"Non-numbers" | Coursera

"Non-numbers"

"Non-numbers"

It is worth restating: ***everything is a number***. This rule is fundamental to understanding how computers work and is one of the most important concepts in programming. For every variable you create in any programming language, the value of that variable—the data that you place "in the box" of every conceptual diagram you draw—is stored in your computer as a series of zeros and ones. This fact is easy to accept for a positive integer, whose base 10 representation is simply converted to base 2 and then stored in a series of bits. Understanding how negative numbers and floating point numbers are also represented as a series of zeros and ones may be a little less straightforward, but is still appeals to our general intuition about numbers.

Extending this rule to things that do not seem like numbers—words, colors, pictures, songs, movies—may seem like a much harder conceptual leap. However, with our newfound understanding that computers can only operate on numbers, we must realize that all of these things must be numbers too—after all, our computers operate on them regularly.

Finding a way to encode these "non-number" data types is a simply a matter of coming up with a new convention for encoding the information as bits, and interpreting the bits to mean the original information. These new conventions are not included as basic data types of the C programming language (though some of them are basic data types in languages other than C). Instead, new types are formed by combining the basic types to achieve the programmer's goals. These more complex types may be widely accepted programming conventions (like the representation of strings), or may be something done by one single programmer specific to their programming task.

✓

Completed