Top_p (nucleus sampling):

1. The model looks down the list from the most probable word.
2. It keeps adding words to a "candidate pool" until the *sum* of their probabilities reaches the chosen `top_p` value (e.g., 0.9).
3. It then picks the next word *only* from this pool, ignoring all the less likely words outside it.
4. Effect: Lower `top_p` (e.g., 0.5) = smaller, more probable pool = more focused/predictable text. Higher `top_p` (e.g., 0.95) = larger pool including more unlikely words = more diverse/random text.

Top_k:

1. The model simply takes the `k` most probable words from the top of the list, ignoring all others.
2. It then picks the next word *only* from these `k` candidates.
3. Effect: Lower `top_k` (e.g., 10) = fewer candidates = more focused/predictable text. Higher `top_k` (e.g., 500) = more candidates = more diverse/random text