**DAY 7 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **25-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **DSP** | **USN:** | **4AL16EC008** |
| **Topic:** | **DSP** | **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 7 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **25-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **PYTHON** | **USN:** | **4AL16EC008** |
| **Topic:** | **Python** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**      **Build a Personal Website with Python and Flask**  Why not making a website before you even learn how to make a website? Just something as simple as this:  basic flask website in python  Yes, it’s just a page with some plain text, but it is still called a website and that’s how it starts. Python is concise so it takes only seven lines of code to build what you see in the screenshot. Let’s do it.  Insert the following lines of Python code inside an empty text file, and name the  file something like hello.py.  from flask import Flask  app = Flask(\_\_name\_\_)  @app.route('/')  def home():  return "Hey there!"  if \_\_name\_\_ == '\_\_main\_\_':  app.run(debug=True)  As you can see we are importing the flask library in the first line. If you don’t have that library installed, you will get an error. To install flask, simply type in pip install flask in your computer terminal/command line. Once you have made sure *flask* is installed, simply run the *hello.py* script.  **HTML templates in flask**  flask html templates  As you see, we don’t have plain text anymore, but text with various formats. That is made possible by returning an HTML template instead of a plain Python string.  Create an empty file, name it something like home.html and put the following HTML code inside it:  <!DOCTYPE html>  <html>  <body>  <h1>My Personal Website</h1>  <p>Hi, this is my personal website.</p>  </body>  </html>  The flask framework has been written in a way so that it looks for HTML template files in a folder that should be named templates. So, you should create such an empty folder and then put all the HTML templates in there. Here is how the web app directory tree should like at this point:  flask directory tree templates folder  So, the Python script stays outside of the templates folder.  Let’s now edit the Python code so that it reads the HTML template and returns it to the webpage. Here is the updated version:    **from** flask **import** Flask, render\_template  app = Flask(\_\_name\_\_)  @app.route('/')  **def** home():  **return** render\_template('home.html')  **if** \_\_name\_\_ == '\_\_main\_\_':  app.run(debug=True) |