## Literals and Identifiers

Expressions like 27 or 'hello' are called literals, coming from the fact that they literally mean exactly what they say. They are distinguished from variables, whose value is not directly determined by their name.

The sequence of characters used to form a variable name (and names for other Python entities later) is called an identifier. It identifies a Python variable or other entity.

There are some restrictions on the character sequence that make up an identifier:

* The characters must all be letters, digits, or underscores \_, and must start with a letter. In particular, punctuation and blanks are not allowed.
* There are some words that are reserved for special use in Python. You may not use these words as your own identifiers. They are easy to recognize in Idle, because they are automatically colored orange. For the curious, you may read the full list:
* **False** **class** **finally** **is** **return**
* **None** **continue** **for** **lambda** **try**
* **True** **def** from **nonlocal** **while**
* **and** **del** **global** **not** **with**
* **as** **elif** **if** **or** **yield**
* **assert** **else** **import** **pass**
* **break** **except** **in** **raise**

There are also identifiers that are automatically defined in Python, and that you could redefine, but you probably should not unless you really know what you are doing! When you start the editor, we will see how Idle uses color to help you know what identifies are predefined.

Python is case sensitive: The identifiers last, LAST, and LaSt are all different. Be sure to be consistent. Using the Alt-/ auto-completion shortcut in Idle helps ensure you are consistent.

What is legal is distinct from what is conventional or good practice or recommended. Meaningful names for variables are important for the humans who are looking at programs, understanding them, and revising them. That sometimes means you would like to use a name that is more than one word long, like price at opening, but blanks are illegal! One poor option is just leaving out the blanks, like priceatopening. Then it may be hard to figure out where words split. Two practical options are

* underscore separated: putting underscores (which are legal) in place of the blanks, like price\_at\_opening.
* using camel-case: omitting spaces and using all lowercase, except capitalizing all words after the first, like priceAtOpening