**🧩 Step 1: Self-Attention recap**

In **self-attention**, each token in a sequence looks at the **other tokens in the *same* sequence**.

* Example: In "Je suis étudiant", the word étudiant might attend to Je and suis to understand context.
* So: **Self-attention = "I look around inside my own group."**

**🧩 Step 2: Cross-Attention**

In **cross-attention**, a token in the **decoder** (the output side) looks at tokens from the **encoder** (the input side).

* Example: In French→English translation:
  + The decoder is trying to generate "student".
  + It looks back at the encoder outputs of "étudiant".
* So: **Cross-attention = "I look across at the other group for help."**

**🧮 Step 3: How it works in math terms**

* Queries (Q) come from the **decoder tokens** (the words being generated).
* Keys (K) and Values (V) come from the **encoder outputs** (the input sentence representations).
* The decoder uses attention to decide which parts of the input are most relevant to generating the next word.

**🖼️ Analogy**

Imagine two classrooms:

* **Class A (encoder)** = kids describing a story (input sentence).
* **Class B (decoder)** = kids writing the translation.

Each kid in Class B can raise their hand and ask:  
*"Hey, Class A, which of you has the info I need right now?"*

That’s **cross-attention**.

**🧑‍💻 In your model output**

You saw:

BaseModelOutputWithPoolingAndCrossAttentions(...)

That means:

* The model is an **encoder–decoder type model** (like BART, T5, Marian, etc.).
* Along with last\_hidden\_state and pooler\_output, it can also return **cross-attention weights** (if you set output\_attentions=True).
* Those cross-attention weights tell you: *“For this decoder token, how much did it attend to each encoder token?”*

✅ So the difference is:

* **Self-attention** = inside one sequence (encoder→encoder, or decoder→decoder).
* **Cross-attention** = decoder tokens looking at encoder outputs to ground generation in the input.