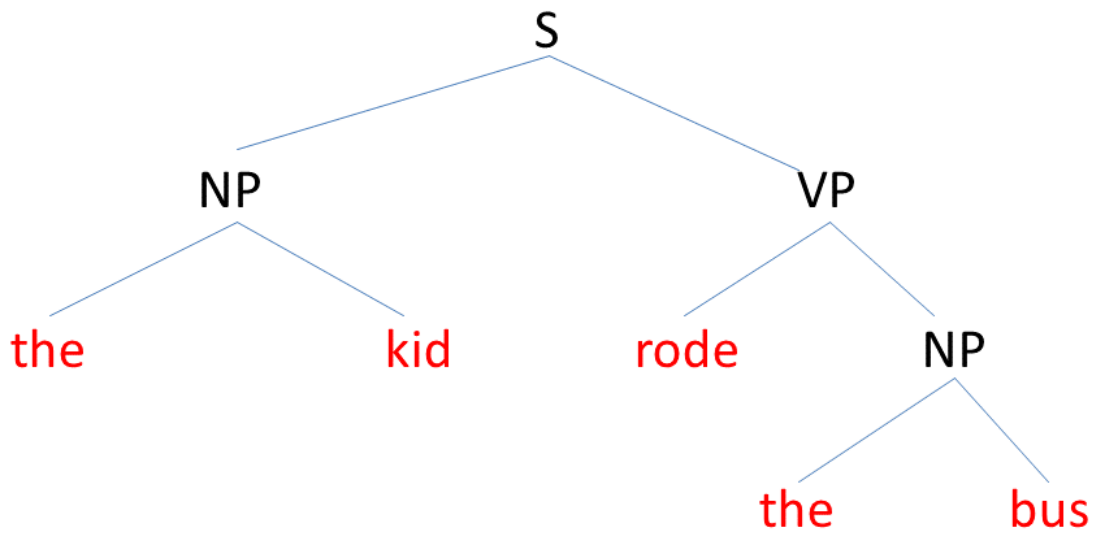*c-command* is a particular kind of relationship between words (or other elements, like morphemes) in a syntactic structure. It is going to sound very abstract, but it turns out it's a very useful concept for understanding many syntactic phenomena, including binding relationships between noun phrases.

C-command is easiest to see in a syntax tree, so let's consider an example. Take a sentence like "The kid rode the bus". Here's how we drew a tree for this sentence in the "Basics of syntax" module:
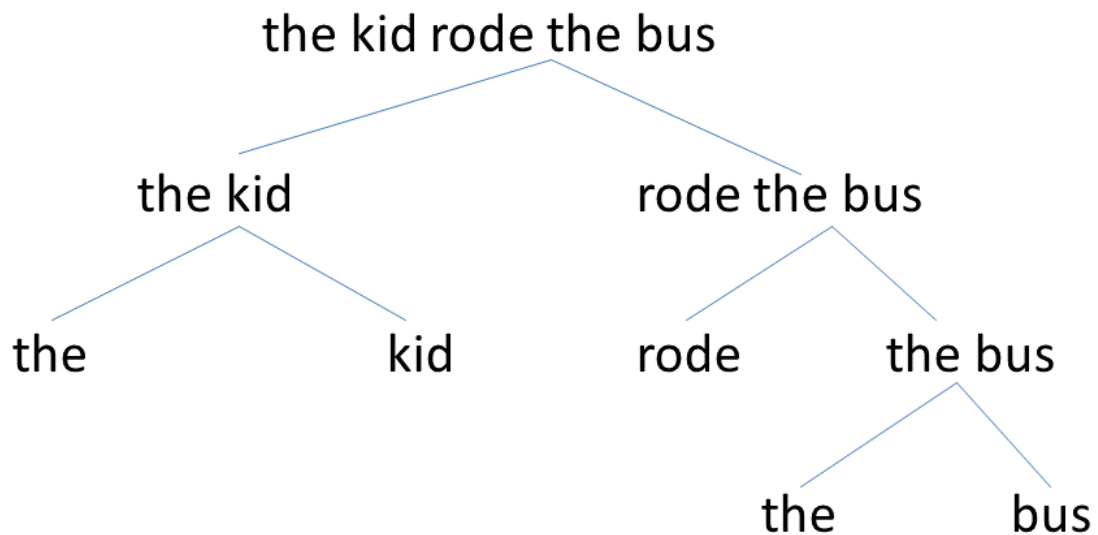
Let's make this tree a little bit more formal by giving syntactic labels to each word and phrase:

```
                              S
                     ╱                ╲
                  NP                    VP
                ╱    ╲               ╱      ╲
             the     kid         rode       NP
                                          ╱    ╲
                                        the     bus
```

In a tree like this, we call things "sisters" if they are both immediately under the same node. For example, "the" and "kid" are sisters because they are both immediately under the same NP; "rode" and the NP "the bus" are sisters because they are immediately under the same VP, etc. "Rode" and "bus" are *not* sisters, because "bus" isn't immediately under the VP (instead it's part of a bigger NP which is immediately under the VP). A head and its complement are sisters; a phrase and its adjunct are sisters.

Two words are in a *c-command* relationship if they are sisters, or if one word is *part of* the other word's sister. This sounds abstract, but it's easier to see on a syntax tree. In short, if you start at one word (like "rode"), then trace a line up to the *first* split in the tree (the VP above "rode"), and then trace a line down, everything you can find below is c-commanded by the first word. In other word, "rode" c-commands every word underneath the VP: it c-

commands every word that you can reach by first tracing a line up to **VP** and then tracing a line as far down as you want. That means "rode" c-commands the second "the", and "bus", and the whole **NP** "the bus". But "rode" does *not* c-command "kid"; you can't draw a line from "rode" to "kid" by going up to **VP** and then down (to get from "rode" to "kid", you would have to go up *two* levels first, and that's not allowed when it comes to determining c-command relationships).

Another way to put that is that a word c-commands its sister, and all the words that make up its sister. The sister of "rode" is the **NP**; thus, "rode" c-commands the **NP** itself, and every word that is part of that **NP**. By the same token, the first "the" c-commands "kid", and nothing else.

You could also think of this with an analogy to a family tree. If you imagine the syntax tree above is like a family tree, you could say that I c-command my siblings (e.g. my brothers and sisters) and all of their kids (e.g. my nieces and nephews), but I don't c-command my cousins, my parents, my aunts and uncles, etc. (This analogy is imperfect because family relationships can be messy and include some kids of relationships that don't have a correlate in a syntax tree; but anyway, you get the picture.)