Let's jump in and learn about the multi head attention mechanism. The notation gets a little bit complicated, but the thing to keep in mind is basically just a big for loop over the self attention mechanism that you learned about in the last video. Let's take a look each time you calculate self attention for a sequence is called a head.

And thus the name multi head attention refers to if you do what you saw in the last video, but a bunch of times let's walk through how this works. Remember that you got the vectors Q K and V for each of the input terms by multiplying them by a few matrices, W Q W K and W V. With multi head attention, you take that same set of query key and value vectors as inputs. So the q, k, v values written down here and calculate multiple self attentions. So the first of these, you multiply the k, q, v matrices with weight matrices, w one q, w one k and w one v. And so these three values give you a new set of query key and value vectors for the first words. And you do the same thing for each of the other words. For the sake of intuition, you might find it useful to think of w one q, w one k and w one v as being learned to help ask and answer the question, what's happening there? And so this is just more or less the self attention example that we walked through earlier in the previous video. After finishing you may think, we have w q, w one q, w one k, w one v, I learn to hope you ask and answer the question, what's happening?

And so with this computation, the word "visiite" gives the best answer to what's happening, which is why I've highlighted with this blue arrow over here to represent that the inner product between the key for [visitie] has the highest value with the query for [Afirce], which is the first of the questions we'll get to ask. So this is how you get the representation for [Africe] and you do the same for Jane, [visitie] and the other words [en] [septembre].

. So you end up with five vectors to represent the five words in the sequence. So this is a computation you carry out for the first of the several heads you use in multi head attention. And so you would step through exactly the same calculation that we had just now for [Africe] and for the other words and end up with the same attention values, a one through a five that we had in the previous video. But now we're going to do this not once but a handful of times

So that rather than having one head, we may now have eight heads, which just means performing this whole calculation maybe eight times. And so so far, we've computed this quantity of attention with the first head indicated by the subsequent one in these matrices. And the attention equation is just this, which you had previously seen in the last video as well. Now, let's do this computation with the second head. The second head will have a new set of matrices. I'm going to write WQ two, WK two and WV two that allows this mechanism to ask and answer a second question. So the first question was what's happening? Maybe the second question is when is something happening?

And so instead of having W one here in the general case, we will have here Wi and I've now stacked up the second head behind the first one was the second one shown in red. So you repeat a computation that's exactly the same as the first one but with this new set of matrices instead. And you end up with in this case maybe the inner product between the september key and the [AFrice] query will have the highest inner product. So I'm going to highlight this red arrow to indicate that the value for september will play a large role in this second part of the representation for [Africe].

Maybe the third question we now want to ask as represented by W Q three, W K three and WV three is who, who has something to do with Africa? And in this case when you do this computation the third time, maybe the inner product between Jane's key vector and the [AFrice] query vector will be highest and self highlighted this black arrow here. So that Jane's value will have the greatest weight in this representation which I've now stacked on at the back. In the literature, the number of heads is usually represented by the lower case letter H. And so H is equal to the number of heads.

And you can think of each of these heads as a different feature. And when you pass these features to a new network you can calculate a very rich representation of the sentence. Calculating these computations for the three heads or the eight heads or whatever the number, the concatenation of these three values or A values is used to compute the output of the multi headed attention. And so the final value is the concatenation of all of these h heads.

And then finally multiplied by a matrix W. Now one more detail that's worth keeping in mind is that in the description of multi head attention, I described computing these different values for the different heads as if you would do them in a big four loop. And conceptually it's okay to think of it like that. But in practice you can actually compute these different heads' values in parallel because no one has value depends on the value of any other head. So in terms of how this is implemented, you can actually compute all the heads in parallel instead of sequentially. And then concatenate them multiply by W zero. And there's your multi headed attention. Now, there's a lot going on on the slide. Thanks for sticking with me all the way to the end of this video. In the next video, I'm going to use a simplified icon. We're going to use this little diagram here to denote this multi head computation. So it takes as input matrices Q, K and V. So these values down here and outputs this value up here. So in the next video, when we put this into the full transformer network, I'm going to use this little picture to denote all of these computations denoted on the slide. So, congratulations. In the last video you learned about self attention. And by doing that multiple times, you now understand the multi head attention mechanism, which lets you ask multiple questions for every single word and learn a much richer, much better representation for every word. Let's now put all this together to build the transformer network. Let's go to the next video to see that.