

Data Academy

Data Science
Python

Module 1

TECH TALENT
ACADEMY |

What is a Computer System?

A computer is a system that takes a set of digital inputs, processes them, and creates an output.

This is done using a combination of hardware and software



What is a Program?



Whenever we use a computer, phone or tablet to undertake any activity, this has been facilitated using programming. This includes operating systems (Windows, Mac OS, Linux), browsers and any apps that you use, to name just a few.



As users, you may often interact with a front-end user interface (UI). “Underneath” this user interface, lines of code are being run to enable the computation to perform the necessary operations.

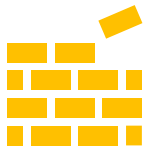
The sequence of **instructions** that specifies how to perform the computation is known as a **program**. These sets of instructions are written by **programmers**.

Programming is not telling the computer what you want to do. It is telling it **EXACTLY** how to do it.

Syntax



The **syntax** of a **programming language** is the set of rules that apply when writing in that language.

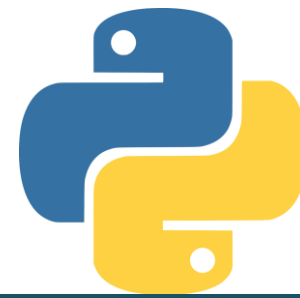


Syntax is made up of special key words, characters, and symbols which are combined to result in a program.



It is important to remember that **ALL** syntax needs to be correct for the program to run.

History of Python



- Python is a widely used, programming language. It was originally designed by Guido van Rossum and released in the early 1990's.
- It came off the back of the ABC language.
- The name was inspired by “Monty Python”.
- Known for its simplistic, concise and modular approach, it gained momentum from the start and is now one of the most popular programming languages.
- Particularly useful for data analysis.
- Currently on Python 3.

Which is the **easiest** to program to say “Hello World”?

```
org 100h
main proc
    mov  ah,9      ;
    mov  dx,offset hello_message
    int  21h      :
retn
hello_message db 'Hello, world!$'
main endp
end main
```

1. Assembler

2. C++

```
#include <iostream>
int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

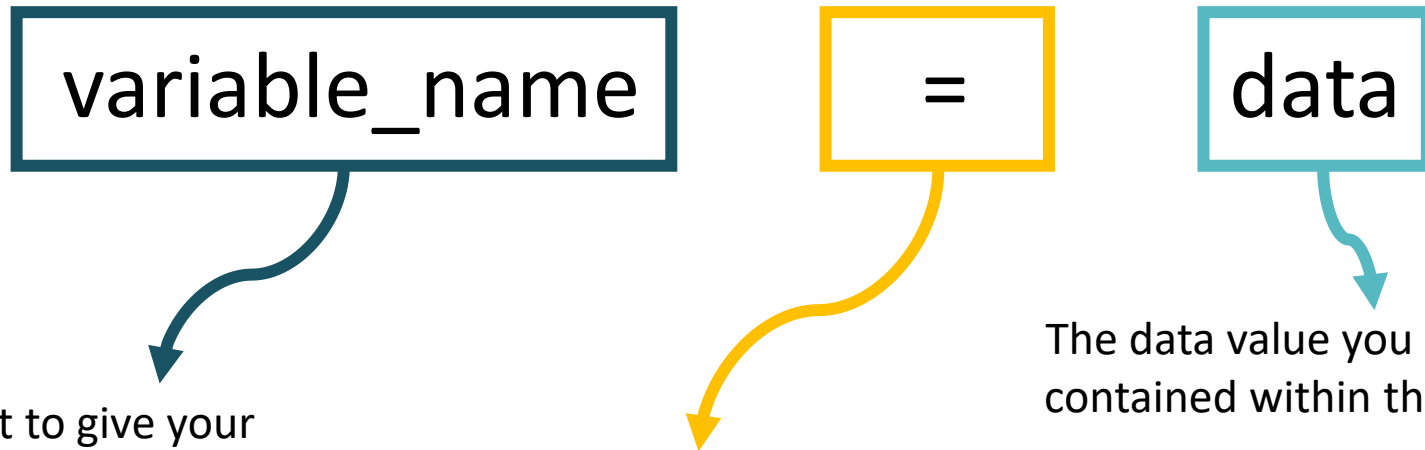
3. Python

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

Variables

In many cases, we need to be able to store data within our program. One way to do this is to use a **variable**. The best way to think of a variable is as a container for a piece of data.

Syntax:




The name you want to give your variable.

Using the equals sign between your variable name and your variable data tells you program that you want to assign the data to the variable.

The data value you wish to be contained within the variable.


Variables

Variables must have an identifiable name. It is best practice to ensure that your naming convention is consistent.



```
user_name      = "Andy"  
price_per_litre = 1.14  
  
userName      = "Donna"  
pricePerLitre = 1  
  
UserName      = "Suzannah"  
PricePerLitre = 1.12
```

Data held within **variables** can change while the program/algorithm is running.



```
user_name      = "Andy"  
print("user_name =", user_name)  
  
user_name = Andy  
  
user_name      = "Satinder"  
print("user_name =", user_name)  
  
user_name = Satinder
```


Data Types

Data types tell your interpreter (and therefore your program) how you intend to use the data.

In a program you may wish perform certain operations on your data, such as:

- conducting a calculation
- displaying a series of words to a user
- setting up a data structure

In order to ensure that our programs can perform these operations without issue, it is essential that we assign the correct data type to our data.



Data Type	Description	Example
String	Strings hold a list of characters, must use single or double quotations when entered	"iPhone 12"
Integer	A whole number without a decimal point can be positive or negative	0,3,2560,67,57
Float	A number which includes a decimal point can be positive or negative	2.5, 5.355
Boolean	A Boolean variable can only have one of two values, TRUE or FALSE	reply = true

TTA

2 is NOT the same as **2.0** which is NOT the same as **"2"**

Data Type	Description	Example
List	A list object is an ordered collection of one or more items of data, this does need to be the same type, but must be put in square brackets.	[1,2,3,4]
Tuple	A Tuple object is an ordered collection of one or more items of data, this does not need to be of the same type but must be put inside parentheses.	(1,2,3,4)
Dictionary	A dictionary object is an unordered collection of data in a key:value pair form. A collection of such pairs is enclosed in curly brackets.	{1:"one",2:"two",3:"three"}

Can you identify what the correct **data type** should be for each piece of mobile phone data stored?

Variable	Example Data	Data type?
Item Name	"iPhone 11"	<input type="text"/>
Number in stock	35	<input type="text"/>
Price	36.6	<input type="text"/>
Need to order	True/Yes	<input type="text"/>

Syntax Errors

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

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

```
print "Hello World!"
```

```
print Hello World
```

```
PRINT ("Hello World!")
```

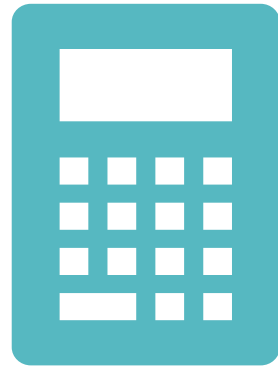
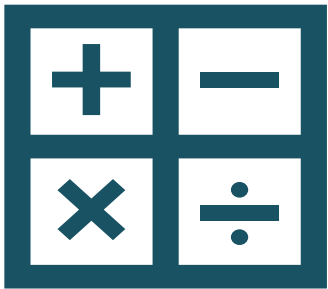
```
print (a + b)
```

```
print (5 + 6 / 7 +)
```

- ▶ **Syntax** is CRITICAL in programming. A computer cannot take an educated guess at your instructions.
- ▶ Computers just follow exactly what you tell them to do.

Mathematical Expressions

Python can perform mathematical computations easily. In order to perform the calculation, you must ensure that your data is set up as a numeric type.



```
print(2+2+2+2+2)
```

```
10
```

```
print(8*6)
```

```
48
```

```
print(10-5+6)
```

```
11
```

```
print(2+2)
```

```
4
```

```
print (10/2)
```

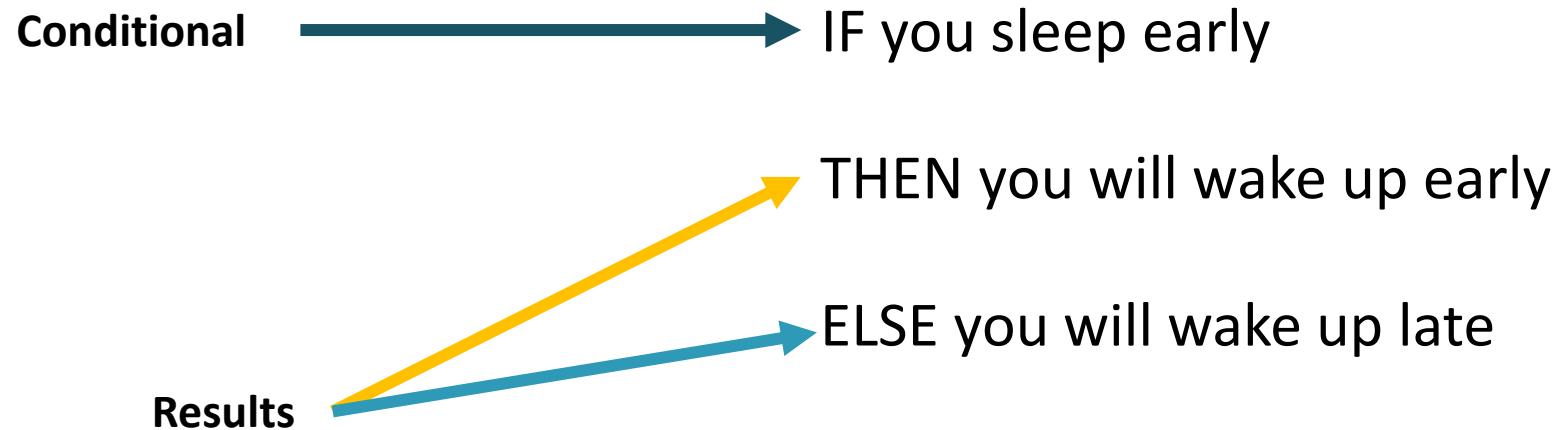
```
5.0
```

Comparison Operators

Operator	Description	Python example
<	Is less than	<code>if age < 12:</code>
<=	Is less than or equal to	<code>if age <= 12:</code>
>	Is greater than	<code>if age > 12:</code>
>=	Is greater than or equal to	<code>if age >= 12:</code>
==	Is equal to	<code>if age == 12:</code>
!=	Is not equal to	<code>if age != 12:</code>

Selection

Selection is used to choose between two or more options in programming we use an **IF STATEMENT**



IF and ELIF

In addition to the **IF** statement, you can use **ELIF** to continue to build conditional statements into your code underneath your original IF statement.

IF Condition 1 met:
 Perform operation 1

ELIF Condition 2 met:
 Perform operation 2

ELIF Condition 3 met:
 Perform operation 3

ELSE:
 Perform operation 4

TTA |

```
if 1 == 2:  
    print("1 = 2")  
elif 2 == 2:  
    print("2 = 2")  
else:  
    print("The values are not the same")
```

2 = 2

Can be done with variables:

```
if user_name == "Donna":  
    print("Donna is your trainer.")  
elif user_name == "Suzannah":  
    print("Suzannah is your trainer.")  
else:  
    print("Andy is your trainer.")
```

Andy is your trainer.

User Inputs

User inputs allow you to ask a user for an input, and then store that input in a variable.

To set up a variable as requiring an input, you need to specify that you wish to record a user input.



```
user_input = input("What is your name? ")
```

When you run this, you will note that there is a text box that appears, enabling the user to type and store their details.




```
user_input = input("What is your name? ")  
What is your name? Joe Bloggs
```

In the above case, the variable **user_input** will hold the data **Joe Bloggs**, based on the data entered by the user.

Casting

Casting is the conversion of one Data Type to another. So converting an Integer into a String or vice versa!


```
into_int    = int("23")  
into_float  = float("23")  
into_str    = str(23)
```



To cast data as a specific data type you must specify the type outside of parentheses containing the data.

int = integer
float = float
str = string

When asking for **user input**, the default data type will be **string**. For example, asking a user their age, the response will be stored as a string unless you specify that you want to record this as an integer.



TTA |

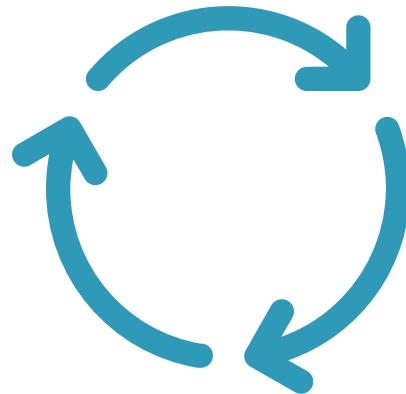
```
user_input_age = int(input("What is your age? "))  
print("In two years time you will be...", user_input_age + 2, "years old")
```

Iteration

When you run your program, this will usually run from the first line of your code to the last line of your code.

A **loop** allows you to run through a particular section of your code until a specific condition is met.

For example, if your program allows a user to enter a password, you will not want to exit the program straight away after someone gets their password wrong once. This could often be as a result of simple human error. Most programs recording passwords will give the user a set amount of goes to try and enter their password correctly. This is done using a loop.



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

To create a comment within your code, you need to use the # symbol. Your program will ignore the comments and will run everything else.



```
# Gather the user's name and store as variable  
user_input = input("What is your name? ")  
  
# Display name and greeting to user  
print("Hello", user_input + "!", "Hope you are well!")
```

Home Learning Tasks



Task 1

Write a program that does the following:

- a) Stores a random number (1-10) in a variable – see hint below.
- b) Asks a user for their name and stores this in a variable.
- c) Asks a user to guess the number between 1 and 10.
- d) Tells the user whether they have guessed correctly.

HINT!

You'll need to import the random library using the following:

```
import random
```

To generate a random number between 1 and 10, you need to use the following piece of code:

```
random.randint(1, 10)
```



DON'T FORGET YOU NEED TO STORE YOUR RANDOM NUMBER IN A VARIABLE!

Task 2

Write a program that asks a user for their favourite number between 1 and 100 and then tells them a joke based on the number. You should use a minimum of 3 jokes.

Task 3

Write a program that allows user to enter their favourite starter, main course, dessert and drink.

Concatenate these and output a message which says – “Your favourite meal iswith a glass of....”

Task 4

A motorbike 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.

Task 5

Write a program which will ask for two numbers from a user. Then offer a menu to the user giving them a choice of operator:

e.g. – Enter “a” if you want to add
“b” if you want to subtract

Include +, -, /, *, ** square (to the power of). Once the user has selected which operator they wish to use, perform the calculation.

Task 1 - Solution

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

Task 1 – Solution Breakdown

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.")

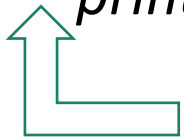
guess = int(input("Take a guess."))

Converts the input from text to an integer

if guess == number:

The == is *equal to*

print("Good job, " + myName + "! You guessed my number")



TTA

Must be indented

TECH TALENT
ACADEMY