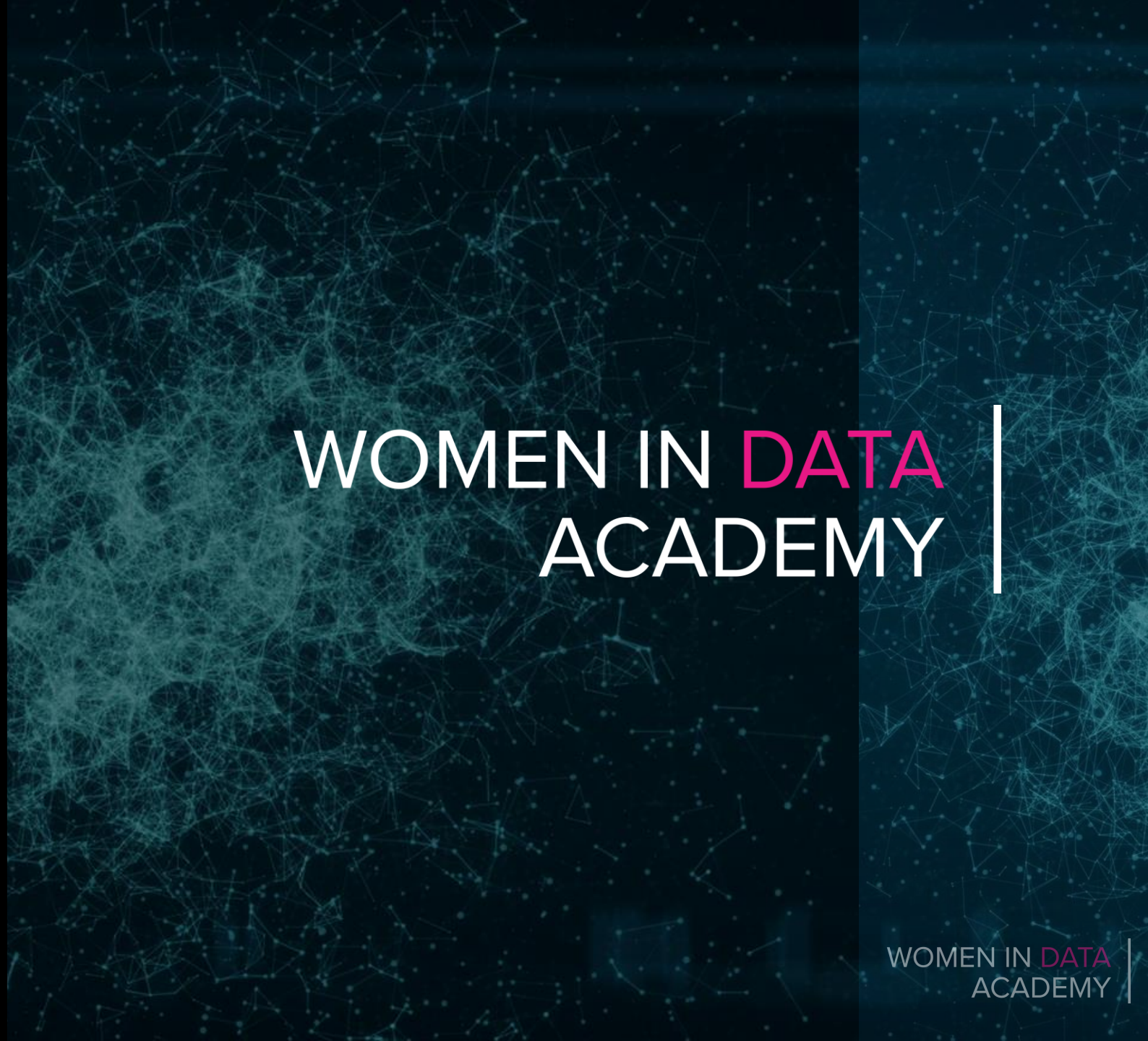
Data Science Python

Module 1 - Session 2

TECH TALENT ACADEMY



WOMEN IN DATA ACADEMY

Session Content

- If, If Else and Else Recap
- Imbedded Tasks
- Logical Operators
- Functions and Procedures

User Input

- We store data in a variable
- We use the input command to get the user to input a value.
- Don’t forget that it comes as a string
- We can cast to a different data type if required.

```
menuOption=int(input("Select an option 1,2 or 3"))
```

If/elif Statements

- We make the computer do condition checks by using if, elif (else if) or else.
- Remember the code needs to be indented to let the computer know it is part of that conditional check.

```
if (menuOption==1):
    print ("These are all the Running Trainers")
elif(menuOption==2):
    print ("These are all the Classics")
elif(menuOption==3):
    print ("This are all the Boots and Shoes")
else:
    print ("You didn't choose the correct option")
```

Imbedded tasks if/loops

- We can imbedded If statements within If Statements.
- We can also do it with loops as well
  - However, with loops be careful about the names you use!

```
number1 = 1
number2 = 2

if number1 == 1:
    if number2 == 2:
        print("Both numbers match up")
    else:
        print("First number matches but not the second")
else:
    if number2 == 2:
        print("Only the second numbers matchs up")
    else:
        print("None match")
```

```
for i in range(10):
    for j in range(10):
        print("i:", i, "j:", j)
    print("Start of next i loop")
```

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

Logical Operators - and

and: Returns true if both conditions are met

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

if age > 17 and age <60:
    print("You can learn to drive")
elif age > 50 and age <60:
    print("You can learn to drive, but better learn soon!")
else:
    print("You cannot learn to drive")
```

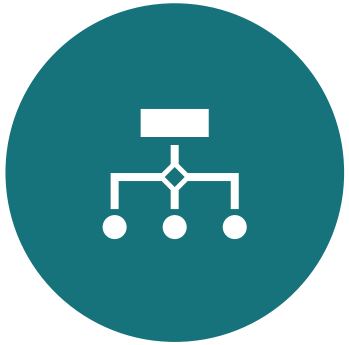
Logical Operators - or

or: Returns true if both or one condition is met

[22]:
```
number = int(input("Enter a number smaller than 10: "))

if number == 1 or number == 3 or number == 5 or number == 9:
    print("You entered ann ODD number")
else:
    print("You entered ann EVEN number")
```

Session Content



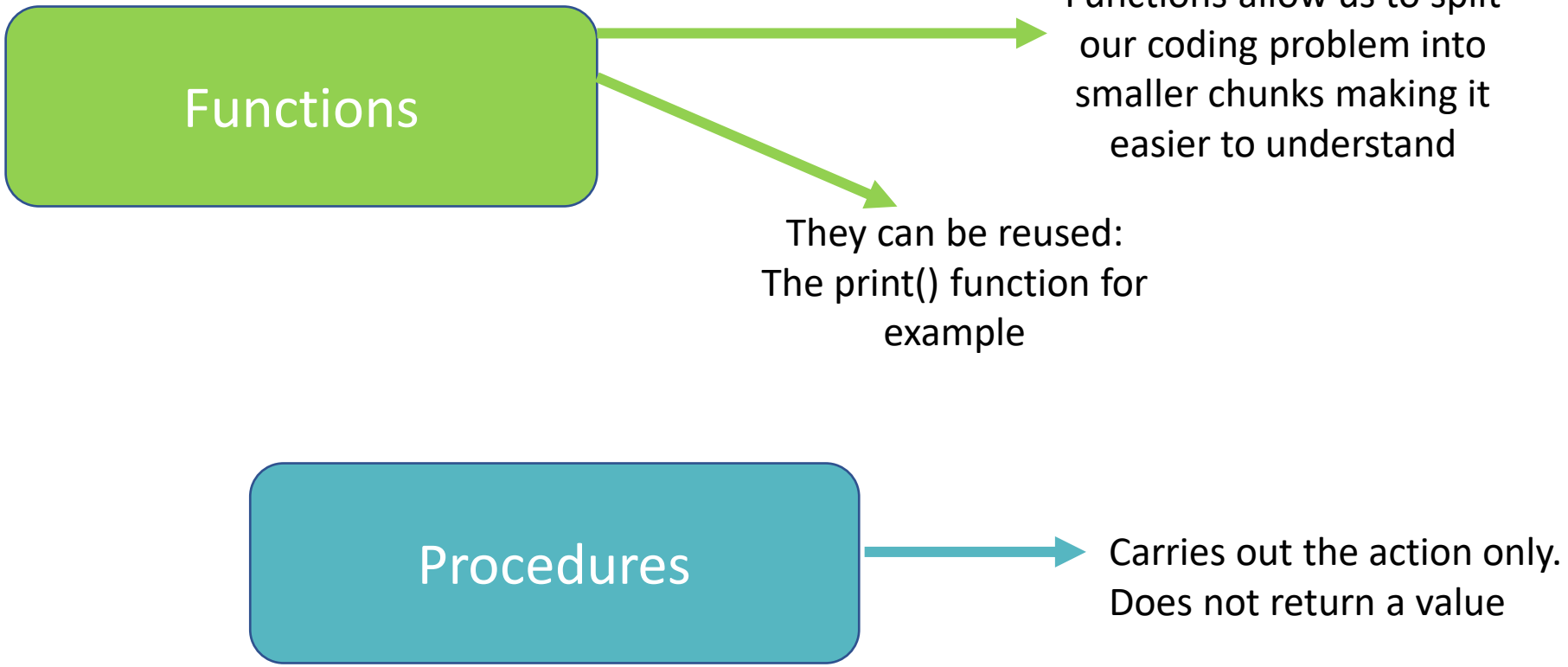
WHAT IS A FUNCTION AND PROCEDURE?

USING FUNCTIONS AND PROCEDURES

WHAT IS PSUEDOCODE?

Procedures & Functions

Both functions and procedures are created the same



Function Example

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

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.

Procedures

- We create a procedure by using the key word def followed by the name_of_the_procedure and ():
- We then write what we want this procedure to do, making sure it is indented.
- We then call the procedure within the main script by: procedure()

```
def procedure_1():
    name = input("What is your name? ")
    age  = int(input("What is your age? "))

    print("Hello", name + "! You will be ", age + 1, "For your next birthday!")
```

```
procedure_1()
```

Functions

- We create a function by using the key word def followed by the name_of_the_function and ():
- We then write what we want this procedure to do, making sure it is indented.
- **However**, we need to make sure we are returning a value using the keyword return.
- We then call the procedure within the main script by: name_of_the_function ()

```
def function_1():
    name = input("What is your name? ")
    age  = int(input("What is your age? "))

    output = "Hello " + name + "! You will be " + str(age + 1) + " For your next birthday!"

    return output
```

```
    message = function_1()
    print(message)
```

```
Hello Andy! You will be 33 For your next birthday!
```

Parameter Passing

Sometimes we need to pass values to the function/procedure.

We need to declare these within the () when creating the function/procedure

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 parameters within the function.
- Allows for code reusability
- DRY (Don't Repeat Yourself!)

Parameter Passing

```
def procedure_2(inp_name,inp_age):
    name = inp_name
    age  = inp_age

    print("Hello", name + "! You will be ", age + 1, "For your next birthday!")
```

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

procedure_2(name, age)
```

```
   procedure_2(age, name)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
 in
----> 1 procedure_2(age, name)

 in procedure_2(inp_name, inp_age)
      3     age  = inp_age
      4
----> 5     print("Hello", name + "! You will be ", age + 1, "For your next birthday!")

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Functions/Parameters can call other functions/parameters

Calling on others!

Procedures can call other procedures and functions.

```
def function_3():
    name = input("What is your name? ")
    age  = int(input("What is your age? "))

    output = "Hello " + name + "! You will be " + str(age + 1) + " For your next birthday!"

    return output

def procedure_3():

    message = function_3()

    print(message)
```

```
    procedure_3()
```

```
Hello Andy! You will be 33 For your next birthday!
```

What is Pseudocode

Pseudocode is a step by step plan of how to solve and complete a program. It can be written or drawn in a Data Flow Diagram (DFD).

Written Example:

Question: Write a program that asks a user their name and age, then outputs it to the screen.

Answer:

String Name = input from user

Integer Age = input from user

Output Name

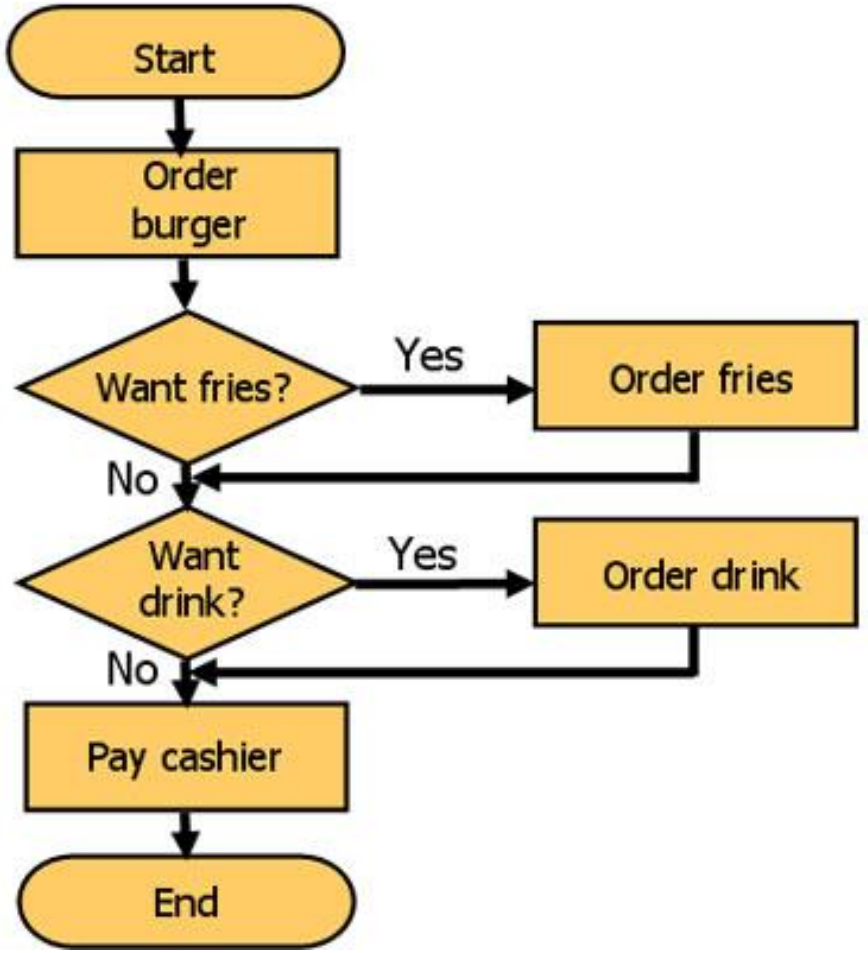
Output Age

Data Flow Diagrams

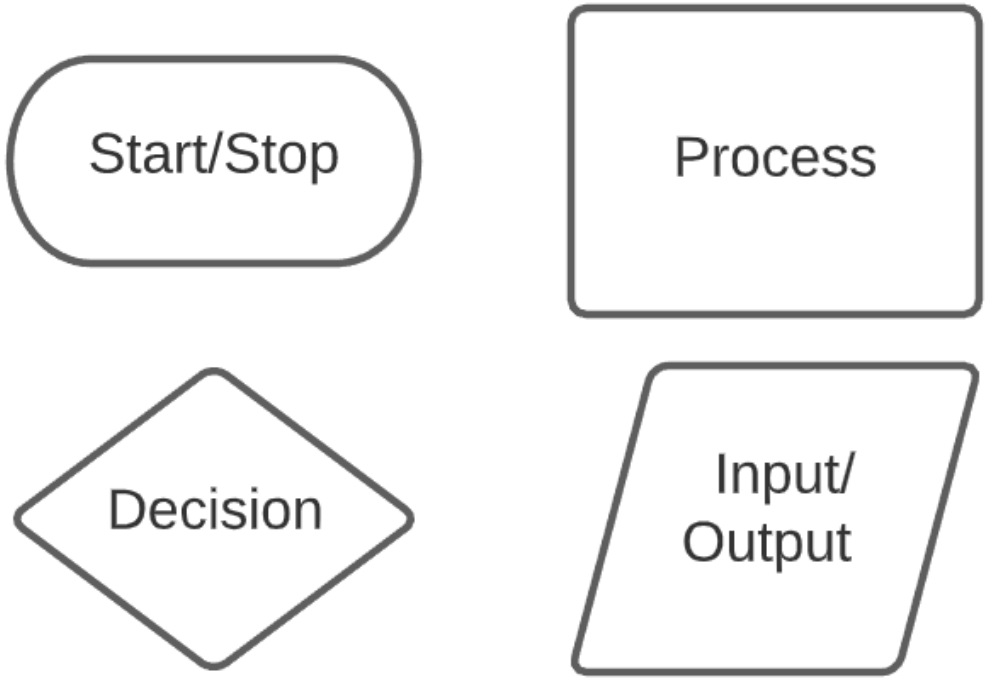
A **flowchart** can help breakdown and visualise the steps in a program including **inputs**, **outputs**, **selection** and **loops**.

Before writing code, you can use a flowchart to create a diagram of all the **steps** in your algorithm.

How can this help?



Data Flow Diagrams

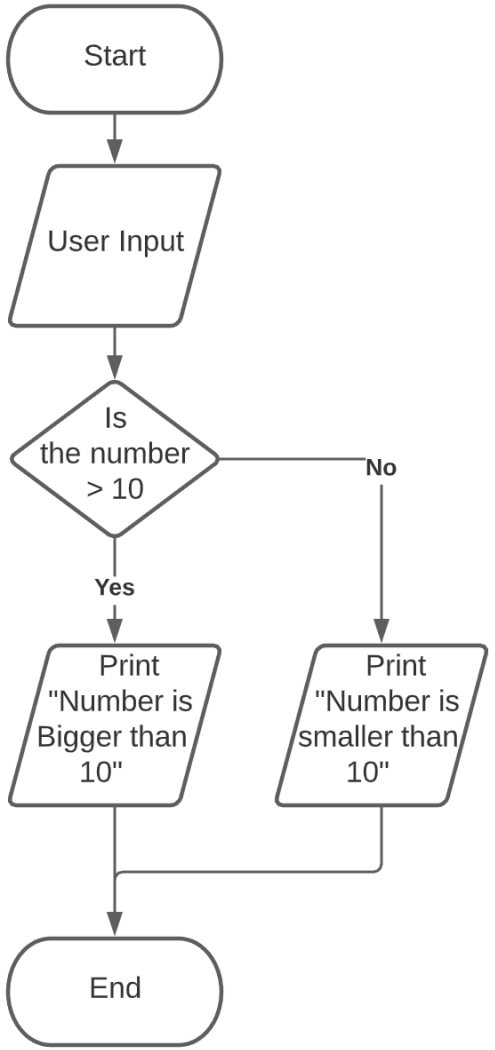


Tasks

- **Task 1:** Write a program that asks the user for an integer number and checks if it is > 10. If it is, it will print “Number is Greater than 10”, else “Number is smaller than 10”.

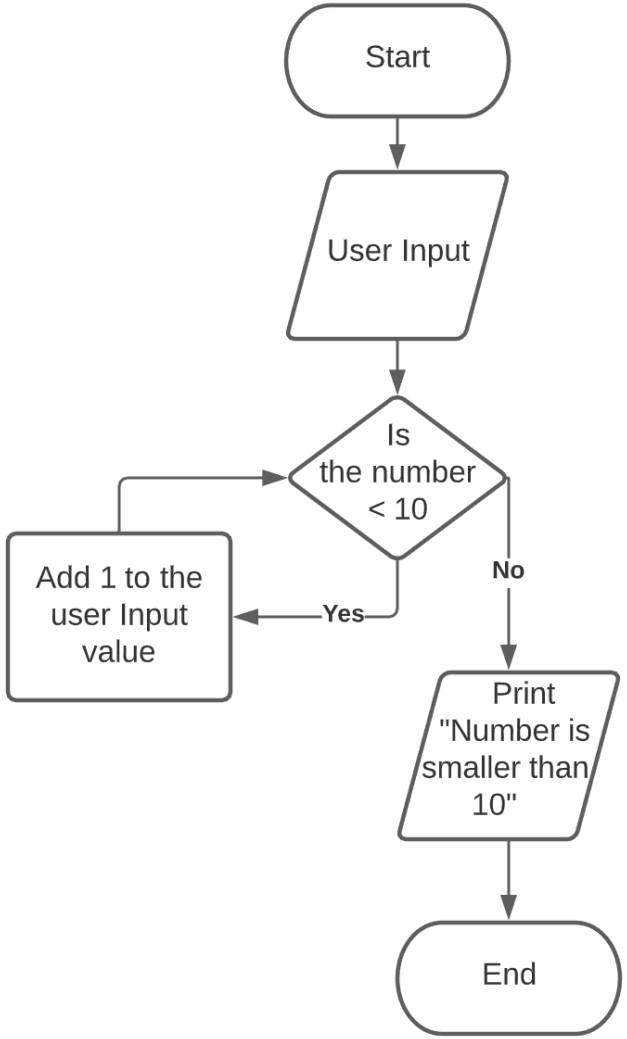
- **Task 2:** Then write a loop program that ask the user for an integer number and check if it is < 10. If it is < 10 then it keeps adding 1 to the value.

Write Conditional DFD– Pseudo Code



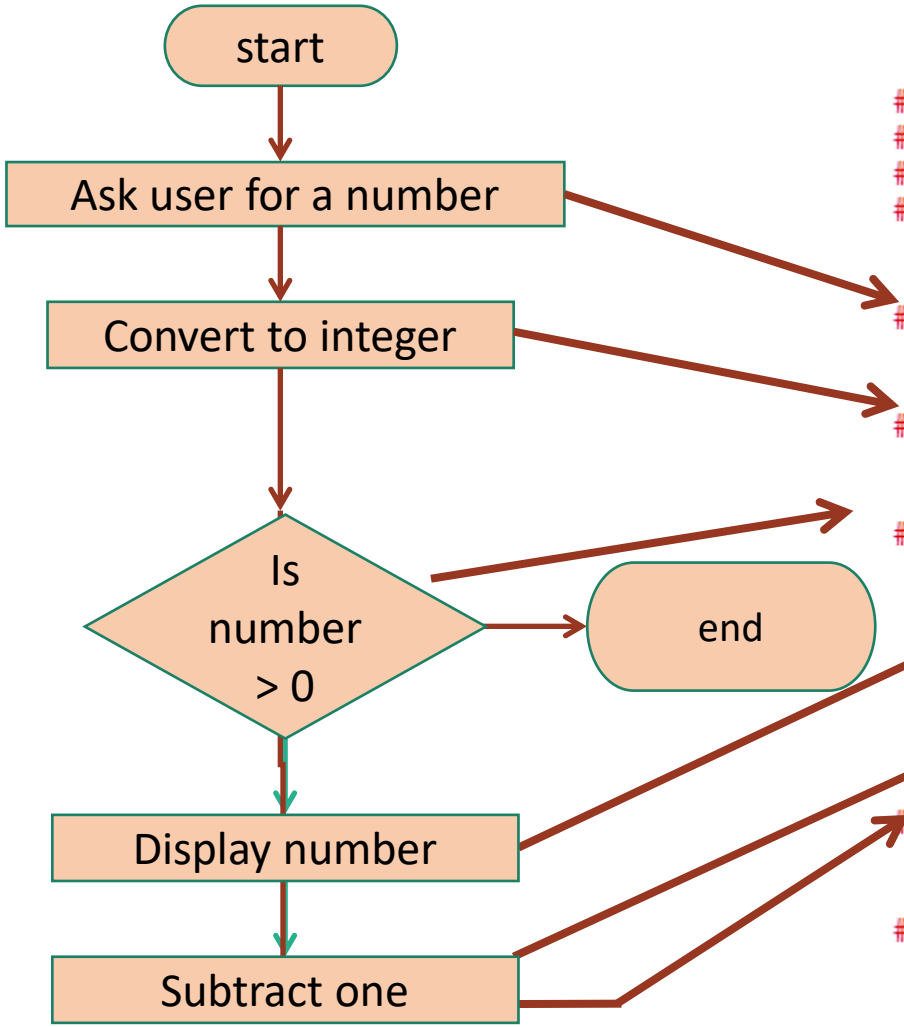
Task 1

Loopy Program – Pseudo Code



Task 2

Write a Loopy Program Example




```
# program to ask the user for a number
# and count down to 1,
# It wil do this by displaying the number
# and subtracting 1 from it

# ask the user for a number

# convert it to an integer

# while the number is greater than zero

    # display it

    # subtract 1 from it

    # end of loop

# end of program
```

Write a Loopy Program – (the code)

```
# program to ask the user for a number
# and count down to 1,
# It wil do this by displaying the number
# and subtracting 1 from it


# ask the user for a number
# convert it to an integer
# while the number is greater than zero
    # display it
    # subtract 1 from it
# end of loop

# end of program
```

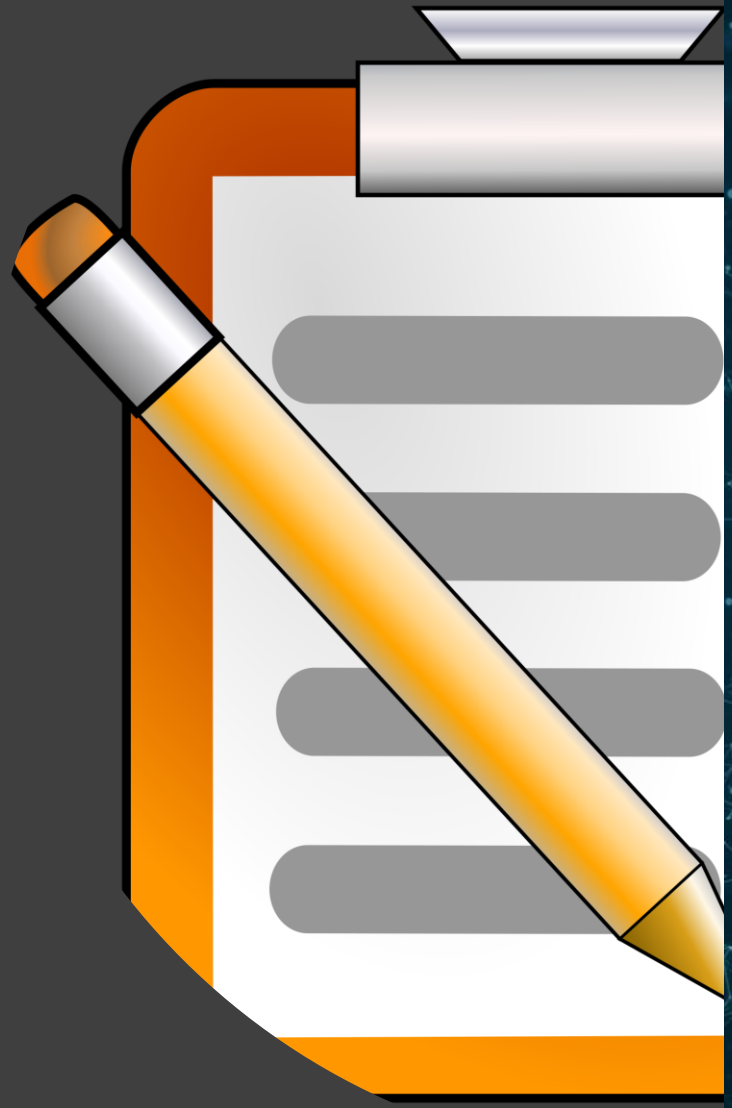
```
# program to ask the user for a number
# and count down to 1,
# It wil do this by displaying the number
# and subtracting 1 from it

# ask the user for a number
user_num = input ("Please enter a number ")
# convert it to an integer
user_num = int(user_num)
# while the number is greater than zero
while user_num > 0 :
    # display it
    print ( user_num )
    # subtract 1 from it
    user_num = user_num - 1
# end of loop

# end of program
```

TECH TALENT ACADEMY

Home Learning Tasks



1. Create your own Data Flow Diagram (DFD), a subject of your own choice, Example: Fast food order and convert it into code.
2. As an extension to the motorbike task that costs £2000 and loses 10% of its value every year. Using a loop, print the value of the bike every following year until it falls below £1000 by using a function and passing in parameters.
3. Write a program which will ask for two numbers from a user. Then offer a menu to the user giving them a choice of maths operators. Once the user has selected which operator they wish to use, perform the calculation by using a procedure and passing parameters.

WOMEN IN DATA ACADEMY

TECH TALENT
ACADEMY

WOMEN IN DATA ACADEMY

TECH TALENT
ACADEMY