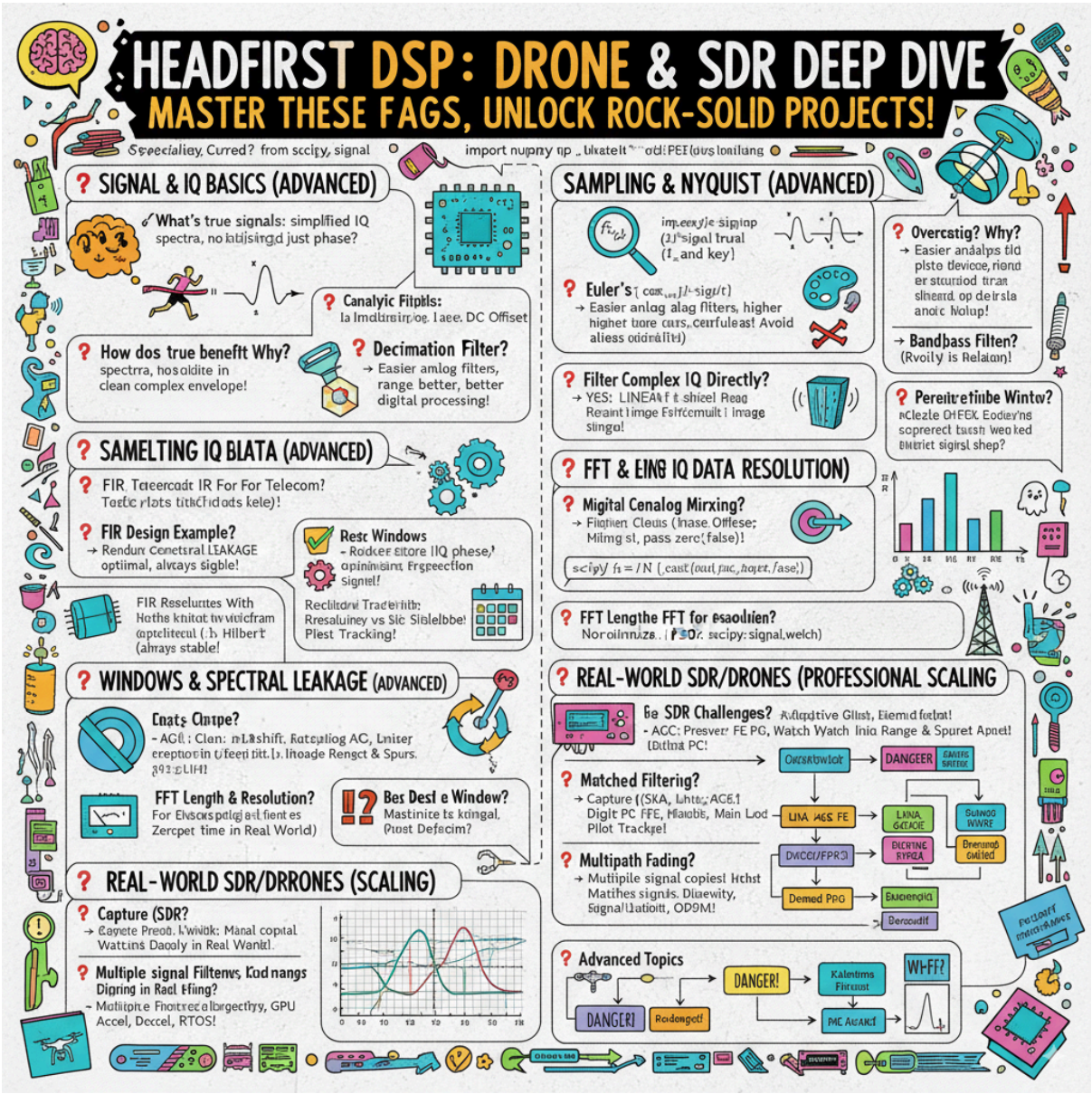
FREQUENTLY ASKED QUERIES



1. Signal & IQ Basics

- ❓ What is an IQ signal?
  → A complex signal with two parts: In-phase (I = cos) and Quadrature (Q = sin). Captures both amplitude & phase.
- ❓ Why not just use real signals?
  → Real signals lose phase info. IQ keeps complete information for modulation/demodulation.
- ❓ How do we represent IQ in Python?

```
t = np.arange(0, duration, 1/fs)
iq = np.exp(1j*2*np.pi*f_signal*t)
```

---

2. Sampling & Nyquist

- ❓ What is Nyquist frequency?
  → fs/2. Maximum frequency you can represent without aliasing.
- ❓ Why is aliasing dangerous?
  → Higher frequencies “fold” into lower ones, corrupting the spectrum.
- ❓ How do I avoid it?
  → Use an **anti-aliasing filter (LPF)** before downsampling.

---

3. Filtering (FIR)

- ❓ Why design filters?
  → To isolate the desired band or remove noise.
- ❓ How do I design a simple LPF?

```
from scipy.signal import firwin, freqz
taps = firwin(numtaps=101, cutoff=0.2)  # cutoff = 0.2 * Nyquist
```

- ❓ How do I check if my filter works?
  → Use `freqz(taps)` → plot magnitude response.

- ❓ What happens if my cutoff is wrong?
  → Signal distortion (too aggressive) or noise leaks (too loose).

---

4. Filtering IQ Data

- ❓ How to apply a filter?

```
from scipy.signal import lfilter
iq_filtered = lfilter(taps, 1.0, iq)
```

- ❓ Why do filtered signals look “smoothed”?
  → High-frequency components are removed.

---

5. FFT & Spectrum

- ❓ Why do FFT?
  → To see which frequencies are present in your signal.
- ❓ What is the link between time and frequency domain?
  → Time signal ↔ FFT ↔ Frequency spectrum.
- ❓ How to do FFT of IQ?

```
spectrum = np.fft.fftshift(np.fft.fft(iq))
```

---

6. Mixing / Frequency Shifting

- ❓ Why shift frequencies?
  → To bring a signal of interest down to baseband (center at 0 Hz).
- ❓ How to shift in Python?

```
f_shift = 1000
iq_shifted = iq * np.exp(-1j*2*np.pi*f_shift*t)
```

- ❓ Where do we use this in drones/SDR?

→ When a drone telemetry signal is not centered, we mix it to baseband for easier decoding.

---

7. Windows & Spectral Leakage

- ❓ Why do I see “spread” in FFT even for pure tones?
  → That’s **spectral leakage** from truncating signals.
- ❓ Fix?
  → Apply windows (Hann, Hamming, Blackman).
- ❓ Python example:

```
iq_win = iq * np.hanning(len(iq))
```

---

8. Real-World SDR/Drones

- ❓ What’s the typical signal chain?

```
Capture (SDR) → LPF → Downsample → Mix → FFT → Detect/Decode
```

- ❓ Why LPF first?
  → To remove out-of-band junk and prevent aliasing.
- ❓ Why FFT at the end?
  → To verify the signal sits at expected frequencies.
  Please scale it up for the professionals