 2021
#MShep TTA
#while loop


number = 1
while number < 10:
    print("This is turn",number)
    number = number + 1
print("The loop is now finished")
```

IDLE Shell 3.9.5

File Edit Shell Debug Options Window Help

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May
D64)] on win32
Type "help", "copyright", "credits" or "
>>>
= RESTART: C:/Users/MichelleSheppard/One
echTalent Academy/DfE/Courses/Cyber WM S
hon/loopy while.py
This is turn 1
This is turn 2
This is turn 3
This is turn 4
This is turn 5
This is turn 6
This is turn 7
This is turn 8
This is turn 9
The loop is now finished
>>>
```

TTA

While Statement

A WHILE loop is ideal if you don't know exactly how many times you will need to repeat the code. This example will keep asking for a number until it gets the right one:

Notice the indentation – this defines the statements that are in the while statement

```
while expression :
    statement 1
    statement 2
    statement 3
```

Don't forget the full colon

Statements are executed while the expression is true

TTA

While Statement *Break*

```
while expression :
    statement 1
    statement 2

    break
```

Statements are executed while the expression is true

Notice the indentation – this defines which statements are in the while statement

This command quits from the while statement – even if the expression is still true.

TTA

While Statement *Break*

```
#Sept 2021
#MShep TTA
#while loop


number=1
while number <6:
    print(number)
    if number == 3:
        break
    number = number + 1
```

What do we expect to happen when we run this code?

```
1
2
3
>>>
```

TTA

For LOOP

START END

```
for number in range (1,6):
    print(number)
```

- A FOR loop is ideal if we now how many times we want to repeat.
- How many ITERATIONS will this LOOP make?

TTA

For LOOP

START END

```
for number in range (1,6):
    print(number)
```

- How many ITERATIONS will this LOOP make?

TTA

For LOOP increments.

START END increments

```
for number in range (2,48,2):
    print(number)
```

- What will be the output?
- Have a go and see

```
for number in range (2,48,2):
        print(number)
```

```
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
```

TTA

For loop

You can also specify how the counter will count:

```
for loopCounter in range(2,9,2):
    print(loopCounter)
print("Who do we appreciate?\n")
```

```
2
4
6
8
Who do we appreciate?
```

Or the three times table:

```
num=1
for loopCounter in range(3,37,3):

    print(num,"x 3 = ",loopCounter)
    num=num+1
```

```
1 x 3 =  3
2 x 3 =  6
3 x 3 =  9
4 x 3 =  12
5 x 3 =  15
6 x 3 =  18
7 x 3 =  21
8 x 3 =  24
9 x 3 =  27
10 x 3 =  30
11 x 3 =  33
12 x 3 =  36
>>>
```

TTA

Guess the number – number of guesses

Look back at the guess the number game we programmed
We are now going to limit the user to six guesses
Do this by using a WHILE loop

```
#MShep TTA
#Sept 2021
#random guess
import random # imports the random library

print ("\n\tHello and welcome to the Guess your number game \n\n")

myName = input("Hello! What is your name? ")
number = random.randint(1,10) # random generation between 1 and 10

print("Well, " + myName +  " I am thinking of a number between 1 and 10.")

guess = int(input("Take a guess:") )

if guess == number:
    print("Good job, " + myName + " You guessed my number")
else:
   print("Wrong, better luck next time")


#How can we improve this?
#think about using iteration ie a loop
#how about any validation - is it a number is it 10 or less
```

TTA

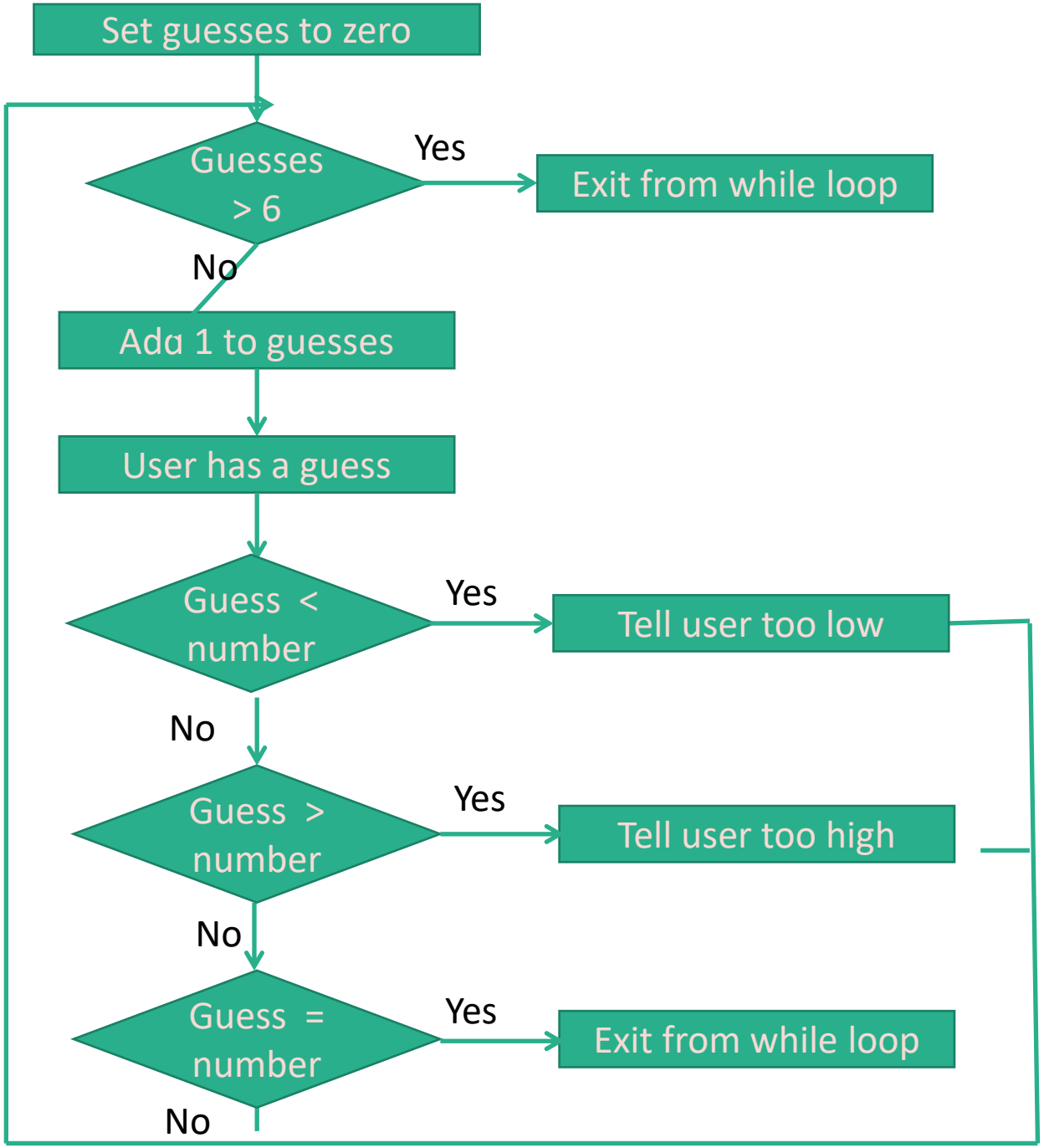
What do we need to Change?

```
import random
print("Hello! What is your name?")
myName = input()
number = random.randint(1, 10)
print("Well, " + myName + ", I am thinking of a number between 1 and 10.")
print("Take a guess.")
guess = int(input())

if guess == number:
    print("Good job, " + myName + "! You guessed my number")
```

Make sure you have 4 spaces here

TTA

```
print("Take a guess.")
guess = int(input())
```



TTA

Practice

```
#random guess
import random # imports the random library

print ("\n\tHello and welcome to the Guess your number game \n\n")

myName = input("Hello! What is your name? ")
number = random.randint(1,10) # random generation between 1 and 10

def main():
    guesses=1
    print(number)
    print("Well, " + myName +  " I am thinking of a number between 1 and 10.")

    while guesses<7:
        guess = int(input("Take a guess:") )

        if guess == number:
            print("Good job, " + myName + " You guessed my number")
            print("Thank you for playing")
            break
        else:
            print("Wrong, have another guess")
            guesses = guesses+1

main()
```

TTA

Practice – extension – tell them if they are too high or too low

```
# set the guess counter to zero
guesses = 0

# while the player still has guesses left
while guesses < 6:
    # add one to the the number of guesses taken
    guesses = guesses + 1

    # ask the user for their guess and convert it to an integer
    print('Take a guess.')
    guess = int( input() )

    # if the player is too low - tell them
    if guess < number:
        print('Your guess is too low.')

    # if the player is too high - tell them
    if guess > number:
        print('Your guess is too high.')

    # is the player guessed correctly get out of the while loop
    if guess == number:
        break
```

TTA

Further Reading

- **Functions**
https://www.w3schools.com/python/python_functions.asp

- **While Loop**
https://www.w3schools.com/python/python_while_loops.asp

- **For Loop**
https://www.w3schools.com/python/python_for_loops.asp

- **Arrays**
https://www.w3schools.com/python/python_arrays.asp

TTA

Session Content

STRING MANIPULATION

DATA STRUCTURES

CYBER RELATED PACKAGES

TTA

String manipulation

- In Python we have a variety of functions that can be applied on a string in order to increase the efficiency of our program by avoiding to print the output.
- We can apply basic built-in string functions like: *upper(), lower()* and *capitalize()* for upper case, lower case and sentence case respectively.

main.py ×

```python
#MShep TTA
#Sept 2021
#string manipulation

word = "cyber Sept 2021"
print("\t Original word is : " , word,"\n")
print(word.upper()) #all upper case
print(word.lower()) # all lower case
print(word.capitalize()) #first letter of sentence upper
print(word.title()) #all first letters are caps
print(word.count("e"))# number of times a letter appears
print(word.find("S"))#returns the place in the sentence the letter appears WHY is this 6?
print(len(word)) # length of the variable (word)

# Are there any more?
#why and where would these functions be useful?

```

TTA

Practice

- Use the program you created for driving or drinking
- How can we improve this with use of the upper() or isdigit() command?

```
def drive():
    age=int(input("How old are you? "))
    licence=input("Do you hold a driving licence")
    if age >=17 and licence=="Y":
        print("You are able to learn to drive in the UK")
    else:
        print(" you are either too young or you need to apply for a licence")
drive()
```

```
def drink():
    age=int(input("How old are you? "))
    food=input("Are you eating food in the pub?")
    if age >=18 and food=="N" or food=="Y":
        print("You are able to legally buy an alcoholic drink in the UK")
    elif age >=16 and food =="Y":
        print(" you can drink but someone over 18 must buy it for you")
    else:
        print("You are too young to drink in a pub")
drink()
```

Suggested answer

- Using upper(),lower() or isdigit() command

```
def drive():
    age=input("How old are you? ")
    licence=input("Do you hold a driving licence").upper() #converts to UPPER case
    while age.isdigit()==False: #checks that the age is a digit
        print("Please enter a valid number")
        age=input("How old are you? ")

    if int(age) >=17 and licence=="Y":
        print("You are able to learn to drive in the UK")
    else:
        print(" you are either too young or you need to apply for a licence")
drive()
```

```
def drink():
    age=input("How old are you? ")
    food=input("Are you eating food in the pub?").lower() #converts to lower case
    while age.isdigit()==False:
        print("Please enter a valid number")
        age=input("How old are you? ")
    if int(age) >=18 and food=="n" or food=="y":
        print("You are able to legally buy an alcoholic drink in the UK")
    elif int(age) >=16 and food =="n":
        print(" you can drink but someone over 18 must buy it for you")
    else:
        print("You are too young to drink in a pub")
```

Practice

```
Please enter a sentence for me to check:
34567
the sentence is all numbers
>
```

```
Please enter a sentence for me to check:
hello everyone
the sentence is all in lower case
>
```

- Create a program that asks for a user input and applies these functions on the obtained string.
- If the string belongs to one of these categories, then print which category the string belongs.
- To check if the string is in some case, you can use the is<check>() methods like

isalnum()	Returns True if all characters in the string are alphanumeric
isalpha()	Returns True if all characters in the string are in the alphabet
isdecimal()	Returns True if all characters in the string are decimals
isdigit()	Returns True if all characters in the string are digits
isidentifier()	Returns True if the string is an identifier
islower()	Returns True if all characters in the string are lower case
isnumeric()	Returns True if all characters in the string are numeric
isprintable()	Returns True if all characters in the string are printable
isspace()	Returns True if all characters in the string are whitespaces
istitle()	Returns True if the string follows the rules of a title
isupper()	Returns True if all characters in the string are upper case

TTA

Practice

- Create a program that asks for a user input and applies these functions on the obtained string.
- If the string belongs to one of these categories, then print which category the string belongs.

```
Please enter a sentence for me to check:
34567
the sentence is all numbers
>
```

```
Please enter a sentence for me to check:
hello everyone
the sentence is all in lower case
>
```

```python
#MShep TTA
#Sept 2021
#Check sentence
#Create a program that asks for a user input and applies these functions on the obtained string.
#If the string belongs to one of these categories, then print which category the string belongs.

check=input("Please enter a sentence for me to check: \n")

if check.isupper()==True:
  print("the sentence is all in Upper case")

if check.islower()==True:
  print("the sentence is all in lower case")

if check.isdigit()==True:
  print("the sentence is all numbers")

```

TTA

The *in* and *not in* operators

- Python also provides membership operators that can be used with strings.
- The *in* operators returns **True** if the first operand is contained within the second and **False** otherwise.
- The *not in* operator does the opposite.

```
x = ["apple", "banana"]

print("pineapple" not in x)

# returns True because a sequence with the value "pineapple" is not
in the list
```

```
x = ["apple", "banana"]

print("banana" in x)

# returns True because a sequence with the value "banana" is in the
list
```

TTA

Built-in String Functions

- At the most basic levels, computers store all information as numbers. To represent character data, a translation scheme is used which maps each character to its representative number.
- The simplest scheme in common use is called ASCII. It covers the common Latin characters.
- *ord(c)* - returns an integer ASCII value for the given character

Function	Description
chr()	Converts an integer to a character
ord()	Converts a character to an integer
len()	Returns the length of a string
str()	Returns a string representation of an object

TTA

String indexing

- The process where individual items in an ordered set of data can be accessed directly using a numeric index or key value is called **indexing**
- In Python, Strings are ordered sequences of character data, and thus can be indexed in this way. Individual characters in a string can be accessed by specifying the string name followed by a number in square brackets [].
- String indexing in Python is zero-based: the first character has index 0, the next has index 1 and so on. The index of the last character will be the length of the string minus 1.

TTA

Practice

E	x	a	m	p	l	e	s
0	1	2	3	4	5	6	7

- text= 'example'
- print (text[2])
- print (text[2:5])
- x=len (text)
- print (x)

f	o	o	b	a	r
0	1	2	3	4	5

```
>>> s = 'foobar'

>>> s[0]
'f'
>>> s[1]
'o'
>>> s[3]
'b'
>>> len(s)
6
```

TTA

Data Structures in Python

Data Structures in any programming language are used to store and retrieve information, which also defines the relationship between the data and operations that are authorized to be performed on that data. There are two types:

1.Primitive (Integer, Boolean, Float, Strings)
2.Non-Primitive (Arrays, Tuples, Lists, Dictionaries, Sets, Files)

We have already discussed the primitive data structures above so let's move on to the non-primitive ones. Of all the aforementioned, Lists and Dictionaries are the most popular ones to be used by the developers.

TTA

Lists

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
  print("Yes, 'apple' is in the fruits list")
```

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
  print(x)
```

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

- A list holds an ordered collection of items which can be strings, integers, etc.
- The items need to be closed in “[]” square braces to indicate that the group of items needs to be treated as a list.

- append()
- count()
- reverse()
- extend()
- insert()
- pop()
- remove()
- sort()

TTA

Dictionaries

- Dictionaries are made of key-value pairs, where key is used to identify the item and the value holds the value of the item. Let us see how to declare a dictionary say “dicts” and try to iterate through each item using the items() function.

```
thisdict =  {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict["model"]
print(x)
```

TTA

Cyber packages in Python

1. Faker

- Let us start with a very simple one! As the name says, Faker is a package that helps to create fake and random data. But, how is that useful? Where testing plays an essentially major role, Faker can be useful to generate test and random data. Many companies use this package to run automation testing and stress testing their software and applications.

main.py

```python
from faker import Faker
fake=Faker()
print(fake.email())
print(fake.country())
print(fake.name())
print(fake.url())
```

Practice

•Generate a fake address with latitude and longitude using faker.

TTA

Practice

Cyber packages in Python

1. Faker

- Generate a fake address using faker in repl.it
- Can you generate any fake.text()

fake_ID

Run ▶

main.py

```python
from faker import Faker
fake=Faker()
for i in range(0,20):
  print(fake.email())
  print(fake.country())
  print(fake.name())
  print(fake.url())
  print("\n\n")
```

Console Shell

```
Leslie Lopez
http://bryant.com/



timothy26@hotmail.com
Reunion
Gina Fitzgerald
http://jones.com/



smithpaula@gmail.com
Kazakhstan
Michael Armstrong
https://dalton.com/



colton62@stephens.com
Pitcairn Islands
Morgan Johnson
https://montgomery.com/
```

TTA

Cyber packages in Python

- **2. NumPy**
- NumPy is predominantly used in analyzing numerical data and it provides a powerful N-dimensional array object that has sophisticated functions to accommodate the computational capabilities. Let us see a very simple example of using numpy to generate a random number array.

```
In [18]:
import numpy as np
e = np.random.random((2,2)) #Generate random numbers
numones = np.ones((2,2)) #Generate ones
print(e)
print(numones)
```

```
[[ 0.65751963  0.37879833]
 [ 0.40361354  0.88010457]]
[[ 1.  1.]
 [ 1.  1.]]
```

TTA

Home Learning Tasks
10 Nov

TTA

1.Look closely at the three times table example. Write a similar program to calculate the four times table.

2.Write a program that will calculate the five times table.

3. Extend your guess the number game to let the user know if they are ‘too high’ or ‘too low’
If possible, limit them to 6 guesses (while loop)

EXT. Try writing a program that will prompt for an integer (whole number) and print the correct times table (up to 12x). Test with all the tables up to 12.

OPTIONAL HOME LEARNING TASK 5:

PASSWORD CHECKER

1. Create a password validation checking program, the program will ask the user to input a password, re enter the password and the tells the user if the password is weak, medium or strong.

TECH TALENT
ACADEMY

TTA