



# KINGS ENGINEERING COLLEGE

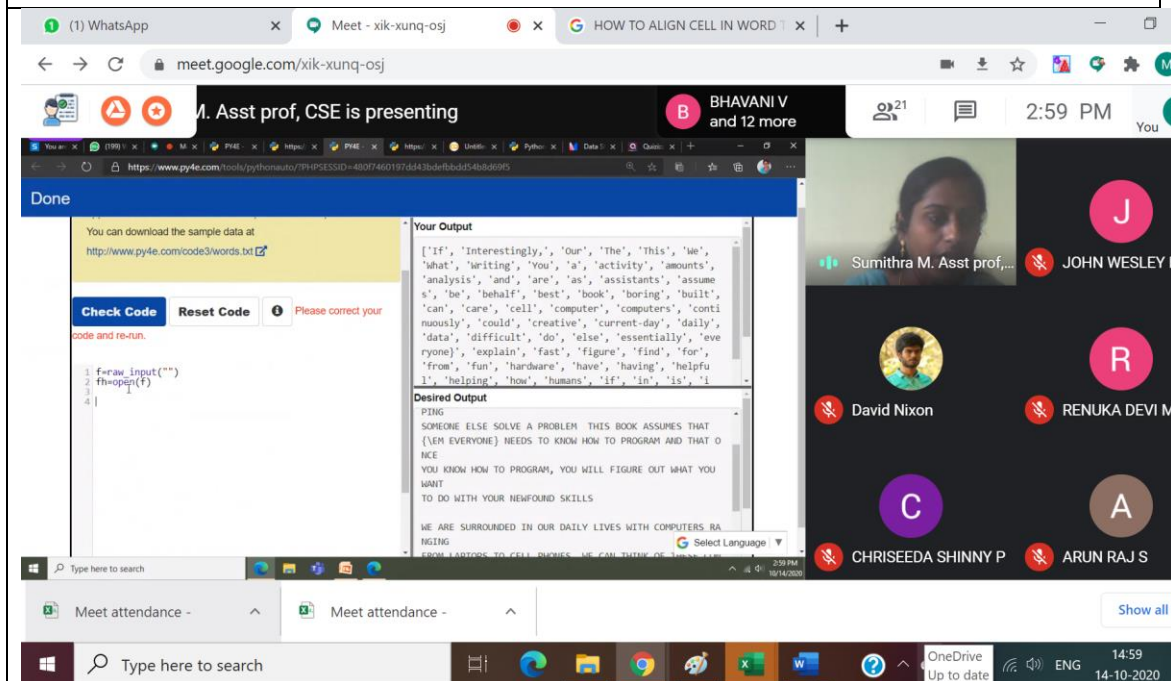
Chennai-Bangalore Highway, Irungattukottai, Sriperumbudur, Chennai - 602 117.  
Ph: 044-71224401 - 08. Fax: 044-71224410

ACCREDITED WITH NAAC & NBA, AFFILIATED TO ANNA UNIVERSITY and APPROVED BY AICTE

## EVENT REPORT

Name of the CLUB	STEAM – CODE CLUB
Name of the Event	Webinar on “Python Datastructures”
Date and Time of the Event	14.10.2021
Location	Online - GoogleMeet
Name of the Coordinator	Mrs.Gracia Nissi S.,AP/CSE
Resource Person Detail	Mrs. M. Sumithra, VelTech Rangarajajn Dr.Sagunthala R&D Institute of Science and Technology, Chennai.
Target Audience / Beneficiaries	II, III and IV year students of CSE,IT and ECE Departments
Number of Participants	43
Objective of the Event	To impart technical knowledge on Python Data structures
Outcome of the Event	Able to use the Data Structures in Python Applications.

### Photo gallery (Minimum 5 Photos)



COORDINATOR

PRINCIPAL

## Programme Schedule:

Event	Time	By
Welcome Address	02.00 – 02.05PM	Mrs.Gracia Nissi S. Code Club Coordinator
Training	02.05 – 03.50PM	Mrs.M.Sumithra
Feedback	03.50 – 03.55PM	Participants
Vote of Thanks	03.55 - 04.00PM	Mrs.Gracia Nissi S. Code Club Coordinator

**Technical Support : Mr.Palani, System Admin.**

## Additional Photo:

The image is a screenshot of a computer screen during a Google Meet session. The main window displays a presentation slide titled "Python Data Types" with a central green cylinder labeled "Python Data Types" and arrows pointing to various data types: Strings (example: 'python', "python"), Numbers (example: 10, 10.5, 10+5j), Lists (example: ['python', 'website']), Tuples (example: ('python', 'website')), Dictionary (example: {'name': 'python', 'number': 1}), Sets (example: {1, 2, 3}), and Boolean (example: 0, 1, True, False). To the right, a WhatsApp chat window is open, showing a list of participants including RAMANAN S, RICHARD D, SOLOMON SUNDAR SINGH R, Sumithra M. Asst prof, CSE (Pr...), Sumithra M. Asst prof, CSE, VIGNESH K, and VIGNESH L. The bottom of the screen shows the Windows taskbar with the search bar and several application icons.

## Registration Detail

Timestamp	Name of the Student	Register No	Year	Department
10-10-2020 10:33:11	K.MariGanesh	210818205018	III	IT
10-10-2020 10:33:20	SOLOMON SUNDAR SINGH R	210817205034	IV	IT
10-10-2020 10:33:41	Rama Lakshmi	210817104057	IV	CSE
10-10-2020 10:37:05	Anusha.C	210817205002	IV	IT
10-10-2020 10:37:09	Austin Emmanuel	210817205005	IV	IT
10-10-2020 10:37:18	Varsha.v	210818205301	III	IT
10-10-2020 10:37:25	John Hawa	210817205014	IV	IT
10-10-2020 10:37:40	SHADRAC GEORGE THIBBLE	210817114047	IV	MECH
10-10-2020 10:40:34	Shalini R	210817205033	IV	IT
10-10-2020 10:49:07	G.Janagamaharajan	210818205009	III	IT
10-10-2020 10:53:27	J.Jensy	210819106020	II	ECE
10-10-2020 10:54:32	V. Naveenkumar	210819106046	II	ECE
10-10-2020 11:11:24	E.MEENA KUMARI	210818205019	III	IT
10-10-2020 11:11:53	Thangeswaran	210818104044	III	CSE
10-10-2020 11:13:13	Jayalakshmi.L	210818104015	III	CSE
10-10-2020 11:14:26	Janani Rajesh	210819104032	II	CSE
10-10-2020 11:33:15	BHAVANI V	210818104008	III	CSE
10-10-2020 11:34:55	B.kiruthika	210818104020	III	CSE
10-10-2020 11:39:51	ANNIE.P	210818104006	III	CSE
10-10-2020 11:46:53	Megala. V	210819104062	II	CSE
10-10-2020 11:56:41	N.Jenifer	210819104038	II	CSE
10-10-2020 12:16:49	SHANKAR G	210817104063	IV	CSE
10-10-2020 12:17:24	ROSHINI G	210818104032	III	CSE
10-10-2020 12:28:29	PAVITHRA. S	210817205021	IV	IT
10-10-2020 12:56:46	RICHARD D	210818104031	III	CSE
10-10-2020 13:10:43	Darcus Angeline	210819104016	II	CSE
10-10-2020 13:38:20	Vignesh k	210819106084	II	ECE
10-10-2020 13:38:54	Priya k	210819106055	II	ECE
10-10-2020 13:44:53	Anisha I	210819104008	II	CSE
10-10-2020 14:19:32	J. Briskilla	210817104015	IV	CSE
10-10-2020 15:05:04	John Wesley B	210819104042	II	CSE
10-10-2020 15:44:03	David Nixon Raj D	210819104017	II	CSE
10-10-2020 16:12:11	Mahalakshmi.S	210819104056	II	CSE
10-10-2020 16:39:35	L.Priyangathirisha	210819104078	II	CSE
10-10-2020 17:04:24	Sowmiya.R	210818205032	III	IT
10-10-2020 17:12:22	B.subisha	210819104110	II	CSE
10-10-2020 17:13:58	M.amaravathy	210818205002	III	IT

Timestamp	Name of the Student	Register No	Year	Department
10-10-2020 17:15:15	S. Kamali	210819104044	II	CSE
10-10-2020 17:17:38	B.subisha	210819104110	III	CSE
10-10-2020 17:18:17	B.subisha	210819104110	II	CSE
10-10-2020 17:22:40	RAMANI.S	210819104082	II	CSE
10-10-2020 17:27:46	Saathvikha	210819104086	II	CSE
10-10-2020 17:42:45	A.Joshini Martina	210817205016	IV	IT
10-10-2020 19:00:21	Athista Lakshmi Swathy D	210817104010	IV	CSE
10-10-2020 22:14:00	Pooja S	210819104070	II	CSE
10-10-2020 22:14:34	Santhoshini c	210819104093	II	CSE
10-10-2020 22:59:49	Vignesh. V	210819104121	II	CSE
10-11-2020 8:07:46	Jerlin.J	210819106021	II	ECE
10-11-2020 10:01:44	PRAVINA.A	210819104075	II	CSE
10-11-2020 10:02:21	PRAVINA.A	210819104075	II	CSE
10-11-2020 12:34:05	Santhana selvam chandru.V	210819205045	II	IT
10-11-2020 12:37:01	S. Sharmila	210819205053	II	IT
10-11-2020 14:25:41	Premkumar G	210819104076	II	CSE
10-11-2020 16:52:31	Nithish kumar.V	210819104069	II	CSE
10-11-2020 16:53:17	Nithish kumar.V	210819104069	II	CSE
10-11-2020 17:01:47	Neena	210819104066	II	CSE
10-11-2020 17:33:13	R. Nithish kumar	210819104068	III	CSE
10-11-2020 17:33:20	R. Nithish kumar	210819104068	III	CSE
10-11-2020 23:11:14	A. Sri Gayathri Devi	210819106073	II	ECE
10-12-2020 8:48:16	Rajavibhu k	210819104080	II	CSE
10-12-2020 8:49:00	Rajavibhu k	210819104080	II	CSE
10-12-2020 8:57:57	Srinithi.k	210819205057	II	IT
10-12-2020 10:02:49	Ramanathan.M	210819104081	II	CSE
10-12-2020 10:36:36	D.praveen	210819104074	II	CSE
10-12-2020 10:36:52	D.praveen	210819104074	II	CSE
10-12-2020 10:36:55	D.praveen	210819104074	II	CSE
10-12-2020 10:37:37	D.praveen	210819104074	II	CSE
10-13-2020 9:13:05	Evangelin. D	210819121018	II	BME
10-13-2020 9:14:35	NEDUMARAN K	210819125015	II	RAA
10-13-2020 10:14:51	Nevedha M	210817104049	IV	CSE
10-13-2020 17:07:50	Ramanan	210819106057	II	ECE
10-13-2020 17:10:00	Renuka Devi M	210819106061	II	ECE
10-13-2020 17:16:53	MUNI RAJ.N	210819106042	II	ECE
10-14-2020 8:54:21	Betsy Lenora. A	210819106009	II	ECE
10-14-2020 14:23:33	Kamesh s	210819104045	II	CSE
10-14-2020 22:18:04	M.U.SURESH RAJA PERUMAL	210819205058	II	IT
10-14-2020 22:18:14	M.U.SURESH RAJA PERUMAL	210819205058	II	IT

# KINGS ENGINEERING COLLEGE

## STEAM – CODE CLUB

### PYTHON DATA STRUCTURES TRAINING PROGRAMME ATTENDED STUDENTS LIST

14.10.2020

S.NO	NAME OF THE STUDENT
1	DAVID NIXON
2	JOSHINI MARTINA A
3	RAMANI S
4	RAGAVAN R
5	POOJA S
6	NITHISH KUMAR V
7	RAMANAN S
8	VIGNESH L
9	JOHN WESLEY B
10	BATHMA S
11	SAMUEL PETER A
12	RENUKA DEVI M
13	SIVASANKARI K
14	NANDHINI P
15	JANANI R
16	JOSHE RAJ SP
17	ANJU M
18	KANIMOZHI K
19	KAMESH S
20	JERUSHA MISHAL J
21	JOBIN MARK D
22	KANIMOZHI K

SNO	NAME OF THE STUDENT
23	MANIKANDAN M
24	KANIMOZHI V
25	JERLIN J
26	SAATHVIKHA G R
27	MEGALA V
28	MOHAMED ASHIK A
29	PRIYANGATHIRISHA L
30	A.B FRANKLIN
31	AMOSE S
32	BHAVANI V
33	CHRISEEDA SHINNY P
34	GAYATHRI SRINIVAS
35	RICHARD D
36	SINDHU S
37	SOLOMON SUNDAR SINGH R
38	VIGNESH K
39	BENNY FRANKLIN A
40	JENIFER N
41	JAYALAKSHMI L
42	PRIYA K
43	RAMA LAKSHMI S

## PROFILE

### SUMITHRA M.

#2/101, Thiruparuthikundram Colony,  
Sevilimedu Post, Kanchipuram – 631502

Phone : 9786429360

Email: [sumithra@veltech.edu.in](mailto:sumithra@veltech.edu.in)  
[blessfulsumi@gmail.com](mailto:blessfulsumi@gmail.com)

---

### Academic Journey:

- *Assistant Professor* in the Department of Computer Science and Engineering.
- *B.E. & M.E.* in Computer Science and Engineering with 83%(Distinction in both UG & PG) .
- *8 Years* of Academic Experience in Engineering Colleges.
- **Currently working for Vel Tech RAngarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai.**
- Progressing research work in Machine Learning domain.
- Published papers in Springer, Elsevier and in Scopus indexed journals.
- Interested in handling programming courses like C, C++, Java and Python
- Best practices in pedagogy approaches.
- Active Learning Strategies deployment in online sessions.

### Area of Specialization:

- Compiler Design
- Data Structures
- Data Mining
- Computer Networks

### Online Classes Handled in:

- Google Meet
- Microsoft Teams
- Zoom

### Personal Details:

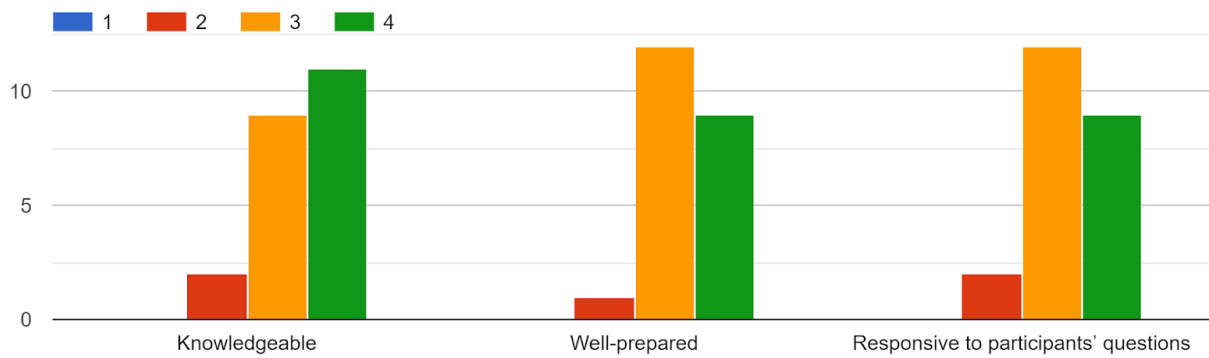
Father's Name : Mr.P.P.Meenatchi Sundaram  
Languages known : English(RWS), Tamil (RWS)& Telugu(S).  
Permanent Address : #2/101, Thiruparuthikundram Colony, Sevilimedu Post, Kanchipuram.

**KINGS ENGINEERING COLLEGE**  
**IRUNGATTUKOTTAI**

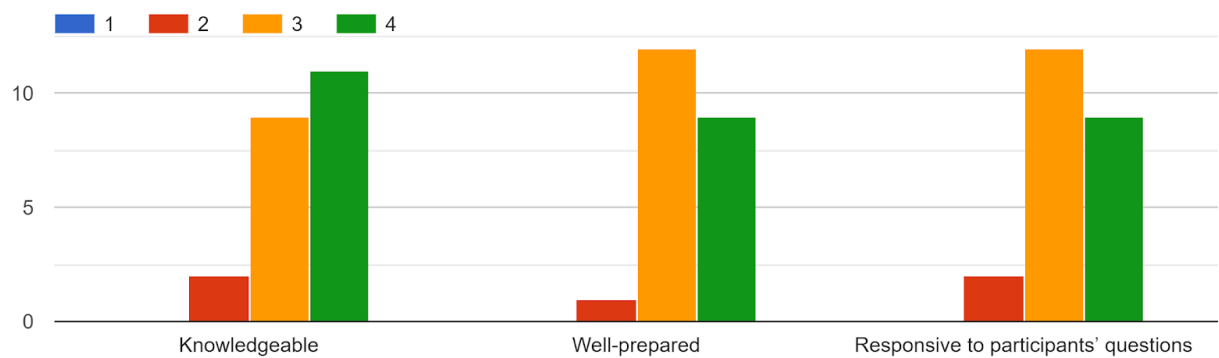
**STEAM - CODE CLUB**

**ONLINE TRAINING PROGRAMME – 14.10.2020**  
**PYTHON DATA STRUCTURES – FEEDBACK RESPONSES**

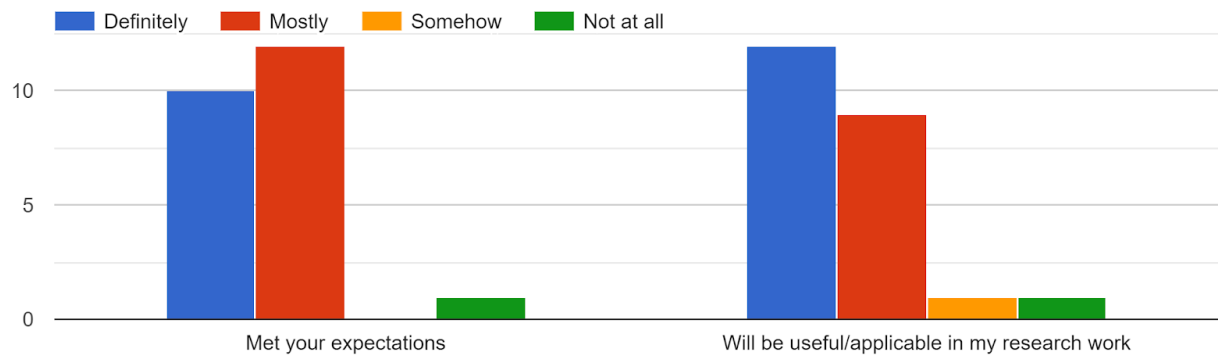
The facilitators were:



The facilitators were:



Knowledge and information gained from participation at this event?



Comments and suggestions :19 responses

- Good
- No
- Useful session
- Nice
- Good ....so informative
- Very useful session
- Very nice session we learnt a lot. Hoping for more session
- Very Useful Webinar
- Webinar was really useful
- Good session
- Useful and informative session
- Interesting and useful for understanding coding
- Very nice session, we learnt many together



**Sample Certificates:**



**KINGS**  
**ENGINEERING COLLEGE**  
Chennai-Bangalore Highway, Irungattukottai, Sriperumbudur, Chennai - 602 117.  
Ph: 044-71224401 - 08, Fax: 044-71224410  
ACCREDITED WITH NAAC & NBA, AFFILIATED TO ANNA UNIVERSITY and APPROVED BY AICTE

---

**CERTIFICATE OF PARTICIPATION**

This is presented to

**DAVID NIXON**

Of KINGS Engineering College has participated in the Webinar on  
"Python for Data Structures" on 14.10.2020.

  
Coordinator  
Mrs. Gracia Nissi S., AP/CSE

  
Principal  
Dr. T. John Oral Bhaskar



**KINGS**  
**ENGINEERING COLLEGE**  
Chennai-Bangalore Highway, Irungattukottai, Sriperumbudur, Chennai - 602 117.  
Ph: 044-71224401 - 08, Fax: 044-71224410  
ACCREDITED WITH NAAC & NBA, AFFILIATED TO ANNA UNIVERSITY and APPROVED BY AICTE

---

**CERTIFICATE OF PARTICIPATION**

This is presented to

**JOSHINI MARTINA A**

Of KINGS Engineering College has participated in the Webinar on  
"Python for Data Structures" on 14.10.2020.

  
Coordinator  
Mrs. Gracia Nissi S., AP/CSE

  
Principal  
Dr. T. John Oral Bhaskar

**Resource Person Certificate:**



**KINGS**  
ENGINEERING COLLEGE

Chennai-Bangalore Highway, Irungattukottai, Sriperumbudur, Chennai - 602117

**CERTIFICATE OF APPRECIATION**

This certificate is awarded to

**Mrs.M.Sumithra, AP/CSE**

of **VelTech Rangarajajn Dr.Sagunthala R&D Institute of Science and Technology, Chennai.**

for sharing her valuable knowledge as a Guest Speaker in the Online  
Training Programme on "Python Data Structures" Organized by

STEAM – Code Club held on **14.10.2020**

A handwritten signature in black ink, appearing to be 'S. Senthil', written over a white rectangular background.

**CONVENER**

A handwritten signature in blue ink, appearing to be 'J. Senthil', written over a white rectangular background.

**PRINCIPAL**

# Material

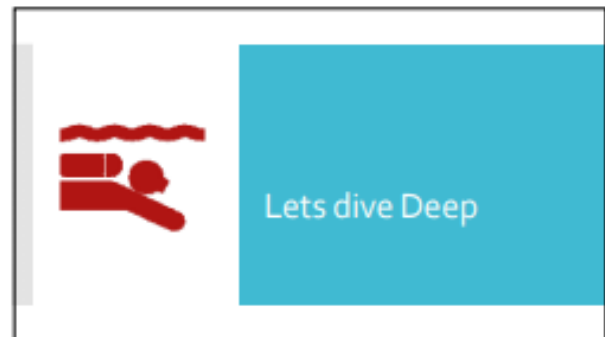
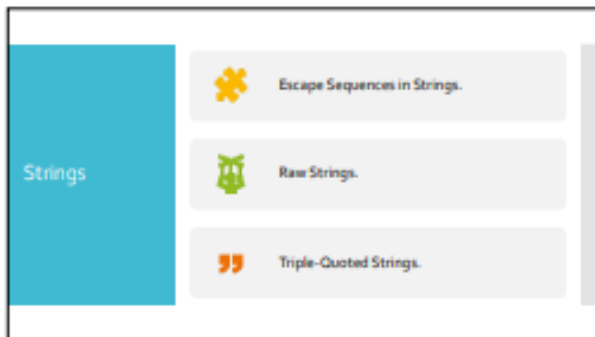
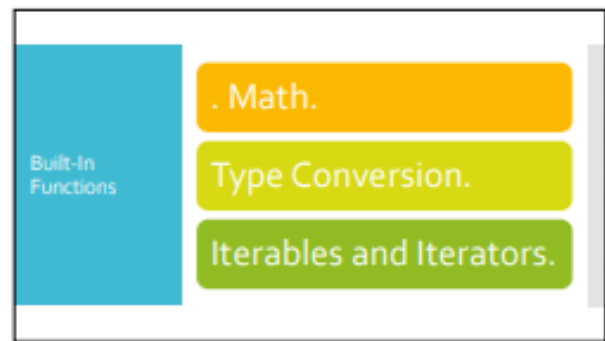
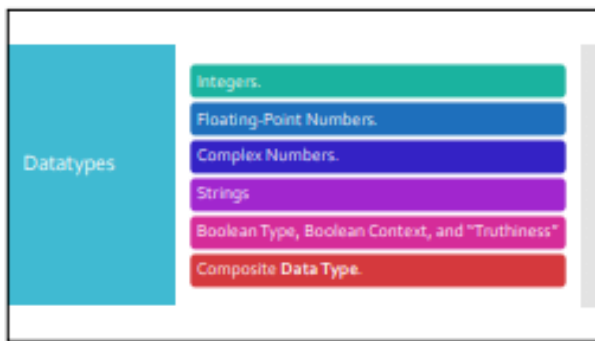
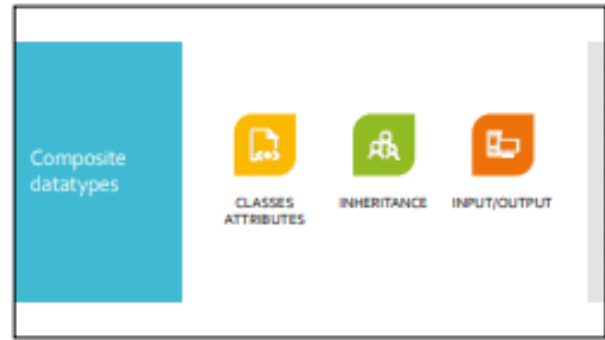


Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview

Basic Datatypes

Category	Python
Numeric	int long complex
Numeric with decimal part	float
Alphanumeric	String
Boolean	boolean





## Complex Numbers

• Complex numbers are specified as **`<real part>+<imaginary part>j`**.

```
>>> 2+3j
(2+3j)
>>> type(2+3j)
<class 'complex'>
```

Solution here:



```
>>> print('single quote (') character.')
- This string contains a single quote (') character.
-
>>> print('double quote (") character.')
- This string contains a double quote (") character.
```

- Strings are sequences of character data.
  - The string type in Python is called **`str`**.
  - Delimiters: **either single or double quotes**.
  - All the characters between the opening delimiter and matching closing delimiter are part of the string.
- ```
>>> print("I am a string.")
I am a string.
>>> type("I am a string.")
<class 'str'>
-
>>> print('I am too.')
I am too.
>>> type('I am too.')
<class 'str'>
```



Escape Sequences in Strings

```
>>> print("This string contains a single quote (\') character.")
```

This string contains a single quote (') character.



How to use special Characters?

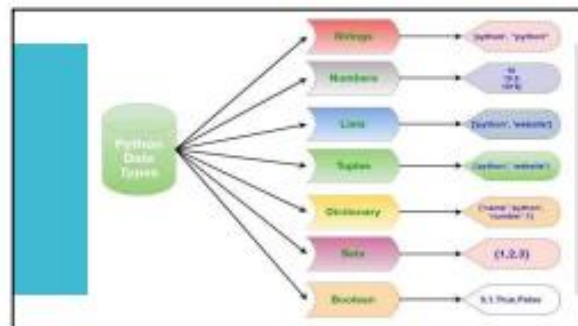
```
>>> print('single quote (') character.')
SyntaxError: invalid syntax
```

Python

```
>>> print("This string contains a double quote (\") character.")
```

This string contains a double quote (") character.

| Function | Description                                                         |
|----------|---------------------------------------------------------------------|
| abs()    | Returns absolute value of a number                                  |
| divmod() | Returns quotient and remainder of integer division                  |
| max()    | Returns the largest of the given arguments or items in an iterable  |
| min()    | Returns the smallest of the given arguments or items in an iterable |
| pow()    | Raises a number to a power                                          |
| round()  | Rounds a floating-point value                                       |
| sum()    | Sums the items of an iterable                                       |



| Function     | Description                                                         |
|--------------|---------------------------------------------------------------------|
| repr()       | Returns a string containing a printable representation of an object |
| str()        | Converts an object to a string                                      |
| int()        | Converts an object to a Python int                                  |
| float()      | Returns string representation of integer given by integer argument  |
| complex()    | Returns a complex number constructed from arguments                 |
| ord()        | Returns floating point object constructed from a number or string   |
| chr()        | Converts an integer to a character string                           |
| hex()        | Returns an integer value constructed from a number or string        |
| oct()        | Converts an integer to an octal string                              |
| id()         | Returns integer representation of an object                         |
| isinstance() | Returns a string containing a printable representation of an object |
| isint()      | Returns string version of an object                                 |
| isfloat()    | Returns the type of an object or creates a new type object          |



| Function    | Description                                                                |
|-------------|----------------------------------------------------------------------------|
| sys.argv[0] | Creates and returns an object of the sys.argv class                        |
| sys.argv[1] | Creates and returns a sys.argv object (similar to sys.argv, but immutable) |
| argv[0]     | Creates a sys.argv object                                                  |
| sys.argv[1] | Creates a sys.argv object                                                  |
| sys.argv[2] | Creates a sys.argv object                                                  |
| sys.argv[3] | Creates a sys.argv object                                                  |
| sys.argv[4] | Creates a sys.argv object                                                  |

### Input / Output

| input() | Reads input from the console           |
|---------|----------------------------------------|
| open()  | Opens a file and returns a file object |
| print() | Prints to a text stream or the console |

The input() function allows user input.  
**Syntax**  
input(prompt)

The print() function print the element in screen  
**Syntax**  
print(value)

| Operator | Description                                                                                                       | Example                                         |
|----------|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| ==       | If the values of two operands are equal, then the condition becomes true.                                         | (x == y) is not true.                           |
| !=       | If values of two operands are not equal, then condition becomes true.                                             | (x != y) is true.                               |
| >        | If values of two operands are not equal, then condition becomes true.                                             | (x > y) is true. This is similar to > operator. |
| <        | If the value of left operand is greater than the value of right operand, then condition becomes true.             | (x < y) is not true.                            |
| <=       | If the value of left operand is less than or equal to the value of right operand, then condition becomes true.    | (x <= y) is true.                               |
| >=       | If the value of left operand is greater than or equal to the value of right operand, then condition becomes true. | (x >= y) is not true.                           |
| <=       | If the value of left operand is less than or equal to the value of right operand, then condition becomes true.    | (x <= y) is true.                               |

**RELATIONAL OPERATOR**

| Operator | Description                                                                                                         | Example                                                                            |
|----------|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| &        | Binary AND<br>Operator copies a bit to the result if it exists in both operands.                                    | (x & y) (means 0001 1100)                                                          |
|          | Binary OR<br>It copies a bit if it exists in either operand.                                                        | (x   y) = 62 (means 0011 1101)                                                     |
| ^        | Binary XOR<br>It copies the bit if it is set in one operand but not both.                                           | (x ^ y) = 49 (means 0011 1001)                                                     |
| ~        | Binary Ones Complement<br>It is unary and has the effect of 'flipping' bits.                                        | ~(x) = -62 (means 1111 1101 in 2's complement form due to a signed binary number). |
| <<       | Binary Left Shift<br>The left operand's value is moved left by the number of bits specified by the right operand.   | x << 2 = 144 (means 1100 1100)                                                     |
| >>       | Binary Right Shift<br>The left operand's value is moved right by the number of bits specified by the right operand. | x >> 2 = 45 (means 0011 1101)                                                      |

**Bitwise Operators**

| Operator | Description                                                                                 | Example                                |
|----------|---------------------------------------------------------------------------------------------|----------------------------------------|
| =        | Assigns values from right side operands to left side operand.                               | x = y = 6 assigns value of 6 to both x |
| +=       | Adds right operand to the left operand and assigns the result to left operand.              | x += 6 is equivalent to x = x + 6      |
| -=       | Subtracts right operand from the left operand and assigns the result to left operand.       | x -= 6 is equivalent to x = x - 6      |
| *=       | Multiples right operand with the left operand and assigns the result to left operand.       | x *= 6 is equivalent to x = x * 6      |
| /=       | Divides left operand with the right operand and assigns the result to left operand.         | x /= 6 is equivalent to x = x / 6      |
| %=       | Takes modulus using two operands and assigns the result to left operand.                    | x %= 6 is equivalent to x = x % 6      |
| **=      | Performs exponential (power) calculation on operands and assigns value to the left operand. | x **= 6 is equivalent to x = x ** 6    |
| //       | Performs floor division on operands and assigns the result to left operand.                 | x // 6 is equivalent to x = x // 6     |

**Python Assignment Operators**

| Logical Operators  |                                                                      |                        |
|--------------------|----------------------------------------------------------------------|------------------------|
| Operator           | Description                                                          | Example                |
| and<br>Logical AND | If both the operands are true then condition becomes true.           | (x and y) is true.     |
| or<br>Logical OR   | If any of the two operands are non-zero then condition becomes true. | (x or y) is true.      |
| not<br>Logical NOT | Used to reverse the logical state of its operand.                    | Not(x and y) is false. |

Bitwise Operators

|                      |                      |
|----------------------|----------------------|
| a = 0011<br>1100     | b = 0000<br>1101     |
| a & b =<br>0000 1100 | a   b = 0011<br>1101 |
| a ^ b =<br>0011 0001 | ~a = 1100<br>0011    |

| Operator | Description                                                                                     | Example                                                                  |
|----------|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| in       | Evaluates to true if it finds a variable in the specified sequence and false otherwise.         | x in y, here in results in 1 if x is a member of sequence y.             |
| not in   | Evaluates to true if it does not find a variable in the specified sequence and false otherwise. | x not in y, here not in results in 1 if x is not a member of sequence y. |

Python Membership Operators

If statement

**Syntax**  
if ( condition):  
    #statement



The for loop in Python is used to iterate over a sequence ([list](#), [tuple](#), [string](#)) or other iterable objects.

For loop



Iterating over a sequence is called traversal.

If - else statement

**Syntax**  
if ( condition):  
    #statement<sub>1</sub>  
else:  
    #statement<sub>2</sub>

Range function

We can generate a sequence of numbers using range() function. range() will generate numbers from 0 to 9 (as numbers).

We can also define the start, stop and step size as arguments. (step, stop, start). step\_size defaults to 1 if not provided.

The range object is "lazy" in a sense because it doesn't generate every number that it "contains" when we create it. However, it is not an iterable since it supports only len and \_\_getitem\_\_ operations.

Nested - if statement

**Syntax**  
if ( condition):  
    #statement<sub>1</sub>  
    elif:  
        #statement<sub>2</sub>  
    else:  
        #statement<sub>3</sub>



We can generate a sequence of numbers using range() function. range() will generate numbers from 0 to 9 (as numbers).



We can also define the start, stop and step size as arguments. (step, stop, start). step\_size defaults to 1 if not provided.



The range object is "lazy" in a sense because it doesn't generate every number that it "contains" when we create it. However, it is not an iterable since it supports only len and \_\_getitem\_\_ operations.

Examples





### Pass Statement

- function definitions cannot be empty, but if you for some reason have a function definition with no content, put in the pass statement to avoid getting an error.

