



✓ Where Python Is Used in DSP

DSP Task	Python Role	Libraries/Tools
Signal generation	Create test signals (sine, square, noise)	<code>numpy</code> , <code>scipy.signal</code>
Visualization	Plot waveforms, spectra, spectrograms	<code>matplotlib</code> , <code>scipy</code> , <code>seaborn</code>
Filtering	Design and apply FIR/IIR filters	<code>scipy.signal</code> , <code>pyfilterbank</code>
Transform analysis	DFT, FFT, STFT, IFFT	<code>numpy.fft</code> , <code>scipy.fft</code>
Modulation/ Demodulation	Simulate AM, FM, QAM, etc.	Custom or <code>commpy</code>
IQ data handling	Parse, visualize, analyze IQ samples	<code>numpy</code> , <code>matplotlib</code> , custom scripts
SDR interfacing	Real-time DSP from SDR hardware	<code>pyrtlsdr</code> , <code>SoapySDR</code> , <code>gnuradio</code>
Machine learning	Signal classification, denoising, feature extraction	<code>scikit-learn</code> , <code>tensorflow</code> , <code>torch</code>

📁 Recommended Practice Tracks (in Python)

◆ 1. Basic Signal Processing in Python

- Sine wave generation, addition, scaling
- Noise addition and filtering
- Moving average filter

◆ 2. FFT / Frequency Analysis

- Compute FFT manually (step-by-step)

- Visualize magnitude spectrum
- Real-time FFT from IQ data

◆ 3. FIR / IIR Filters

- FIR design with windowing (e.g., Hamming)
- IIR filter design (Butterworth, Chebyshev)
- Filter application and response visualization

◆ 4. Working with Real IQ Data

- Read `.bin` or `.wav` IQ data
- Plot constellation diagrams
- Detect frequency shifts or signals

◆ 5. Projects & Experiments

- Bandpass filter for a specific signal
- Spectrogram display (short-time FFT)
- Signal detection from noisy IQ streams
- Simulate simple AM/FM transmitter and receiver

🔧 Tools & Libraries Setup

Make sure you have these installed:

```
pip install numpy scipy matplotlib
pip install soundfile # for audio signals
pip install pyrtlsdr  # for SDR work
```

🎯 Next Steps Based on Your Level

If you're in early/mid stages:

- Focus on writing simple signal generators and filters.
- Visualize time-domain and frequency-domain plots.

If you're moving to advanced DSP:

- Start working on SDR data streams.
- Build a full DSP pipeline: read IQ → process → visualize → demodulate.