Data Types
Lesson Plan: Class 07 / PRG / 03

Overall goal of the lesson	**Children will get introduced to data types and will be able to use these different data types and various string operators**
Prior knowledge required	Basic input and print statements.

MODULE 1: **Module time:** 35 minutes

Goal:	To get children to differentiate between the different data types.
Description:	Children will be able to know how to input integer data and how to input string data. Also, children will be able to differentiate between integer and string datatype.
Material required:	**Physical:** Writing material to copy down the instructions written on the blackboard. **Electronic:** A computer with a python compiler (version 3.x) for each student and the teacher. If there is an internet connection, https://www.codechef.com/ide can be used. A projector with the computer screen of the teacher projected for the students to see.
Procedure Summary:	Teacher will demonstrate the difference through contrasting examples and activities.
Procedure Details:	1. Activity hook a. Start by writing the following and ask students to pick the odd one out. i. 1, 2, a, 10 ii. A , b, c, 23 iii. 23, 45, man, 16 b. Now ask how were they able to find the odd one out. (Expected Student Response: Some were numbers and others were letters/text so we could see the odd one out) c. Great, so you could see two things here 1. Numbers 2. Text d. 2. Strings a. String as it suggests is a sequence of characters or symbols surrounded by quotes (double or single). b. Can you help me find out the string data types here?

Example	ESR	Example	ESR	Example	ESR
“h”	Yes	“A#ejh”	Yes	“AbcDE”	Yes
“Ae6435”	YES	“11”	?	164265	?

c. The last second example “11”, is surrounded with quotes, so it is a string - it has no numeric value. It is like a symbol 11. Whereas, the last example 164265 is not surrounded with quotes, so it is considered as integer.
d. So anything written in quotes or taken as input is treated as mere symbols without considering the values.

e. We already learnt in our first lesson that any input that we take from the user using input() function is of string datatype. Example: When we take input from user using a = input(). Here the value inside variable a is of string datatype.
f. Ever wondered why we put quotes in several cases of print statement? e.g >>>print("Hello") , or print ('hello'); here we are telling the computer that Hello is a collection of characters, it's a string type so print it as it is .

3. Integers:
   a. Integers, as the term suggests, are similar to the whole numbers we use in mathematics. They can be positive or negative.
   b. Let's try to find out which ones come under integer datatype. Ask students to answer in YES / NO for whether the given value belongs to integer data type.

145	Yes	**-157**	Yes	**"2"**	No
3.57	No	**"A"**	No	**"ABC"**	No

   c. So, now when we have clarity about what comes in integer data type. Let's learn how to use it in Python.
   d. When we ask a user for input, we write **a=input().** Now, if we need the integer data of the input, we need to write it inside the integer conversion function, **int(),** so that the input gets converted to integer data type. So, we write **a=int(input()).**
   e. Note that the value that is inside the int() function, should be convertible to an integer. Example,
      i. a = int("3") - Here "3' is convertible to an integer. So the value inside a after the execution of the above statement will be 3.
      ii. a = int("3 4") - Here, "3 4" is not recognised as a valid convertible integer because there is a space in between the two. So it gives an error.

4. Floats:
   a. Now what about the decimal point number?
5. Can you take two inputs from a user , where one of them should be integer input . (ESR : var = input(), var2=int(input()))

6. Now consider these examples and tell me what would be the output ?

Print statement	Output (ESR)
>>>print(1+2)	3
>>>print('1+2')	1+2
>>>print('1'+'2')	12

   a. So, you see how putting quotes around the characters changes the way the characters are considered.

Assessment:	1. Input following values from a user: a. Two string values b. Two integer values c. One integer and two string values 2. Write a program which takes input and tells whether it is an integer or not. 3. Problem which uses float and integer data types [To introduce float() function] https://www.codechef.com/problems/HS08TEST
Information Broadcast:	

MODULE 2: **Module time:** 35 minutes

Goal:	To get the children to use various string operators.
Description:	Children will be able to use new line, escape characters, prefix of r, string concatenation and string repetition operators
Material required:	**Physical:** Writing material to copy down the instructions written on blackboard. **Electronic:** A computer with a python compiler (version 3.x) for each student and the teacher. If there is an internet connection, https://www.codechef.com/ide can be used. A projector with the computer screen of the teacher projected for the students to see.
Procedure Summary:	Teacher will demonstrate through examples.
Procedure Details:	**New line ("\n")** ● Let's say I want you to print the following on the screen. ● Try this out >>>print("Hello this is a long sentence and I want it to be visible to me as a paragraph because writing such long sentences is fun") ● What if I want you to edit it to look the output like this: Hello This is a long sentence And I want it to be visible to me As a paragraph Because Writing such long sentences Is Fun ● What will you do to get the above output ? (ESR : Write many print statements) ● Yes , that is one way of doing it but there is another way of doing the same . There are special characters in Python which have special meanings . ● One of such characters is "\n" (backslash n) . When computer encounters it in a string . Computer makes a new line. So if we write >>>print ("Hello\nWorld") , what will be the output ? Hello World ● Can you now get the following output using only one print statement ? Hello This is a long sentence And I want it to be visible to me

As a paragraph
Because
Writing such long sentences
Is
Fun

Escape characters

- Now I want you to print “I am studying about \n in the class”
  ESR : It is coming in the new line , We couldn’t print \n .
- So , sometimes we need to print these characters but computer thinks that we want to add a new line . In this case we need to use escape character (“ \”) . This character prints whatever is after the backslash (\) as it is without any changes .
- Try >>>print(“I am studying about \\n in the class”)

Prefix of r

- Try printing >>>print(“Test\new\never”)
- What does it print ?

ESR :Test
ew
ever

- We can use escape character to print the desired output . But, we’ll have to use escape characters with all the special characters then.
- There is another way of doing it . It is called **prefix of r** i.e. prefixing the statement with r ;
- '\n' will be treated as a newline character, while r'\n' will be treated as the character \ followed by n.
- When an 'r' or 'R' prefix is present, a character following a backslash is included in the string without change, and all backslashes are left in the string. For example, the string literal r"\n"consists of two characters: a backslash and a lowercase 'n'.
- String quotes can be escaped with a backslash, but the backslash remains in the string;
  - for example, r"\"" is a valid string literal consisting of two characters: a backslash and a double quote;
  - r"\" is not a valid string literal (even a raw string cannot end in an odd number of backslashes). Specifically, a raw string cannot end in a single backslash (since the backslash would escape the following quote character).
- Note also that a single backslash followed by a newline is interpreted as those two characters as part of the string, not as a line continuation.
- Try this >>>print(r “test\new\never”) it prints the following

Test\new\never

String concatenation and Repetition operator

- So, all of us know about basic arithmetic operators . And we have seen them work on integers.
- Let’s try them out on strings and see if they work.
- Let’s take two strings and an integer say ,
  a= ‘hello’
  b=’world’
  c= 5
- Now , let’s try these operators (+ , - , * , /) on them .

+	-	*	/

	<table><tr><td>**a+b (Works)**</td><td>a - b (Error)</td><td>a*b (Error)</td><td>a/b (Error)</td></tr><tr><td>a+c (Error)</td><td>a-c (Error)</td><td>**a*c (Works)**</td><td>a/c (Error)</td></tr></table> ● So , we can see that we can perform **a+b** which gives **HelloWorld .** That is it combines both the strings . This is called **string concatenation .** ● Also , we can see that we can also perform **a*c** i.e multiplying a string with an integer and it gives us integer times string , in this case **HelloHelloHelloHelloHello .** This is called **string repetition.** ● So, we can repeat a string by multiplying it with an integer.
Assessment:	● Print the following using one print statement. Hello I am writing this In one line . ● Print this output “Hello\new things have arrived” ● Print this output “I am printing \new things \never stop \never argue” ● Print “I am new here” using a= “I am” , b= “new” , c= “here” ● Print “Hello” 100 times . ● Solve the following questions : ○ A = “Hello” , B=”This” , C= “is” , D= “a” , E=’string’. Print the following : ■ Hello this is a string. ■ This is a string ■ a string ■ This is Hello ■ HelloHelloHelloHello ■ Hello this Is a String ■ Hello\n this is \n a string
Information Broadcast:	

Example	ESR	Example	ESR	Example	ESR
h	Yes	H	Yes	AbcDE	Yes
A#ejh	Yes	Ae6435	???	164265	???