2026 SPARK ACADEMY

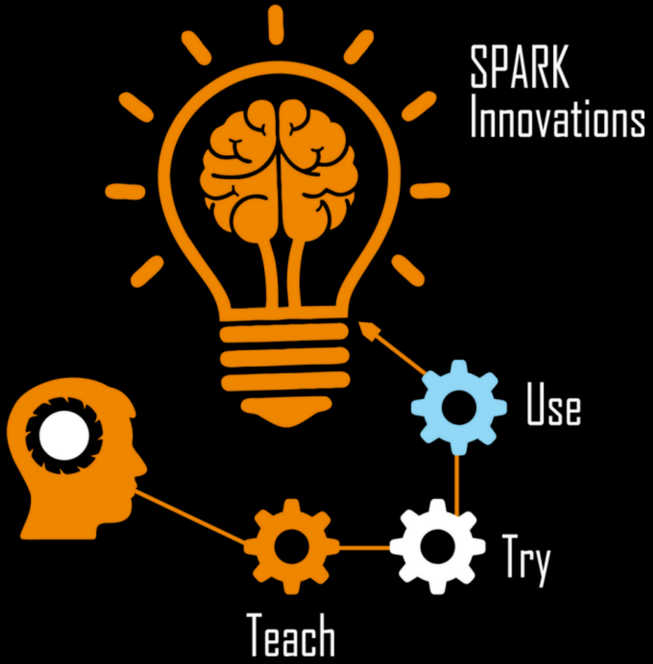
TRAIN FOR CHANGE, FROM SCIENCE TO PRACTICE

THE SPRINT AI TRAINING FOR AFRICAN MEDICAL
Imaging Knowledge Translation (SPARK) Academy

IN DEEP LEARNING & MEDICAL IMAGING

Week 1 Tutorial (Introduction to python)

FEBRAURY 21 2026



"I don't have programming knowledge..."

"I don't know DICOM, NIfTI, medical imaging..."

"These are big words and confusing..."

YES, you can still do SPARK!

We've got you covered.

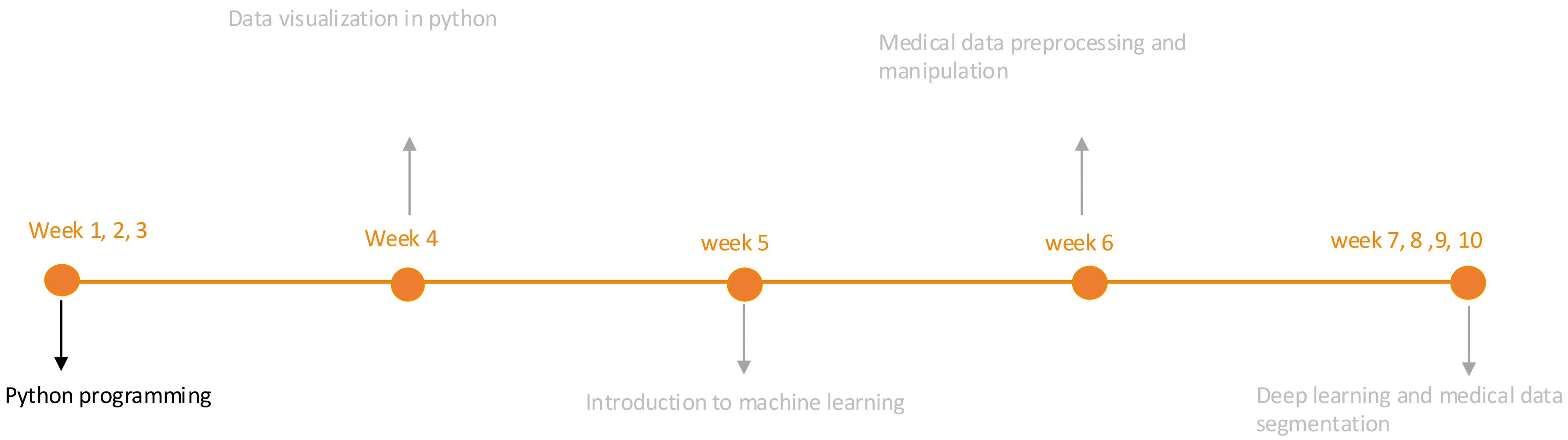
Welcome to SPARK 2026

Exited to start ??!!

https://event.fourwaves.com/spark

Week 1
Introduction to Python

What you are going to learn in SPARK


Data visualization in python
Medical data preprocessing and manipulation
Week 1, 2, 3
Week 4
week 5
week 6
week 7, 8 ,9, 10
Python programming
Introduction to machine learning
Deep learning and medical data segmentation

https://event.fourwaves.com/spark

What you are going to learn in SPARK

Week 1, 2, 3

Python programming

Week 4

Data visualization in python

week 5

Introduction to machine learning

week 6

Medical data preprocessing and manipulation

week 7, 8 ,9, 10

Deep learning and medical data segmentation

https://event.fourwaves.com/spark

What you are going to learn in SPARK

Week 1, 2, 3

Python programming

Week 4

Data visualization in python

week 5

Introduction to machine learning

week 6

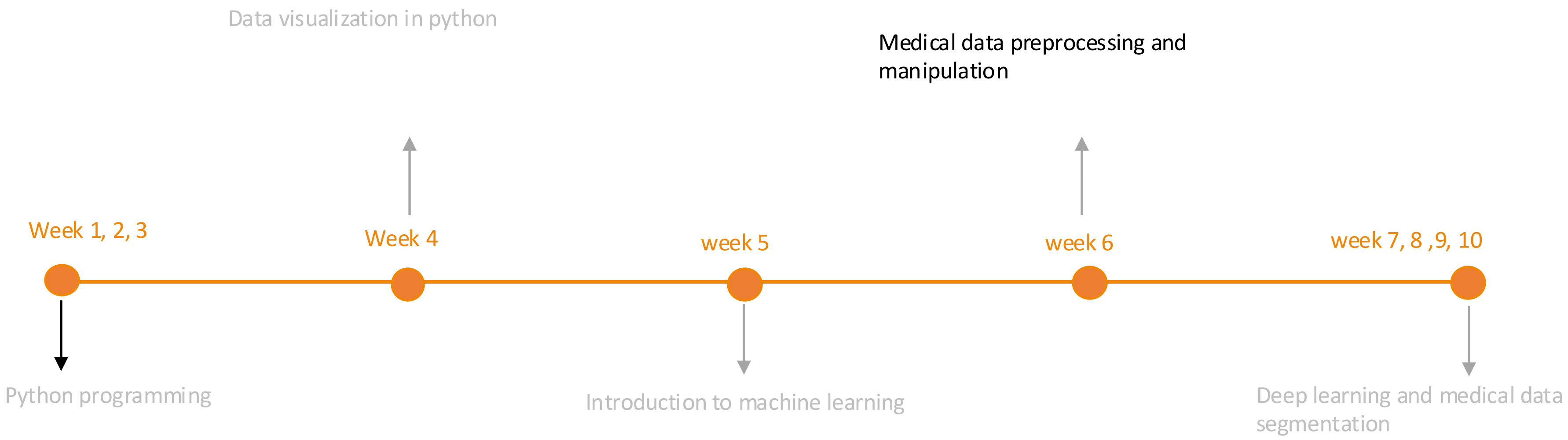
Medical data preprocessing and manipulation

week 7, 8 ,9, 10

Deep learning and medical data segmentation

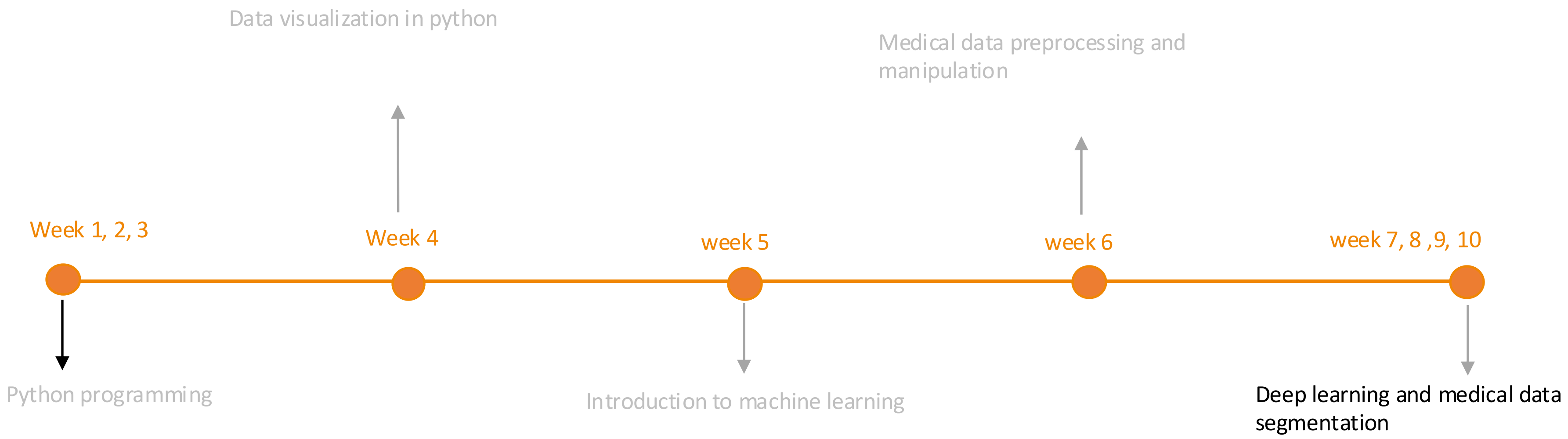
https://event.fourwaves.com/spark

What you are going to learn in SPARK


Data visualization in python
Medical data preprocessing and manipulation
Week 1, 2, 3
Week 4
week 5
week 6
week 7, 8 ,9, 10
Python programming
Introduction to machine learning
Deep learning and medical data segmentation

https://event.fourwaves.com/spark

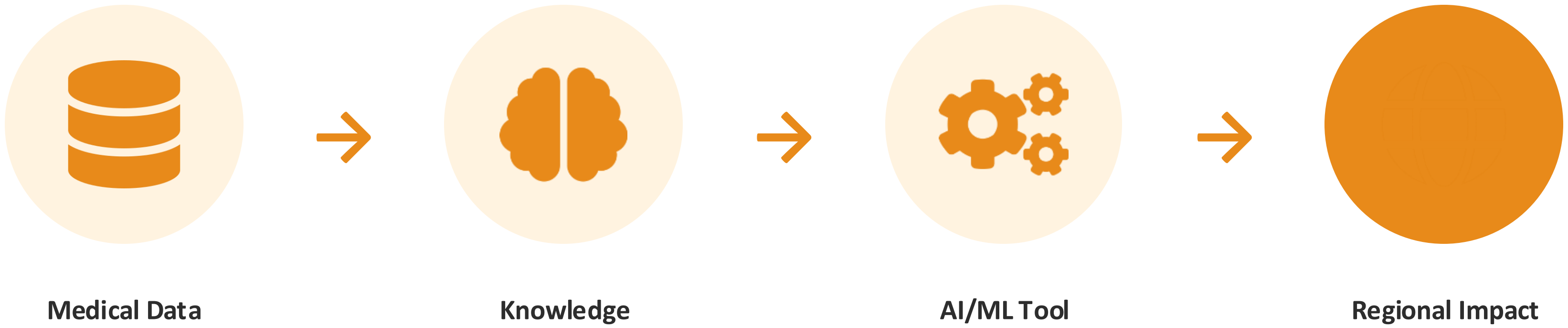
What you are going to learn in SPARK


Data visualization in python
Medical data preprocessing and manipulation
Week 1, 2, 3
Week 4
week 5
week 6
week 7, 8 ,9, 10
Python programming
Introduction to machine learning
Deep learning and medical data segmentation

https://event.fourwaves.com/spark

The End Goal

Take the knowledge you gain to build real solutions within your region



What is Programming?

How do we tell computers what to do?

The Communication Gap

Human

Speaks English,
French, Yoruba...

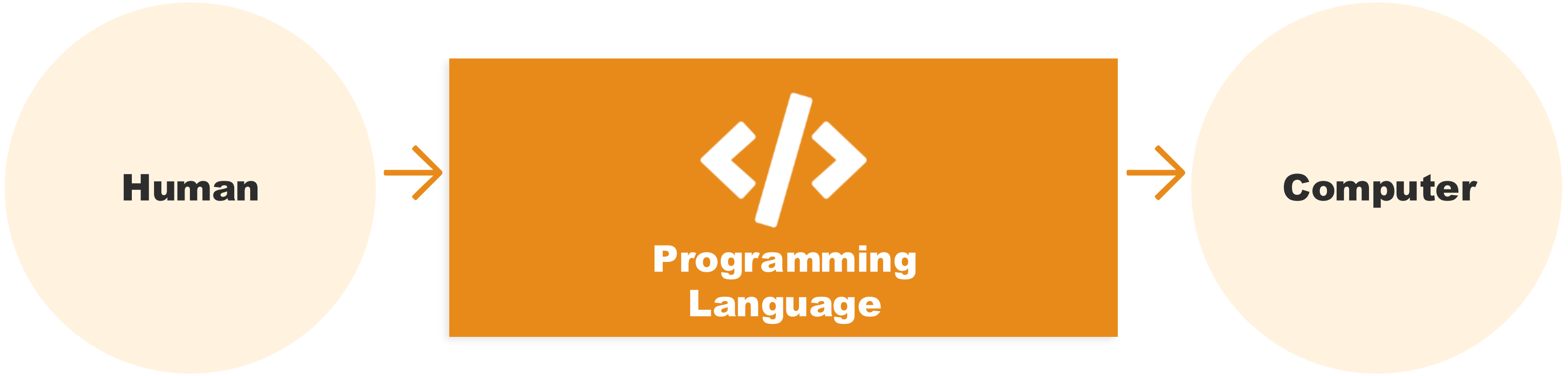
VS

Machine

Speaks in
0s and 1s

The Bridge: Programming Languages

We communicate with computers using a programming language



Programming Languages

There are many to choose from

Python	**JavaScript**	**C++**	**Java**
R	**MATLAB**	**Julia**	**Go**

Which One Should I Pick?

It depends on the task!

Python

For Medical Imaging & Machine Learning

Easy to learn • Huge community • Rich libraries

How Do I Learn Python?

Start with the building blocks

Syntax

The rules for writing code

How you structure your code
so the computer understands it

```
print("Tumor detected")
```

Semantics

The meaning behind the code

What the code actually does
when it runs

```
# Displays text on screen
```

Setting Up Your Environment

VS Code

Local editor
Full control

Google Colab

Browser-based
Free GPU access

Recommended

Kaggle

Datasets + Code
Competitions

Recommended

Let's Learn Python!

Time to write some code

Comments

Notes for humans — Python ignores them

```
# This is a comment - Python skips it
# Store patient scan information
scan_type = "MRI"   # T1-weighted brain scan

"""
This is a multi-line comment.
Useful for longer explanations
about your medical imaging pipeline.
"""
```

Why Comment?

Explain your code
Help others understand
Remind future you!

The print() Statement

Display output to the screen

```
print("Welcome to Radiology Department")
```

Output:
```
Welcome to Radiology Department
```

```
print("Patient ID:", 2041)
print("Scan Type:", "Brain MRI")
```

Output:
```
Patient ID: 2041
Scan Type: Brain MRI
```

Quick Tip

print() can display text, numbers, and variables

Strings

Text data enclosed in quotes

```python
diagnosis = "Glioblastoma"
organ = 'Brain'

print(diagnosis)
print(len(diagnosis))
print(diagnosis.upper())
print(organ.lower())
```

Output:
```
Glioblastoma
12
GLIOBLASTOMA
brain
```

Common Methods

```
.upper()

.lower()

.replace()

.split()

.strip()

len()
```

f-Strings (Formatted Strings)

Embed variables directly inside text

```
patient = "Chidi"
age = 45
scan = "CT Chest"

print(f"Patient {patient}, Age {age}")
print(f"Scheduled for: {scan}")
```

Output:
```
Patient Chidi, Age 45
Scheduled for: CT Chest
```

```
tumor_size = 3.456
print(f"Tumor size: {tumor_size:.1f} cm")
```

Output:
```
Tumor size: 3.5 cm
```

Key Syntax

f"text {variable}"

The f before the quotes is required!

Data Types

Python has different types of data

int	float	str	bool
45	3.14	"MRI"	True
Patient age	Tumor size (cm)	Scan type	Tumor present?

```
# type() tells you what type a value is
age = 45
print(type(age))       # <class 'int'>

tumor_size = 2.3
print(type(tumor_size)) # <class 'float'>
```

Output:
```
<class 'int'>
<class 'float'>
```

Variables

Store data for later use — like patient records

A hospital records: Patient Name, Age, Diagnosis, Scan Type, Tumor Size

```python
patient_name = "Fatima"
age = 52
diagnosis = "Meningioma"
scan_type = "Brain MRI"
tumor_size = 2.8

print(f"Patient: {patient_name}")
print(f"Age: {age}")
print(f"Tumor: {tumor_size} cm")
```

Output:
```
Patient: Fatima
Age: 52
Tumor: 2.8 cm
```

Variable Rules

✓ Letters, numbers, _

✓ Case sensitive

✗ Can't start with number

✗ No spaces in names

Getting User Input

Ask the user for information

```
# input() always returns a string
patient = input("Enter patient name: ")
age = int(input("Enter patient age: "))

print(f"Registering {patient}, age {age}")
```

Output:

```
Enter patient name: Amina
Enter patient age: 34
Registering Amina, age 34
```

Remember

input() returns text

Use int() to convert to a number

Use float() for decimals

Data Structures

Organizing medical data in Python

Lists

Store multiple items — like a list of patient scans

```
scans = ["MRI", "CT", "X-Ray", "Ultrasound"]
tumor_sizes = [2.3, 1.8, 4.1, 0.9]

print(scans)
print(len(scans))
```

Output:
```
['MRI', 'CT', 'X-Ray', 'Ultrasound']
4
```

```
# Indexing & Slicing
print(scans[0])         # MRI
print(scans[-1])        # Ultrasound
print(scans[1:3])       # ['CT', 'X-Ray']
```

Lists can hold:

Mixed types

Duplicates

Are ordered

Are mutable

List Operations

```
# Managing a patient scan queue
queue = ["MRI - Fatima", "CT - Chidi", "X-Ray - Ama"]

queue.append("MRI - Kofi")      # Add patient
queue[1] = "CT - Bola"          # Update record
queue.remove("X-Ray - Ama")     # Remove patient

print(queue)
```

Output: `['MRI - Fatima', 'CT - Bola', 'MRI - Kofi']`

Useful Methods

`.append()`

`.remove()`

`.pop()`

`.sort()`

`.reverse()`

`.insert()`

Dictionaries

Key-value pairs — perfect for patient records

```
patient = {
    "name": "Fatima",
    "age": 52,
    "diagnosis": "Meningioma",
    "scan": "Brain MRI",
    "tumor_cm": 2.8
}

print(patient["name"])
print(patient["tumor_cm"])
```

Output:
```
Fatima
2.8
```

Key Features

Keys must be unique

Mutable (can change)

Access by key name

Ideal for records

Tuples & Sets

Tuple ( )

Ordered & Immutable (cannot change)

```
# MRI volume dimensions
dims = (256, 256, 128)
print(dims[0])  # 256
```

Output: 256

Set { }

Unordered & No duplicates

```
# Unique scan types today
scans = {"MRI", "CT", "MRI", "CT"}
print(scans)
```

Output: {'MRI', 'CT'}

Operators

Compare values — essential for medical decisions

Operator	Meaning	Medical Example
==	Equal to	scan == "MRI" → True
!=	Not equal	status != "Normal" → True
>	Greater than	tumor > 3.0 → True
<	Less than	age < 18 → False
>=	Greater or equal	dose >= 50 → True
<=	Less or equal	bmi <= 25 → True

```
tumor = 3.5
threshold = 3.0
print(tumor > threshold)  # True
```

Logical Operators

and	– both must be True
or	– at least one True
not	– reverses result

Conditional Statements

Make decisions in code — if this, then that

Should the patient be referred for surgery based on tumor size?

```
tumor_size = 3.5  # cm

if tumor_size > 3.0:
    print("Refer to surgery")
elif tumor_size > 1.0:
    print("Monitor closely")
else:
    print("No intervention needed")
```

Syntax:

```
if condition:
    do this
elif condition:
    do that
else:
    default
```

Output: `Refer to surgery`

For Loops

Repeat actions for each item in a collection

```
# Check each patient's tumor size
tumor_sizes = [1.2, 3.5, 0.8, 4.1]

for size in tumor_sizes:
    if size > 3.0:
        print(f"{size} cm - REFER")
    else:
        print(f"{size} cm - Monitor")
```

Output:
```
1.2 cm - Monitor
3.5 cm - REFER
0.8 cm - Monitor
4.1 cm - REFER
```

For Loop

Iterates through each item in a list, string, or range

While Loops & range()

```
# Process 5 scans using range()
for i in range(5):
    print(f"Processing scan {i + 1}...")
```

Output:
```
Processing scan 1...
Processing scan 2...
...Processing scan 5...
```

```
# While loop - repeat until condition is False
slices_remaining = 3
while slices_remaining > 0:
    print(f"Analyzing slice {slices_remaining}")
    slices_remaining -= 1
print("Scan analysis complete!")
```

Output:
```
Analyzing slice 3
Analyzing slice 2
Analyzing slice 1
Scan analysis complete!
```

Built-in Functions

Python comes with useful functions ready to use

```
# Tumor sizes from 4 patients (cm)
sizes = [2.3, 1.8, 4.1, 0.9]

print(sum(sizes))      # 9.1
print(len(sizes))      # 4
print(max(sizes))      # 4.1
print(min(sizes))      # 0.9
print(sorted(sizes))   # [0.9, 1.8, 2.3, 4.1]
print(round(2.567, 1)) # 2.6
```

`sum()`
Total of values

`min()`
Smallest value

`len()`
Count items

`sorted()`
Sort a list

`max()`
Largest value

`round()`
Round number

How Do I Become Good?

By Practicing!

Code every day • Break things • Fix them • Repeat

Let's Practice!

20 Questions

Thank You!

SPARK Academy 2026

Train for Change, From Science to Practice