**Abstraction**

Although many other answers here defined abstraction as generalization, I personally think that definition is misguided. I would say that generalization is actually a specific *type of* abstraction, not the other way around. In other words, all generalizations are abstractions, but all abstractions are *not* necessarily generalizations.

Here's how I like to think of abstraction:



Would you say the image there is a tree? Chances are you would. But is it *really* a tree? Well, of course not! It's a bunch of pixels made to look like something we might call a tree. We could say that it's an abstraction of a real tree. Notice that several visual details of the tree are omitted. Also, it does not grow, consume water, or produce oxygen. How could it? it's just a bunch of colors on a screen, represented by bytes in your computer memory.

And here is the essence of abstraction. It's a way of simplifying things so they are easier to understand. Every idea going through your head is an abstraction of reality. Your mental image of a tree is no more an actual tree than this jpeg is.

In programming, we might use this to our advantage by creating a Tree class with methods for simulated growing, water consuming, and oxygen production. Our creation would be something that represents our experience of actual trees, and only includes those elements that we really care about for our particular simulation. We use abstraction as a way of representing our experience of something with bytes and mathematics.