**DAY 3 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **TCS-ION** | **USN:** | **4AL16EC008** |
| **Topic:** | **1. Ace Corporate Interviews**  **2. Learn Corporate Etiquette**  **3. Write Effective Emails** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Introduction to Fourier Series & Fourier Transform**  Fourier series expansion or harmonic analysis extracts appropriately weighted harmonic components from a general periodic waveform. Any function f(x), which is periodic between −𝜋 and + (or L to + L) can be expanded in this interval by a Fourier series. The Fourier series expansion of the function f(x) is defined by    The Fourier transform is a generalization of the [complex](https://mathworld.wolfram.com/ComplexNumber.html) [Fourier series](https://mathworld.wolfram.com/FourierSeries.html) in the limit as L->infty. Replace the discrete A_n with the continuous F(k)dk while letting n/L->k. Then change the sum to an [integral](https://mathworld.wolfram.com/Integral.html), and the equations become    **Inner product in Hilbert transform**  An n-D inner product vector space, called a Euclidean space, is a set of all n-D vectors with inner product defined. This space can be spanned by a set of n linearly independent basis vectors $\{ {\bf b}_1,\cdots,{\bf b}_n \}$ (none of them can be represented as a linear combination of the rest), so that any vector ${\bf x}$ in the space can be expressed as a linear combination of these basis vectors:  \begin{displaymath}{\bf x}=\sum_{k=1}^n c_k {\bf b}_k=[{\bf b}_1,\cdots,{\bf b}_n] \left[\begin{array}{c}c_1 \vdots  c_n\end{array} \right] \end{displaymath}  This can also be expressed in element form:  \begin{displaymath}{\bf x}=\left[\begin{array}{c}x_1 \vdots  x_n\end{array} ... ...{array}{c}c_1 \vdots  c_n\end{array}\right]={\bf B}{\bf c} \end{displaymath}    where ${\bf B}=[{\bf b}_1,\cdots,{\bf b}_n]$ is an n by n matrix with the n basis vectors as its columns, and ${\bf c}=[c_1,\cdots,c_n]^T$ is a column vector composed of n coefficients or weights for the the basis vector. These coefficients can be obtained by solving this linear system:  \begin{displaymath}{\bf c}={\bf B}^{-1}{\bf x} \end{displaymath}  In particular, if the basis vectors are orthonormal:  \begin{displaymath}<{\bf b}_i,{\bf b}_j>={\bf b}_i^T {\bf b}^*_j=\delta[i-j] \end{displaymath}  then ${\bf B}^{*T}={\bf B}^{-1}$ is a unitary matrix (or orthogonal matrix if ${\bf B}^*={\bf B}$), and the equation above becomes:  \begin{displaymath}{\bf c}=\left[\begin{array}{c}c_1 \vdots  c_n\end{array}\... ...}^{*T}_1\\ \vdots  {\bf b}^{*T}_n\end{array}\right]{\bf x} \end{displaymath} |

**DAY 3 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **PYTHON** | **USN:** | **4AL16EC008** |
| **Topic:** | **Python** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**      **Loops in Python**  A for loop is used for iterating over a sequence that is either a list, a tuple, a dictionary, a set, or a string. This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages. With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.  **Program to loop over colors.**  colors = [11, 34, 98, 43, 45, 54, 54]  for x in colors:  print(x)  **Output**:  11  34  98  43  45  54  54  **Program to loop over big colors.**  colors = [11, 34, 98, 43, 45, 54, 54]  for color in colors:  if color > 50:  print(color)  **Output**:  98  54  54  **Functions with multiple arguments**  A Routine is a named group of instructions performing some tasks. A routine can always be invoked as well as called multiple times as required in a given program. When the routine stops, the execution immediately returns to the stage from which the routine was called. Such routines may be predefined in the programming language or deigned or implemented by the programmer. A Function is the Python version of the routine in a program. Some functions are designed to return values, while others are designed for other purposes. We pass arguments in a function, we can pass no arguments at all, single arguments or multiple arguments to a function and can call the function multiple times.  **Program to find average function.**  def Average(lst):      return sum(lst) / len(lst)  lst = [15, 9, 55, 41, 35, 20, 62, 49]  average = Average(lst)  print("Average of the list =", round(average, 2))  **Output**:  35.75  **File Processing**  One of the most common tasks that you can do with Python is reading and writing files. Whether it’s writing to a simple text file, reading a complicated server log, or even analyzing raw byte data, all of these situations require reading or writing a file. At its core, a file is a contiguous set of bytes used to store data. This data is organized in a specific format and can be anything as simple as a text file or as complicated as a program executable. In the end, these byte files are then translated into binary 1 and 0 for easier processing by the computer.  When you want to work with a file, the first thing to do is to open it. This is done by invoking the open() built-in function. open() has a single required argument that is the path to the file. open() has a single return, the file object:  file = open('dog\_breeds.txt') Reading and Writing Opened Files Once you’ve opened up a file, you’ll want to read or write to the file. Program to read file is shown below.  file = open("bear.txt")  print(file.read())    Below code opens the the file bear.txt and then read the contents of the file and prints out the content.  **A python program to create a file with name file.txtwrite the text snail there.**  with open("file.txt", "w") as file:  file.write("snail") |