CS 1173: Introductory Concepts

Nomenclature of structures: Data is stored in a variety of formats.

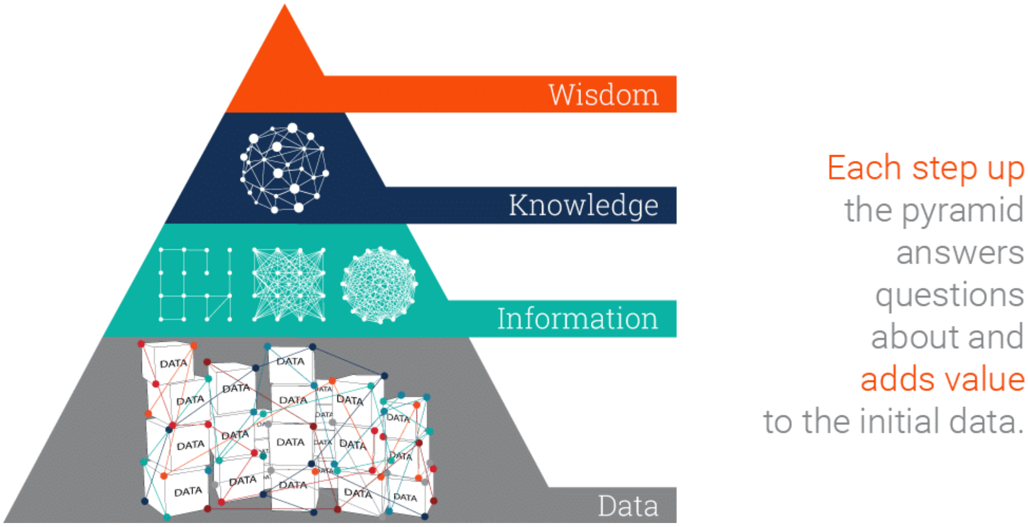
If you have a single data point, it can generically be called a **scalar** – a single number. Most software, MATLAB included, stores it in a single place. Anytime you refer a variable containing a scalar, the value of the scalar is what you are working with. For example, you need to keep track of the number of birds you have, which is 9. *Numbirds* = 9, so anytime you refer to *Numbirds*, you are working with the value 9 (until you change it).

If you have a collection of numbers that is best stored in a single column or row, that is a **vector**. The sequence of the numbers may or may not be important. To reference the entire vector set of numbers, you use the variable name. To reference an individual value of the vector, you use the variable name and the index of the individual value. For example, you need to track the weight of your 9 birds, and you create a variable called *birdweight*. To access all 9 values, use *birdweight*. To access the weight of the 3rd bird, use *birdweight(3)*. MATLAB doesn't differentiate whether the vector data set is a row or a column. A vector is an example of 1 dimensional data – it has length.

A more complicated collection of numbers that requires both rows and columns is called either **matrices** or **arrays** – frequently used interchangeably. An array has BOTH rows and columns, and is an example of 2 dimensional data. Again, to reference the entire set of data, you use the variable name. But since these have both rows and columns, you must address each when not working with the entire set. MATLAB addresses the rows first (1st dimension) and the columns second (2nd dimension). For example, if you want to keep track of the food consumption of your 9 birds, you create a variable called *food_intake_2020*, and you plan to have each bird have a column – that is, the first bird's consumption is in the first column, the 2nd in the second, and so on. You will use the Julian date for the rows. The Julian date is the number of days since January 1, so row 1 is January 1, January 2 is row 2, February 1 is row 32 (since there are 31 days in January) and so on. To access the consumption of the bird 3 on January 4, you use *food_intake_2020(4,3)*, where the 4 refers to the row (4th row) and 3 refers to the column (3rd bird).

DIKW Triangle:

Starting at the bottom is **data**. It is a collection of facts in a raw or organized form, such as numbers or characters. Without context, doesn't mean much. Reading a value off a graph is data. This is level 1 information.



Each step up the pyramid answers questions about and adds value to the initial data.

Putting some context to data makes it **information**. It has been cleared of most errors and further processed that makes it easier to measure, visualize or analyze for a purpose. This is level 2 information.

Digging into the information, asking 'who', 'what', 'when' or more, makes it more valuable. Asking the question 'how' and 'which means?' leads to the **knowledge** level. This level 3 information is where we use the information to address a problem or answer a question of interest to us. Here is where we uncover relationships and develop insights into the problem.

We use the insights and knowledge to reach the final stage '**Wisdom**' – level 4. Here is where we use the knowledge learned to apply to our problem to make a decision or decide on a path. Wisdom is knowledge applied in action.

Summarized from: https://www.ontotext.com/knowledgehub/fundamentals/dikw-pyramid/, accessed Aug 2019.