1. $c^{<t>} = \sum_{t'} \alpha^{<t,t'>} a^{<t'>}$
   $\alpha^{<t,t'>}$ = amount of "attention" $y^{<t>}$ should pay to $a^{<t'>}$. In plain terms, this equation is a sum of the attention and weights from each activation for predicting the next token from the sequence. In other words when an attention value is high or low, it will cause the corresponding activation to be scaled with it.

2. A decoder gets trained by trying to predict the next token from an input sequence. The loss is measured by how accurately it predicts the next token in the sequence. The encoder is a sub-module of the encoder-decoder sequence and they are trained together. The loss for the encoder comes from the decoder as it is partly responsible for the accuracy of the decoder.

3. The query, key, and value vectors are abstractions useful for calculating and thinking of attention. Multiplying our embeddings matrix with trained weight matrices will produce the query, key, and value matrices for self-attention. A Query is from the decoder hidden state, the key and value are from the encoder hidden states. Loss is a measure of the compatibility between query and key. The query is a vector that represents the word encoded, key and value is the "memory" used for retrieval of the target. The query measures its distance to the keys, and the keys correspond to specific output values.

4. Multi-headed attention works by assigning a weight or contribution each previous token has on the next token in the sequence. All of these weights are sent through Softmax so they all add up to one. Multiple attention heads allows us to pay different amounts of attention to different areas of the sequence, and not just one area. In this way multi-headed attention facilitates longer term memory and can learn longer more complex sequences.

5. Positional encodings are a way to account for the order of words from an input sequence. Transformers add a vector to each input embedding. The positional encodings help us determine the distance between different tokens and the order that they should appear in. Models without positional encodings could suffer from terrible predictions of phrases and longer sequences by inadvertently breaking grammatical laws of many languages. One example of this is placing adjectives before nouns in English ie "The apple green was picked early from the tree apple". The words are still around where they need to be, it just does not make sense to us. If this sentence was translated directly to Spanish ("La manzana verde estuvó cosechedado temprano del arbol") however, this ordering may make more sense. In this way positional encodings are a great tool for accurate language translation.