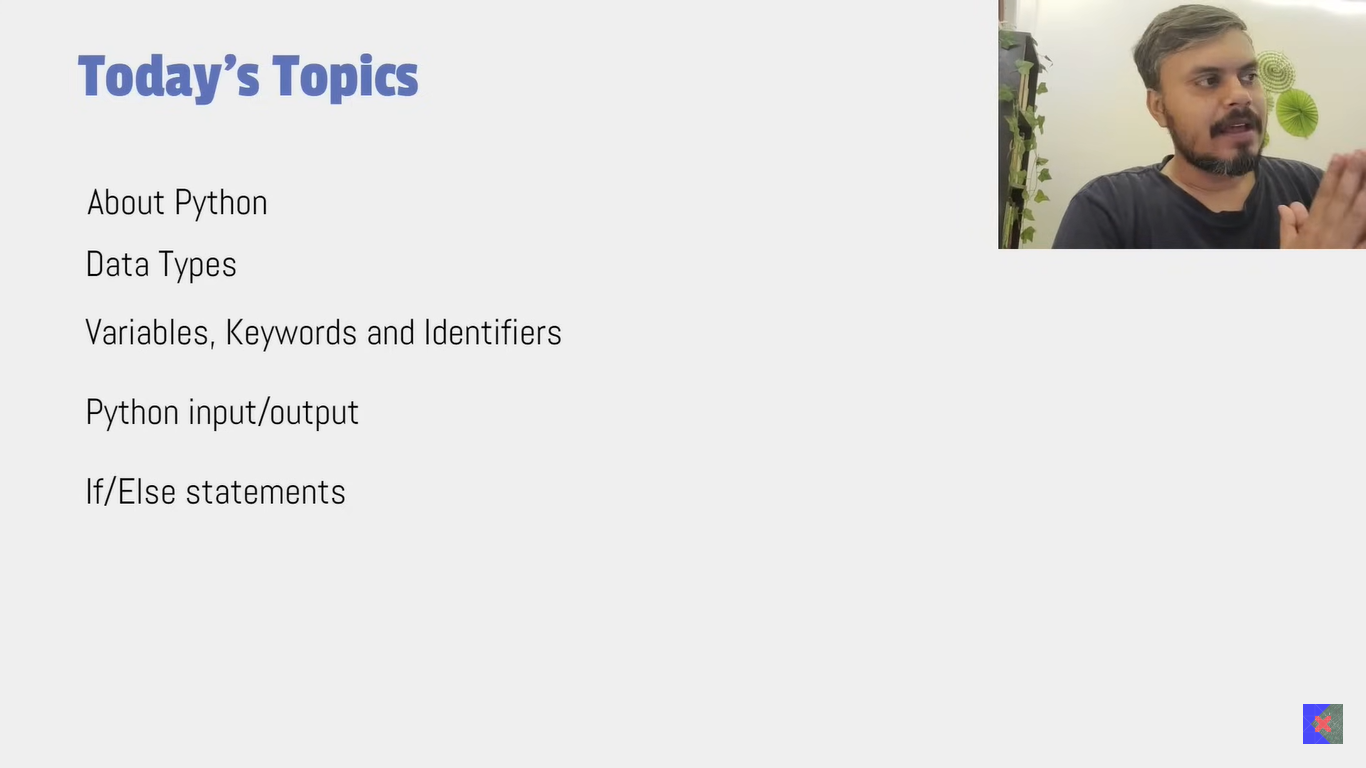
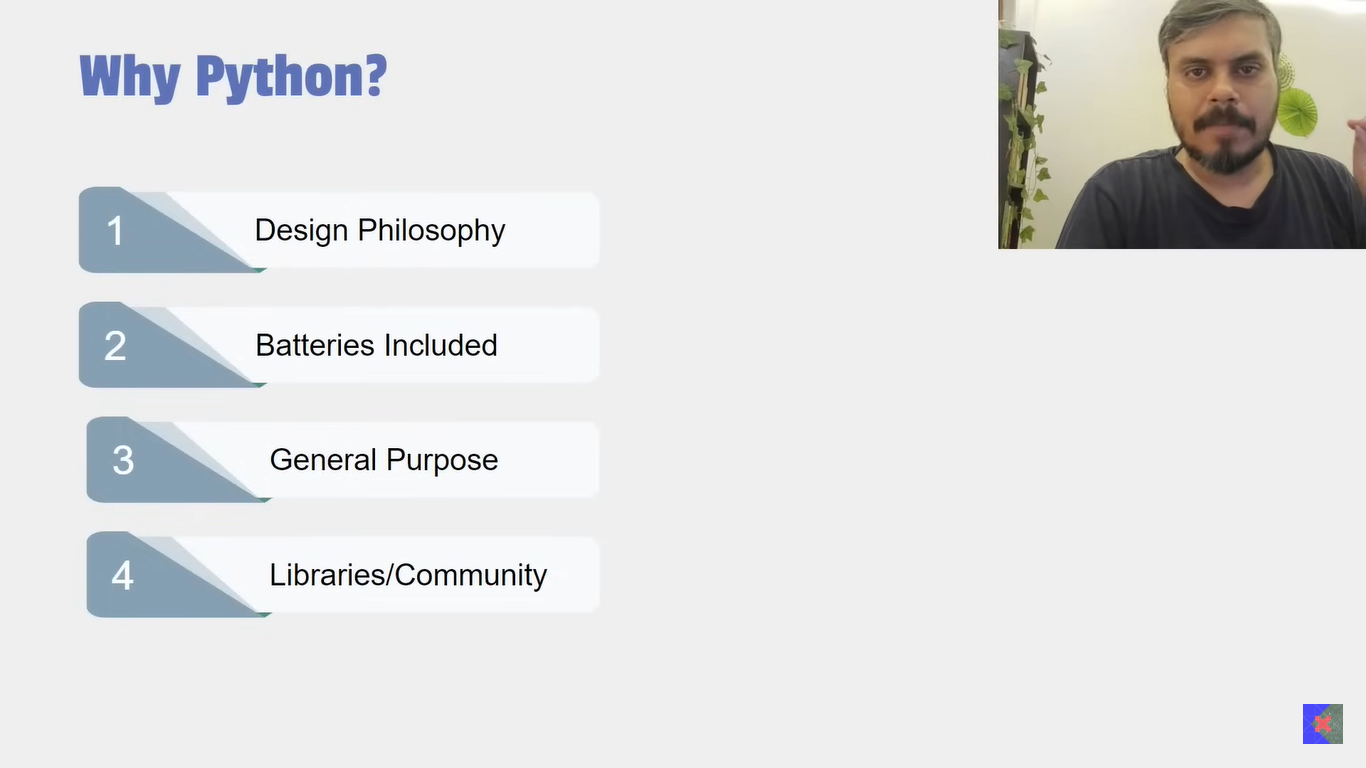
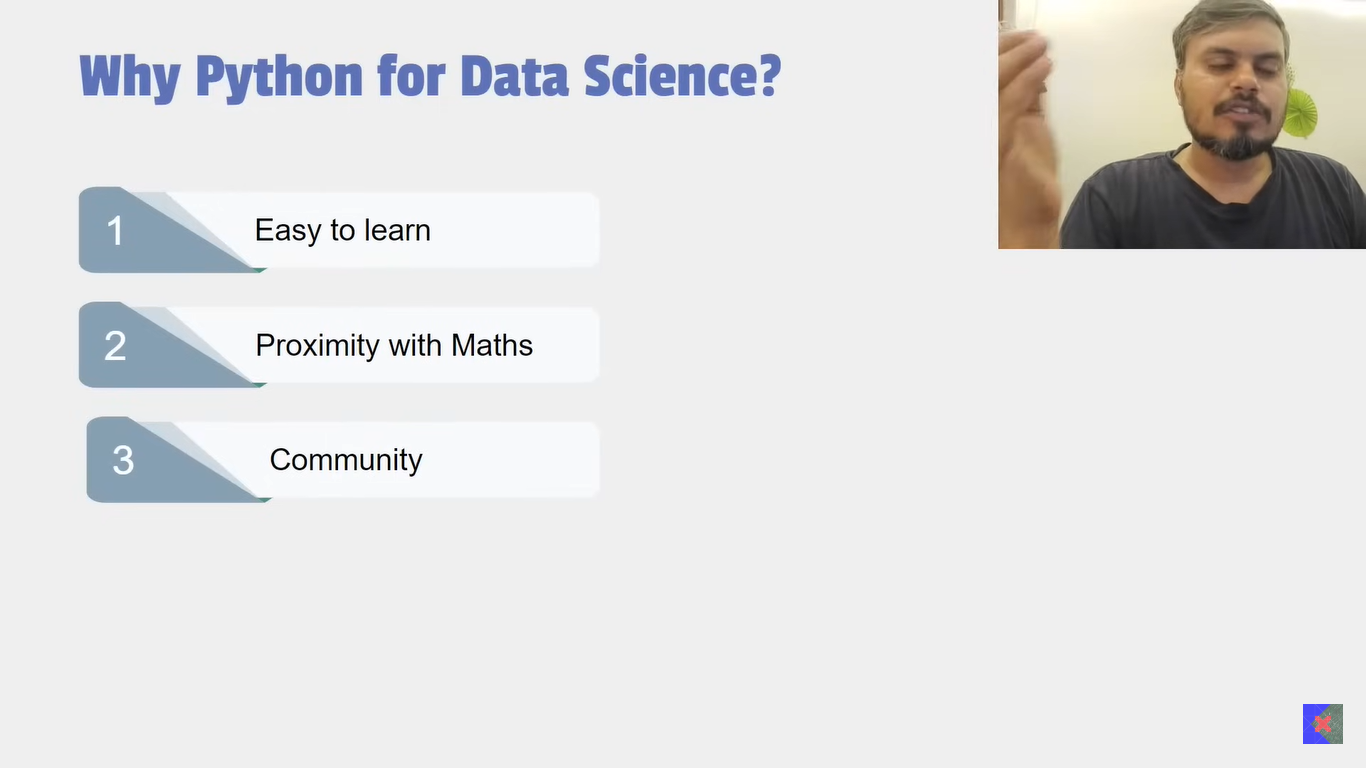
* For each session, we will have **Tasks** which will contain 20 problems that we need to solve.
* We will code in Google collab which is an online interpreter for python.
* 
* 
* 
* One thing to notice that Python is comparatively slower w.r.t C/C++. But the solution to this which Python has found is: Let’s say we have Numpy there we have something called ND\_Array which is written in C++ to support faster processing.
* Now we will learn to write our first program where we will print Hello World..

To do this we will use a built in function called **print**. Built in functions are those which are already predefined. Now to identify a function is to see if it has (). If it has so then it is a function.

Ex: X() -> In these braces, you can give anything as a input and it will give you a output.

* Please understand that Python is a **Case Sensitive** language which means print and Print are not same.
* To Print Hello World:

1. print(‘Hello World’)
2. print(‘Salman Khan’)
3. print(Hello World) // This will throw an error as it does not know what is it which you have passed inside the function.

* You not only can print a string but also any other data type like: integer, decimal, Boolean etc.
  + print(7) //7
  + print(7.3) //7.3
  + print(True) // True [Boolean]
* Also in a print function you can multiple things at the same time:
  + print(‘Hello’,7,7.3,True) // Hello 7 7.3 True
  + If you notice in the above function all the values are separated by a space. This is because in print function we have a parameter called **sep** which is by default a space character. Therefore if you want to change the default space with a front slash. It can be done like: print(‘Hello’,7,7.3,True,sep = ‘/’) **output**: Hello/7/7.3/True
* Let’s say I write the following:
  + Print(‘Vicky’)
  + Print(‘Jha’)
  + Output:
    - Vicky
    - Jha
  + This is because in print function we have another parameter as **end** which by default ‘\n’ which means a new line. So let’s say I want to print it like : Vicky-Jha
  + In that case:

We can write it like:

Print(‘Vicky’,end = ‘-’)

Print(‘Jha’) // Vicky-Jha

* Datatypes: It means the kind of data python supports.
  + Integer
  + Float/decimal
  + Boolean
  + String
  + Complex Number
  + None
  + List
  + Set
  + Tuple
  + Dictionary

For details, Please have a look in the google collab notebook and see 2nd Topic: Datatypes

* To detect a datatype of a variable, wehave type() function:

Ex: type(3) //int

Type(3.0) //float

Type(“True”) // string

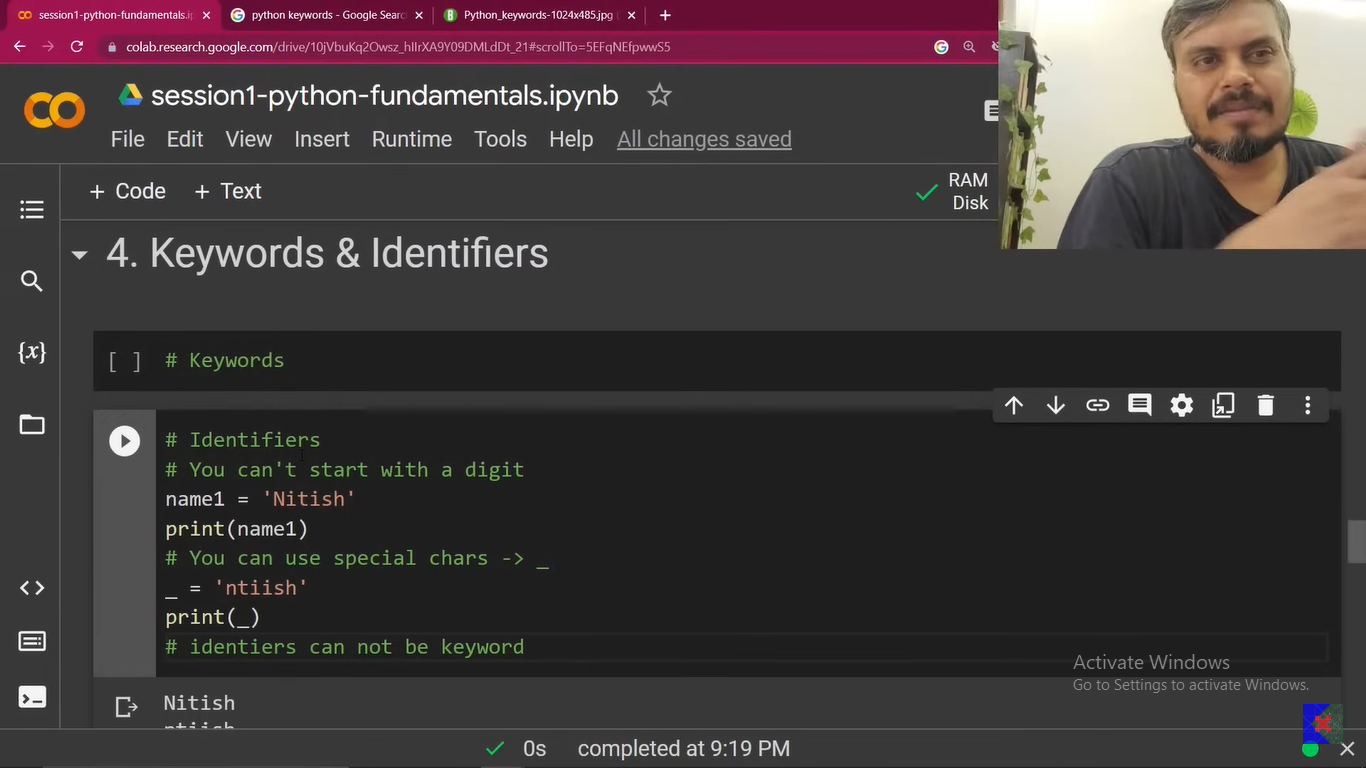
And so on.

* Variables: It is a container which stores some data. Variable is used when we want to use it for future.

What I meant by that is let’s say, user logging to website -> we want to show Hello {user\_name} which we don’t know now. Here we will use variable.

* In Python we need not to declare a variable. We directly create it whenever we need it.
  + For Example: name = “Vicky” Here we have directly created a variable called name. we need not to tell what kind of value it will store. It will automatically understand.
* **Dynamic Typing and Static Typing**:
  + **Dynamic Typing** is while creating a variable you are not defining any data type of it. For example: a=5. If you see here, for variable ‘a’ we have not defined any data type from the value it is referring that it is an integer. This is called Dynamic Typing. Python supports Dynamic Typing.
  + **Static Typing** is where you have to define the datatype also while creating a variable. For example: int a =5; Java/C/C++ supports static typing.
* **Dynamic and Static Binding**:
  + **Dynamic Binding**  is let’s say there is a variable:
    - Name = ‘Vicky’
    - Now after sometime, I can make it to let’s say: Name = 123
    - This is supported by Python which means a variable in a program can store any value. This is dynamic Binding.
  + **Static Binding** is let’s say there is a variable.
    - **Int a =5;**
    - Now in future the variable ‘a’ can store any int value but it cannot store any other datatype value let’s say string. So throughout the program it has to store any integer value only.
* **Different ways of defining a variable**:
  + a,b,c = 1,2,3  
    print(a,b,c) // 1 2 3 [Here we defining multiple variables at once.]
  + a=b=c=5

print(a,b,c) // 5 5 5 [To assign same value to all the variables]

* **Keywords**:
  + Please understand the background: when we write a code, we write in simple English language i.e high level language but a machine doesn’t know English it only knows machine level language which is binary language i.e 0 or 1.
  + So the process of converting the high level language to the binary language is called Compilation/Interpretation. Only difference between compiler and interpreter is compiler converts the entire code from high level to machine level language at once whereas interpreter does it line by line. Interpreted language ex: Python. Complied Language: Java.
  + So while the process of converting the high level language to a low level language machine reserves certain words that is used by them to identify what is required to be done. For ex: if, else, True, False. These reserved words are called **Keywords.**
  + There are total of 32 keywords in Python.
* **Identifiers** : It is basically the ones you create and give a name to. Let’s say you create a variable, function or a class you give a name to it that is called an identifier.
  + **Rules to create an identifier**:
    - 
* # take input from users and store them in a variable and add them. Code is already there in the google colab notebook.
* 1 thing to understand here is while we use input function to take data from user by default it is in string and you use type conversion to convert it to the integer and then add them.
* In case of Type Conversion, we have basically 2 types of type conversion:
  + Implicit Type Conversion: Ex: 5+6.5 = 11.5 which means ideally int and float addition should not have been allowed but here Python is intelligent enough to understand that in real world it is possible hence it is allowing it. But let’s say you would have wrote: 5+’6’ in this case it would throw an error since int and str addition is not allowed.
  + Explicit Type Conversion: In such scenarios where you have to explicitly change the datatype of a variable then we call it a Explicit type conversion.

Example: a=‘5’

b=‘6’

print(int(a)+int(b)) //11

* Now understand the scenario:

a = input(“Enter first num”)

b = input(“Enter second num”)

result = int(a) + int(b)

print(result) // 11

print(type(a),type(b)) // str, str

You might have a question, when I converted a,b into int and then did addition then why its type is showing as str here.

That is because while storing the result you converted a and b to int and since string is immutable it created another copy of the same data in form of int and then did the addition.

Therefore original data is still in str.

However, if you do

a = int(input(“Enter first num”))

then it will permanently store integer as while taking the input you first converted it to an integer and then assign it to the variable.

\*\*It is really very important\*\*

Literals: The value you store to a variable is called a Literal.

The remaining details please find in the google colab notebook.