**1) Token**

**Definition**: The smallest unit of data processed by a model (e.g., words, sub words or a characters).

**Description**: Tokens are created by splitting input text into meaningful chunks. Used in NLP models   
**Example: The sentence "I know AI!" might tokenize to ["I", "know", "AI", "!"].**

**2) Input and Output**

**Input**: Data fed into a model (e.g., text, images, numerical values).

**Output**: The result generated by the model (e.g., predictions, classifications).

**Example**: Input: "Translate 'Hello' to French." → Output: "Bonjour."

**3) Token Embedded, Input Embedded, Position Embedded**

**Token Embedding**: Converts tokens into dense vectors to capture semantic meaning.

*Example*: The token "cat" → [0.2, -0.5, 0.7].

**Position Embedding**: Adds positional information to tokens (critical for transformers, which lack inherent sequence awareness).

*Example*: For the sequence ["A", "B", "C"], position embeddings might *be [0.1, 0.2], [0.3, 0.4], [0.5, 0.6].*

**Input Embedding**: Combined token + position embeddings used as model input.

*Example: Token embedding [0.2, -0.5] + position embedding [0.1, 0.2] → [0.3, -0.7].*

**4) V Bias**

**Definition**: Bias term added to the **Value** matrix in attention mechanisms.

**Description**: Adjusts the transformed value vectors to improve model flexibility.

*Example: In Value = (input × V\_weights) + V\_bias.*

**5) K Bias  
Definition**: Bias term added to the **Key** matrix in attention mechanisms.  
**Description**: Helps shift key vectors during transformation.  
*Example: Key = (input × K\_weights) + K\_bias.*

**6) Q Bias Definition**: Bias term added to the **Query** matrix in attention mechanisms.

**Description**: Adjusts query vectors to refine attention focus.

**Example**: *Query = (input × Q\_weights) + Q\_bias.*

**7) V Weights Definition**: Learned matrix to project input embeddings into **Value vectors**.

**Description**: Transforms input into vectors used to compute attention outputs.

*Example: A weight matrix of shape [embed\_dim, d\_model].*

**8) K Weights**

**Definition**: Learned matrix to project input embeddings into **Key vectors**.

**Description**: Keys determine how input tokens relate to each other.

**Example**: Used to compute similarity scores with query vectors.

**9) Q Weights**

**Definition:** Learned matrix to project input embeddings into Query vectors.

**Description:** Queries represent what a token is "searching for" in attention.

**Example:** In self-attention, queries and keys come from the same input.

**10) V Vectors**

**Definition**: Transformed input embeddings using **V\_weights** and **V\_bias**.

**Description**: Represent the "content" of tokens used to compute attention outputs.

*Example: For input token "apple", V vector might be [0.4, -0.1, 0.9].*

**11) K Vectors**

**Definition**: Transformed input embeddings using **K\_weights** and **K\_bias**.

**Description**: Used to calculate attention scores (how much focus each token gets).

*Example: A key vector for "dog" might be [0.3, 0.7, -0.2].*

**12) Q Vectors**

**Definition**: Transformed input embeddings using **Q\_weights** and **Q\_bias**.

**Description**: Represent "queries" to find relevant keys in attention computation.

**Example**: A query vector for "animal" might focus on keys like "dog" or "cat".

**Summary**These terms are foundational in transformer-based models (e.g., GPT, BERT). **Tokenization** breaks data into units, **embeddings** encode meaning and position, and **Q/K/V** components drive attention mechanisms by learning relationships between tokens. Biases and weights are trainable parameters that refine these transformations.