

# USE OF PYTHON IN MEDICAL IMAGING(PACS)

FEMI OLADELE

# What is Medical Imaging?

- The technique and process of creating **visual representations** of the interior of a body for clinical analysis and medical intervention
- Medical Imaging is used for both diagnosis and therapeutic purposes as such it is a powerful tool available to effectively care for patients
- Common imaging types for diagnosis include: CT(Computer Tomography), MRI(Magnetic Resonance Imaging), Ultrasound, X-ray.

# Magnetic Resonance Imaging Machine



# Evolution of Medical Imaging

- Began in November 1895 with Wilhelm Conrad Roentgen's discovery of X-ray
- Computers really made an entrance into the world of Medical Imaging in early 1970s with the advent of CT and the MRI
- 1995 DICOM (Digital Imaging and Communications in Medicine) Standard was formalized) to solve the problem of compatibility of images generated by various machine types.
- First Generation PACS in early '90s was designed to deliver relatively large files over radiology departments network

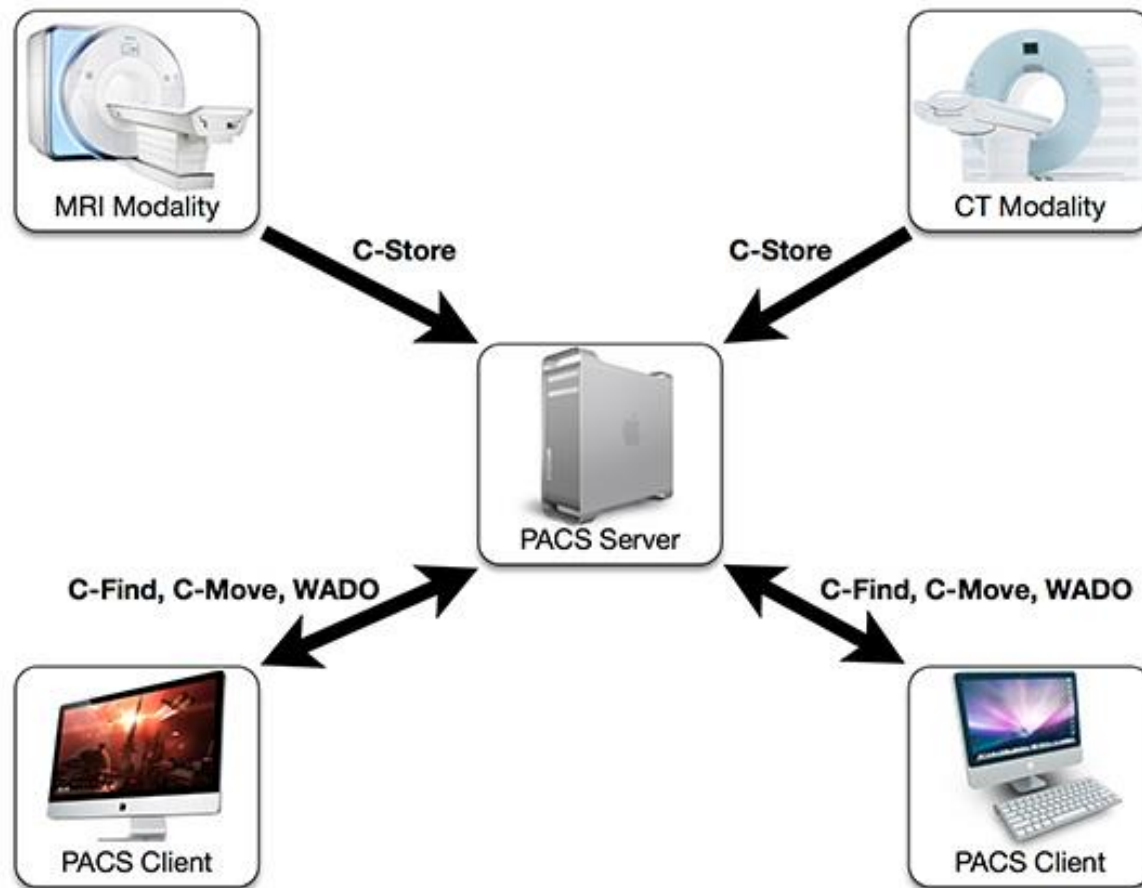
# Digital Imaging and Communications in Medicine(DICOM)

- This is a standard for handling, storing, printing and transmitting information in medical imaging.
- It includes a file format definition and a network communications protocol.
- Files are saved in .dcm extensions

# Picture Archiving and Communications Systems(PACS)

- PACS is a medical imaging technology that can do the following:
- Provide economical storage of and convenient access to, patient study images from multiple imaging modalities
- Transmit digital images and electronic reports via network
- Enhances Radiology productivity and workflow
- Improves patient care

# PACS



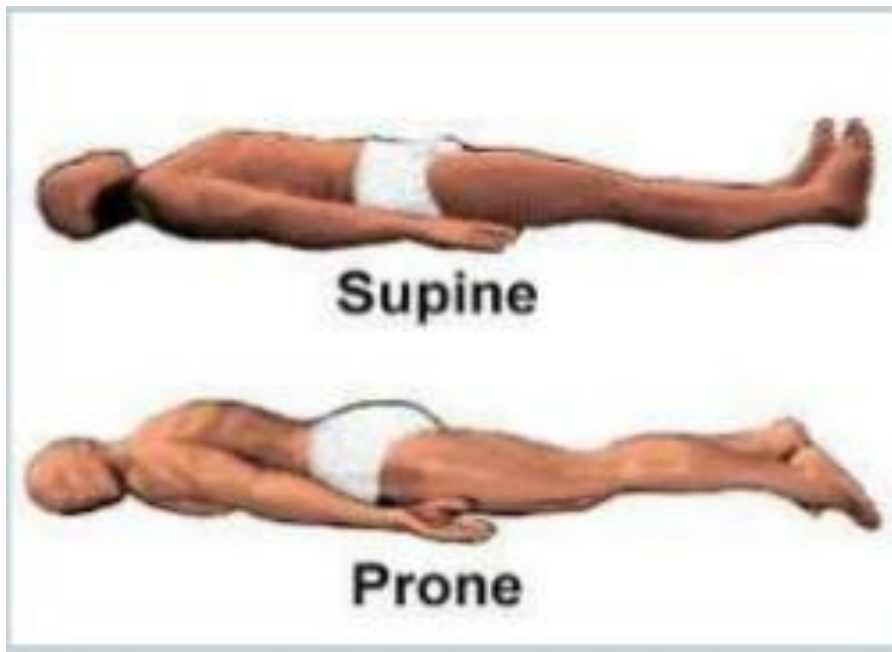
# PYDICOM

- A pure Python package for working with DICOM files such as medical images, reports and radiotherapy objects.
- Makes it easy to read complex files into pythonic structures for easy manipulations
- Modified datasets can be written again to DICOM file formats
- Is not a DICOM server and is not primarily about viewing images, it shines most in the areas of manipulating data elements in DICOM files using python codes



# Simple usage of PYDICOM

- To read a radiotherapy plan file, change the patient setup from **head-first-supine** to **head-first-prone** and save to a new file.



# Supine to Prone with PYDICOM

## Code Example 1

- <https://gitlab.com/profemzy/pyconng-medical-imaging.git> (supine-prone.ipynb)

# Viewing Images

- As mentioned earlier Pydicom is more focused on accessing the DICOM data elements in files but it is also desirable to view pixel data as an image
- The following options are available:
- Use any of the many [DICOM viewer](#) programs available
- Use pydicom with [matplotlib](#)
- Use pydicom with Python's stdlib Tkinter module
- use pydicom with the [Python Imaging Library \(PIL\)](#)
- use pydicom with [wxPython](#)

# Using PYDICOM with matplotlib

## Code Example 2

- <https://gitlab.com/profemzy/pyconng-medical-imaging.git> (pymatplot.ipynb)

# A lot more to do with PYDICOM

- [https://pydicom.github.io/pydicom/stable/auto\\_examples/index.html](https://pydicom.github.io/pydicom/stable/auto_examples/index.html)

# So much to be done

- Reviews and Reporting about medical images such as CT Scans, MRI Scans and other such images can be a frustrating task when it is limited to views from films under bright light, many hospitals and diagnostic centers have moved to Picture and Communication Archiving Systems(PACS) this is still limited because Radiology consultants have to physically go to the centers to report because most of these systems are intranet based. Python Libraries such as PyDicom helps us solve this problem and build applications that are accessible regardless of locations.

# A good project to start with

- A web simple application that will allow people upload their scans(dcm) format to the app, and allows experts to view, examine and generate medical reports on such images and have the reports sent back to the uploader.

# References

- <https://www.dicomlibrary.com/dicom/>
- [https://www.bridgeheadsoftware.com/2013/11/the evolution of pacs picture archiving and communication system/](https://www.bridgeheadsoftware.com/2013/11/the-evolution-of-pacs-picture-archiving-and-communication-system/)
- [https://pydicom.github.io/pydicom/stable/getting started.html](https://pydicom.github.io/pydicom/stable/getting-started.html)
- <https://www.osirix-viewer.com/resources/pacs/>