How to Use the Pumping Lemma
for Regular Languages

When you are given a language \( L \) and are using the pumping lemma to prove that it is not regular, do this…

1. Assume \( L \) is regular.
   
   If it is, then the P.L. would apply. So, there is some \( n \) such that any string longer than \( n \), say the string \( x \), can be broken up into substrings \( u, v, w \) such that
   
   - \(|uv| \leq n\) — which means the pumping part, \( v \), lies within the first \( n \) characters
   - \(|v| > 0\) — which means there is at least one character in \( v \)
   - \(uvw\) is the string \( x \)
   - And \(uv^mw\) is also in \( L \) for any \( m \geq 0\) — that is, pumping at \( v \) won’t take the string out of the language.

   The key observation is that this is true for any sufficiently long \( x \). You will pick a particular one from which you can get a contradiction.

2. Choose some string longer than \( n \), where \( n \) is the (unknown) value guaranteed to exist by the P.L.
   
   Choose this string with a view to what you’re about to do, so you can pump it up and derive a contradiction.

3. You know \( v \) is within the first \( n \) characters, so take your string and pump it up (or deflate it) to get a new string.

4. Show that this new string is \emph{not} in \( L \).
   
   Since the P.L. says it \emph{is} in \( L \) if \( L \) is regular, it must be the case that \( L \) is \emph{not} regular.

5. You are done. Celebrate.