1. **Convolution** is the basic operation:

$$y[n] = \sum h[k] \cdot x[n-k]$$

2. Different choices of $h[n]$ shape different outputs:
   - $h[n] = [1, 1, 1]/3$: averaging (smoothing)
   - $h[n] = [1, 0, -1]$: edge detection (high-pass)
3. **It's a local, weighted sum** — like a sliding window
4. **In frequency domain**, convolution = multiplication →
   "Sculpting the spectrum" — YES, **it becomes obvious** once you see it.

So you've covered the **core idea** of FIR filtering **already**.

🤔 Then What's Worth Adding to Remember for FIR Filters?

Here's where **real filter design and usage** go **beyond just convolution**:

🔷 1. Filter Design Strategy Matters

You didn't just "pick" $h[n]$ randomly — you need **rules** to generate:

- A **low-pass filter with cutoff 1 kHz**
- A **notch at 50 Hz**
- A **linear-phase filter with minimal ringing**

This is why you need:

- **Windowed sinc design**
- Kaiser, Hamming, Blackman windows
- Tools like `firwin`, `remez`

🔶 **What to remember**:

> FIR filters aren't just "any" $h[n]$. They're designed for specific spectral goals using formal methods.

🔷 2. Linear Phase FIR Is Special

A symmetric FIR filter:

- Has **linear phase**
- Preserves signal **shape and delay**
- Very important in comms, audio, image processing

🔍 **What to remember**:

> Symmetry in $h[n]$ → constant group delay → no waveform distortion.

🔷 3. Tradeoffs: Length vs Transition Sharpness

- More taps (longer $h[n]$) = sharper filter
- But this means:
  - More computation
  - More delay
  - Higher design complexity

🔍 **What to remember**:

> Filter sharpness comes at a cost — delay and compute. Optimize per use case.

🔷 4. FIR Filters Are Always Stable

- No feedback → no poles → always stable
- You can't "blow up" an FIR filter like you can an IIR

🔍 **What to remember**:

> FIR = safe and predictable, even if slower or longer than IIR.

◈ 5. Multirate FIR Filters (later)

- Used in **decimation, interpolation**
- Comb + sinc-like filters

◈ Not needed now, but remember: FIRs scale well with **sample rate changes** — key for SDR, RF systems.

◈ Summary: What to Remember Beyond Basic Convolution

Concept	Why It Adds Value
FIR = Convolution	You already get this. ✓
Shape of $h[n]$ defines frequency response	This you've felt hands-on. ✓
FIR = Designed $h[n]$, not guessed	Think `firwin()`, windowed sinc
Symmetric $h[n]$ = linear phase	Preserves waveform shape
Filter length affects sharpness + delay	Crucial in real-time systems
FIR is unconditionally stable	Advantage over IIR