Python

KS4 Programming Workbook

Tick ✓

(Once completed)

Topics covered:

- Data types and variables. □
- Advanced Strings and Input. □

□

Name ________________________

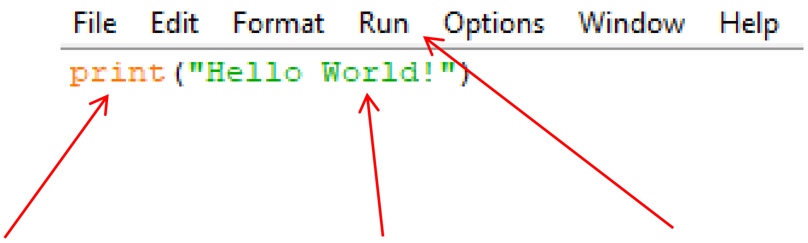
Teacher ______________________ Form

Getting Started

We will be using the Python Idle to program in the next few lessons. This is useful because:

- It distinguishes keywords using different colours.
- It runs your code and provides an output with a click of a button.

Here is an example:



```
print("Hello World!")
```

Orange – built in Python command.

Green – built in Python command.

Click 'Run' then 'Run Module' to run the program. (Or press 'F5')

Recap

Explore the Python shell and answer the following questions:

1) How would you print 'Hello' to the screen?

____

2) What is the result of 12345 + 69011?

____

3) What is the result of 344 ÷ 67?

____

____

4) What is the result of (18-30) × (20÷22)?

____

* Use python as a calculator *

5) How would you print your name in a sentence using a variable? (Write the code, it is two lines)

____

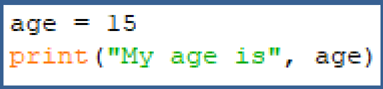
____

Data Types

Definition	Example
String is a set of characters.	Computer Science is fun
Integer is a whole number.	34
Float is a decimal number.	34.62
Char is one character.	T
Boolean has two values, either true or false.	True

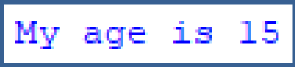
Variables

A variable allows you to store temporary values in a named location. These names should be relevant to the data being stored. To program a variable we define a name and an initial value to store.



```
age = 15
print("My age is", age)
```

»

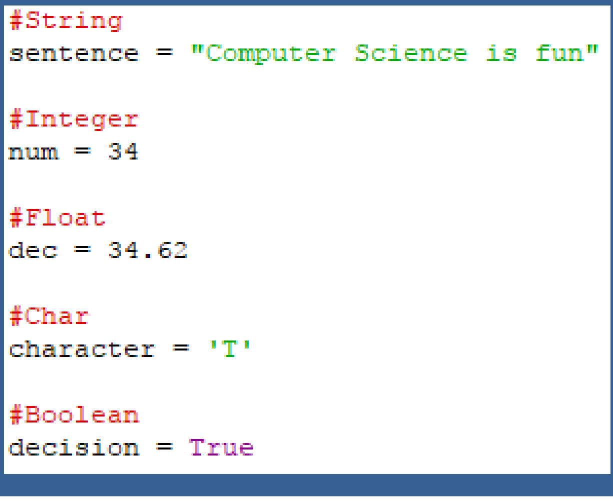


```
My age is 15
```

We use assignment symbol = to place a value into a variable.

- Left hand side – name of variable (age).
- Right hand side – initial value (15).

There are different types of variables based off the data types in Python. Python does not require programmers to enter a data type, whereas most other languages do. Below is a Python file creating variables from the examples above.



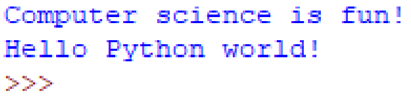
```
#String
sentence = "Computer Science is fun"

#Integer
num = 34

#Float
dec = 34.62

#Char
character = 'T'

#Boolean
decision = True
```

Exercise I

Each task shows an example of how your output of the program should look like. Save files as 'Task1.py', 'Task2.py' etc so they can be marked!

1) Store a message in a variable and print it to screen then print another message to the screen using the same variable.

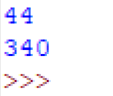
Example output =



```
Computer science is fun!
Hello Python world!
>>>
```

2) Assign a variable with the value 10 and another variable with the value 34. Print the sum and product of both variables in separate print statements.

Example output =



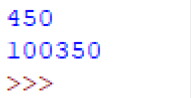
```
44
340
>>>
```

3) Assign 100 and 350 to integer variables and add them. Secondly, assign the two numbers as string variables and add them. Print both answers to the screen. Why is there a difference?

____

____

Example output =

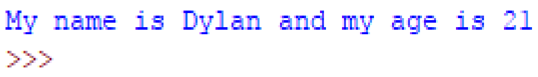


```
450
100350
>>>
```

4) Write a sentence stating your name and age using variables.

Hint: Commas "," separate strings from variables in print().

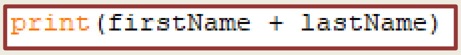
Example output =



```
My name is Dylan and my age is 21
>>>
```

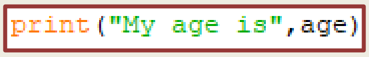
Advanced Strings and Input

Print statements allow you to add strings together using +



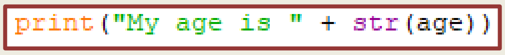
```
print(firstName + lastName)
```

If we want to use integers or another data type then we have to use commas.



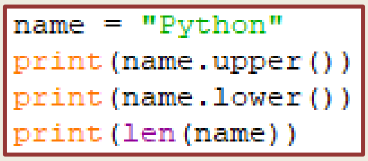
```
print("My age is",age)
```

Alternatively, we can convert data types such as making the age a string.



```
print("My age is " + str(age))
```

What will each of these operations return?



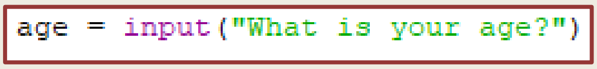
```
name = "Python"
print(name.upper())
print(name.lower())
print(len(name))
```

upper() will return ____________________

lower() will return ____________________

len() will return ____________________

We can also read strings from the Python shell which is essentially user input. We can do this by using the Python function input(). It would be useful to say something before taking an input in order to prompt the user, we can do this by entering a string inside the brackets of input().

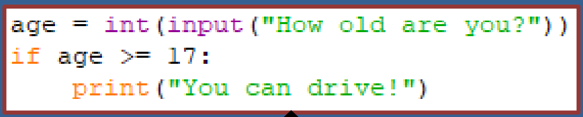


```
age = input("What is your age?")
```

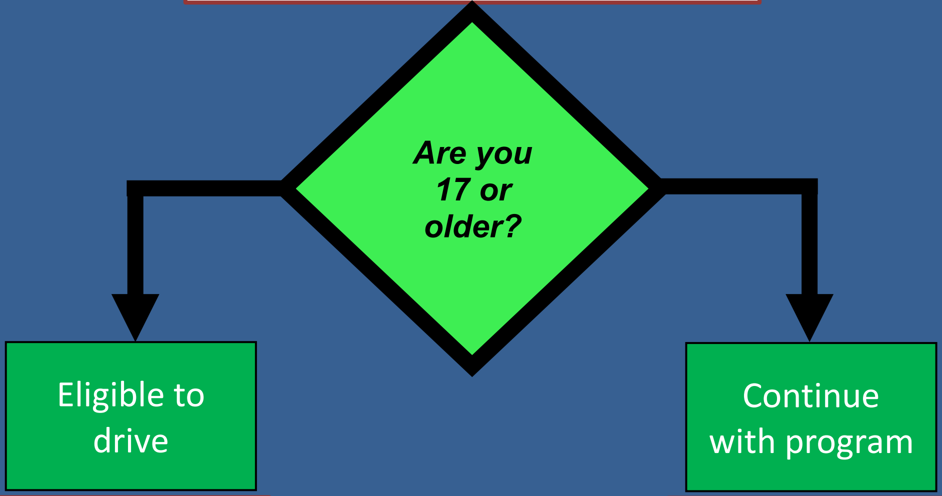
Why assign it to a variable?

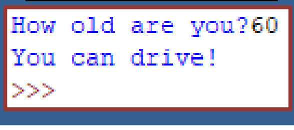
IF Statements

IF statements allow programmers to make **decisions**. We can check for a condition and then carry out an action depending whether the condition is true or false.

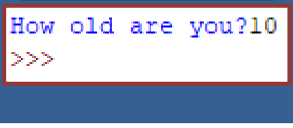


```
age = int(input("How old are you?"))
if age >= 17:
    print("You can drive!")
```





```
How old are you?60
You can drive!
>>>
```



```
How old are you?10
>>>
```

Exercise 2

1) Assign 3 variables for day, month and year as integers. Now create another variable named dob(date of birth), using the 3 variables create a dob format such as 12/08/2017. Finally, print dob to the screen.

Hint: You will need to convert variables to string.

2) Prompt the user with a maths question then print the answer in a full sentence such as "The answer is 4."

3) Write a program that takes in a number by asking "What mark did you receive in coursework?", if it is above 60 say "Well done!". What happens if you enter a number below 60 and why?

____

____

4) Write a program that asks for the name of the users friend, if that friend is "william" then say "william loves Python". If it is not William then say <friends name> "loves football!" using two IF statements. You should make the name entered by the user lowercase, why?

____

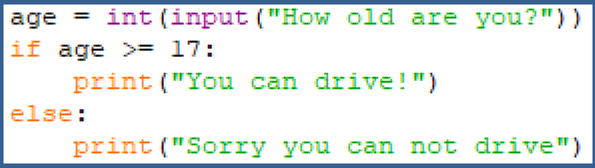
Hint: Operator for not equals comparison is !=

Challenge

Write a program that checks if a year entered by the user is a leap year. Firstly, check if the year is divisible by 4. If the condition is true, check if the year is divisible by 400 and check if the year is not divisible by 100. Only one of the conditions in the second IF statement needs to be true for there to be a leap year. Good luck!

Else IF Statements

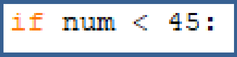
Else IF statements allow you carry out an extra action. You do not specify a condition for the else branch! It is true aslong as the first condition is not true. What if we want to let the user know they cant drive...



```
age = int(input("How old are you?"))
if age >= 17:
    print("You can drive!")
else:
    print("Sorry you can not drive")
```

Lets create a scenario where the else action prints "I like pizzza!"...
Would the following programs print "I like pizza!"? Yes or no?
Tick the box if **yes** and put a cross if **no**!

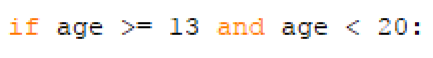
Example:-



```
if num < 45:
```

Input = 56 ☑ Input = 20 ☒

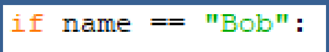
1)



```
if age >= 13 and age < 20:
```

Input = 14 □ Input = 20 □

2)



```
if name == "Bob":
```

Input = "Wilfred" □

Exercise 3

Each task shows an example of how your output of the program should look like. Save files as 'Task1.py', 'Task2.py' etc so they can be marked!

1) Write a program that follows exercise 2 task 4 but uses one else IF statement instead of two IF statements.

2) Write a program that checks if a number entered by the user is positive or negative as efficiently as you can. Print either positive or negative.

3) Write a program to check if a person is eligible to join a club by first taking their name and then their age. The eligible ages are 15, 16 and 17. If the person is eligible, print their name and state that they are. If not then print a sorry message with their name.

Hint: You need to use the "and" operator!

Help with Python!

Syntax	Example	Description
print()	print("Hello world!")	Prints to screen
input()	input("how are you?")	Waits for user input and optionally prints a string to the screen.
Boolean Operators		
==	a == b	Returns true if a and b are equal.
!=	a != b	Returns true if a and b are not equal.
>	a > b	Returns true if a is greater than b.
<	a < b	Returns true if a is less than b.
>=	a >= b	Returns true if a is greater than or equal to b.
<=	a <= b	Returns true if a is less than or equal to b.
and	a<b and a<c	Returns true if both sides of the and are true.
or	a<b or a<c	Returns true if at least one side of the or is true.
String Operations		
lower()	name = name.lower()	Changes the string to all lowercase.
upper()	name = name.upper()	Changes the string to all uppercase.
len()	len(name)	Returns the amount of characters in a string.
Arithmetic Operations		
+	a + b	Returns a plus b.
/	a / b	Returns a divided by b.
*****	a * b	Returns a multiplied by b
%	a % b	Returns the remainder of a divided by b.