| Sr. No. of Event / Activity | Day/Date | Year | Month | Time | Venue | Description of Activity |
|---|---|---|---|---|---|---|
| 1. | 26-27 | 2020 | July | 5:00pm - 7:00pm | Online | Intro to Python Programming |

**Details of participation in the event/activity**
**(Fill wherever is applicable or information is available. At least total need to fill)**

| No of students / faculty | COMP | ETRX | EXTC | IT | MECH | Total no of participants |
|---|---|---|---|---|---|---|
| **Day1** | | | | | | 180 |
| **Day2** | | | | | | 70 |

**Report of event/ activity**

**Event: Intro to Python Programming**

Every year, KJSCE Codecell hosts a huge Python workshop to help anyone get started with programming by learning the basics of Python. It is also useful for people who already know python since we host both Beginners and Advanced batches.

Why was the Python programming language chosen for this workshop? There were three main reasons.

First, Python has an easy syntax. You won't waste a lot of time memorizing the arcane syntax that other programming languages will present you. Things like declaring variables, printing operations, and running loops have a very easy and intuitive code.

Second, Python has excellent code readability. Its code is so "English-like" that even a non-programmer might be able to understand what the code's purpose is. Also, it uses indentation to define separate blocks of code. This further enhances its readability and makes the code look cleaner.

Third, Python has tons of useful libraries like NumPy, Pandas, and Matplotlib. It makes things like computation of multidimensional arrays, creating manipulable objects from simple data and creating publication-quality plots and figures very easy without writing separate code for it. This is also the reason why Python has become very popular in the last few years because of its usefulness in AI and ML related fields.

Hence, to get them started with Python would increase their interests in various complex fields of programming too. But Python's disadvantage is that it is a very slow language compared to C/C++ and may not be used in applications where time is critical.

The Introduction to Python Programming workshop was held over two days online in separate batches for Beginners and Advanced, which were further divided into two more batches to accommodate the large number of people. We had over 180 participants from all departments like Electronics and Communications Engineering, Mechanical Engineering and not just Computer and IT students. Python helped them become familiar with basic programming concepts very useful for them both now and in the future.

Participants were instructed to download the official Python IDLE for programming. They were also given the option to run the code online using repl.it in case they couldn't download the official IDLE for any reason. Which one to use for teaching, IDLE or Repl, was entirely up to the instructors.

Day 1:

We started with absolute basics as many of the attendees were new to programming. On the first day, we explained to them the difference between a compiled and interpreted language. We began with the concept of print statements and how it was used to output to the console. The participants used the print() statement in Python to print sentences, concatenate strings and perform basic calculations. While concatenating and printing various things people noticed that they were not able to concatenate a sentence and a number(unless the number was in quotes). And hence various basic data types like Integer (int, numerical value without a decimal point), Floating Point (float, fractional numbers), String (str, sentences) were introduced. The two new data types for those who did a bit of programming were Boolean and None. Boolean (bool) can hold only one of two possible values: True and False. The last type was None which is a special type indicating nulls.Along the way, we also introduced the participants to the concept of variables.

Many of those who had experience with C++/Java remarked how easy it was to declare a variable in Python(without specifying its data type) and change its data type fluidly. The reason is that Python doesn't actually "hold" any data but is merely pointers to data objects. We then moved on to the basic assignment and arithmetic operations in programming. Many of them were new to the modulo (%), the floor division (//), and the power of (**) operators. We then gave them a simple problem to solve for some practice using the things they learned.

Comparison operators were then taught along with conditional statements, its syntax, working and how the or, and operators were used in it. Next, we moved on to looping statements. While being taught for loop, they were also introduced to the range() function in Python along with its start, stop and step parameters, which work exactly the same way in string slicing, also explained at the time.

In the end, participants were given a few questions to solve, going from beginning to intermediate and they were encouraged to ask any doubts since we were nearing the end of the session.

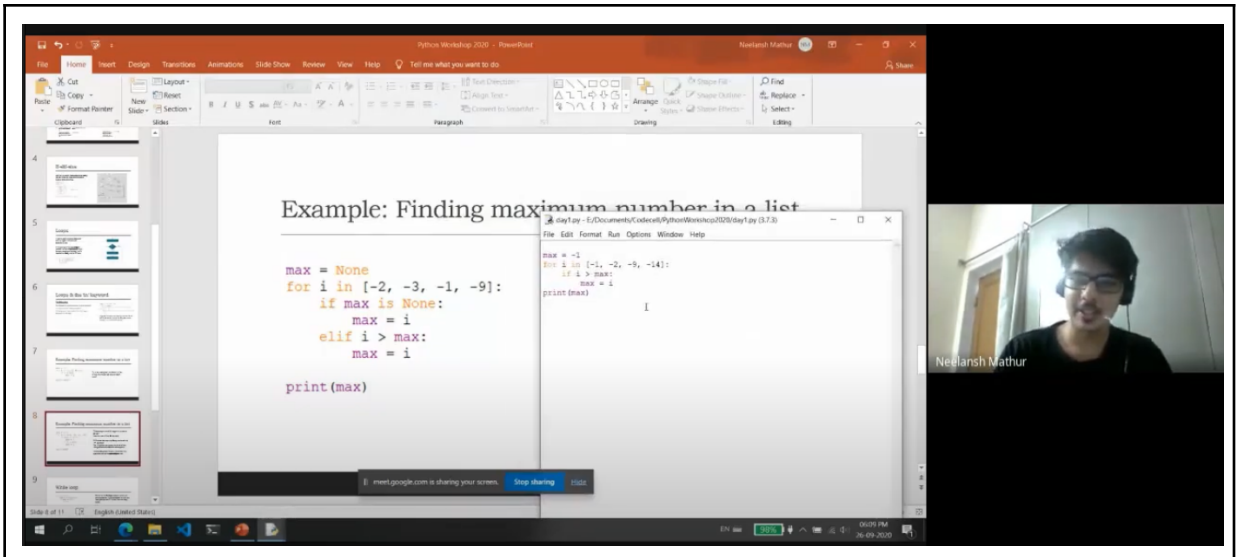
With this, we concluded Day 1.

Day 2:

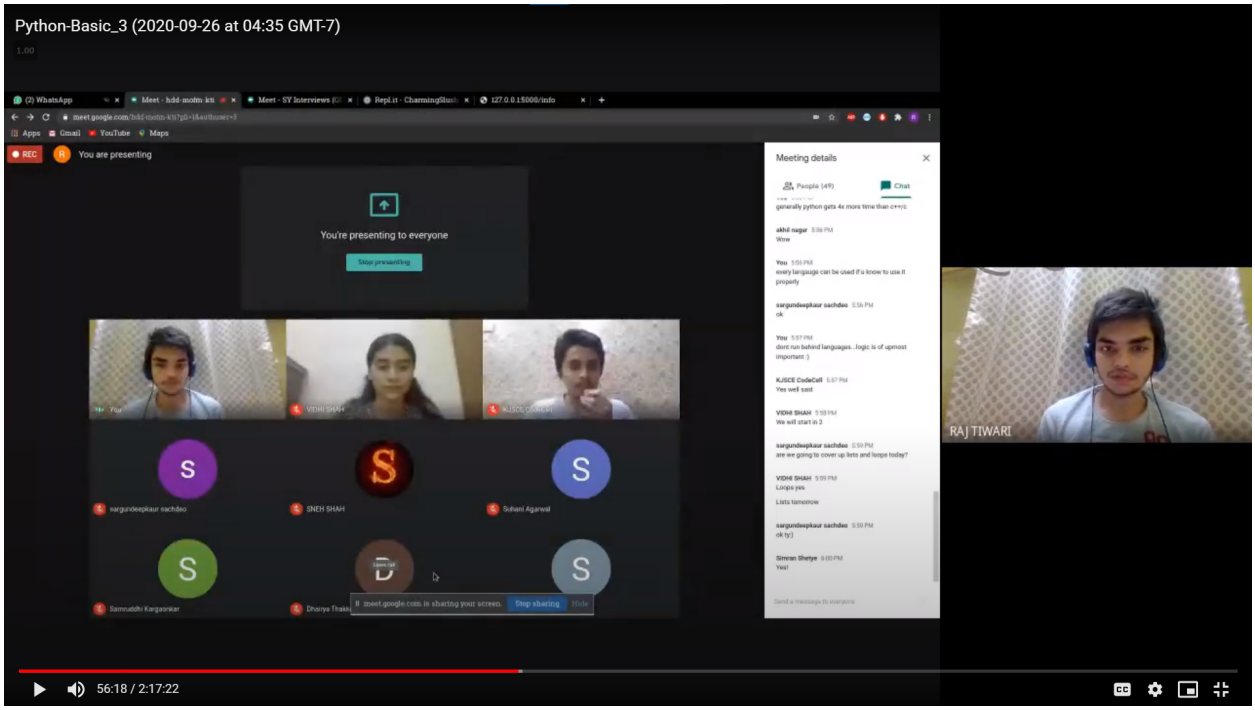
It began with revising and solving any doubts from the previous day. Then we commenced the session with built-in data structures of Python (lists and dictionaries). The list is an ordered collection of data in Python. Its syntax,indexing, working and basic operations like len(), append(), sort(), and list concatenation were taught. They were also introduced to the map() function and list slicing. Next, we moved on to dictionaries in Python and its key-value structure was explained. Since dictionaries make use of Hashmaps, searching in dictionaries is faster than lists. Different dictionary functions like .keys(), .values(), .get(key), etc. were taught.

When functions were taught, the participants immediately remarked on its usefulness and how it makes code reusable. Functions are majorly useful for their recursion property. Many of them were a bit confused about this concept but they understood it with examples such as calculating the factorial of a number, Fibonacci numbers, Euclidean Method of calculating GCD which also gave them a much better understanding of the importance of recursion.Last but not least, the participants had to give a small MCQ-test to be eligible for a certificate for this workshop. It was overall an excellent result with almost all scoring above merit.We later gave them a short introduction on Competitive Programming(CP) and took them on a tour on the Codechef website. We chose this CP website since CodeCell is a campus-chapter of Codechef. A very basic problem was solved by all our contestants on the site and everyone managed to get their first green tick.We received very positive reviews from all our participants. They liked the choice of language, the course content and the hands-on approach of our workshop. They were very happy to have a head-start for programming. The team was overwhelmed by the response and the feedback and couldn't have been more thankful to all the freshers who showed great enthusiasm in our first workshop itself.

## Sample Photographs of the Event/Activity




python_advance
print(l4)
['odd', 'Even', 'odd', 'Even']
x y z n
1 1 1 2
(i,j,k)
i+j+k != n
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]
list[[]]
['Even', 'Even']
# map
# map(function,iter)
x=1
y=1
z=1
n=2
print ([[a,b,c] for a in range(0,x+1) for b in range(0,y+1) for c in range(0,z+1) if a + b + c != n ])
[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]
NISHIT PATEL


Python-Basic_3 (2020-09-26 at 04:35 GMT-7)
You are presenting
You're presenting to everyone
Meeting details
RAJ TIWARI
56:18 / 2:17:22