Declaring Variables:

In computer programming, declarations are most commonly used for functions, variables, constants, and classes, but can also be used for other entities such as enumerations and type definitions.  Beyond the name (the identifier itself) and the kind of entity (function, variable, etc.), declarations typically specify the [data type](https://en.wikipedia.org/wiki/Data_type) (for variables and constants), or the [type signature](https://en.wikipedia.org/wiki/Type_signature) (for functions); types may also include dimensions, such as for arrays. A declaration is used to announce the existence of the entity to the [compiler](https://en.wikipedia.org/wiki/Compiler); this is important in those [strongly typed](https://en.wikipedia.org/wiki/Strongly_typed)  languages that require functions, variables, and constants, and their types to be specified with a declaration before use, and is used in [forward declaration](https://en.wikipedia.org/wiki/Forward_declaration). In informal usage, a "declaration" refers only to a pure declaration (types only, no value or body), while a "definition" refers to a declaration that includes a value or body. However, in formal usage (in language specifications), "declaration" includes both of these senses, with finer distinctions by language: in C and C++, a declaration of a function that does not include a body is called a [function prototype](https://en.wikipedia.org/wiki/Function_prototype), while a declaration of a function that does include a body is called a "function definition". By contrast in Java declarations always include the body, and the word "definition" has no technical meaning in Java.

Defined Variables:

A variable is a value that can change, depending on conditions or on information passed to the program. Typically, a program consists of instruction s that tell the computer what to do and data that the program uses when it is running. The data consists of constants or fixed values that never change and variable values (which are usually initialized to "0" or some default value because the actual values will be supplied by a program's user). Usually, both constants and variables are defined as certain data type “s”. Each data type prescribes and limits the form of the data. A variable is also is a symbolic name for (or reference to) information. The variable's name represents  what information the variable contains. They are called  variables  because the represented information can change but the operations on the variable remain the same. In general, a program should be written with "Symbolic" notation, such that a statement is always true  symbolically.

For variables, definitions assign values to an area of memory that was reserved during the declaration phase. For functions, definitions supply the function body. While a variable or function may be declared many times, it is typically defined once (in C++, this is known as the One Definition Rule or ODR).

Dynamic languages such as JavaScript or Python generally allow functions to be redefined, that is, re-bound; a function is a variable much like any other, with a name and a value (the definition).

Undefined Variables:

An undefined variable  in the [source code](https://en.wikipedia.org/wiki/Source_code) of a [computer program](https://en.wikipedia.org/wiki/Computer_program) is a [variable](https://en.wikipedia.org/wiki/Variable_(programming)) that is accessed in the code but has not been previously [declared](https://en.wikipedia.org/wiki/Declaration_(computer_science)) by that code. In some programming languages an implicit declaration is provided the first time such a variable is encountered at compile time. In other languages such a usage is considered to be an error, which may resulting in a diagnostic message.