Terminology

Boolean: A data type that stores true/false values.
String: A text data type. This can be letters (characters), words, phrases, etc.

Variable: An object that is defined in code, the contents of which can change. In Lava the variables will live on the server and their values can be changed based on work done on the server.

Entity Type: The blueprints for a kind of data we can get from Rock. This will correspond to tables in our database. Person is an entity type that will always have an Id, First Name, Last Name, and Birthdate properties. When looking at a Person Record (singular entity, like Ted Decker or Admin Admin) you know since they belong to the Entity Type Person they will always have those fields.
Entity: There are two definitions for Entity.

One is synonymous with Entity Type.

The other is when we refer to a singular entity. We are talking about a variable that contains a specific record from the database. For example, Workflows have an “Attribute Set to Entity” action that will save whatever data was passed to the workflow and save it to the attribute of your choosing.

JSON: (JavaScript Object Notation) A standard format for displaying the data in a variable.
Object: A singular entity, written as `{}`
Property: The data within the object. Always written in “Key”: “Value” format.
Key: A string representing the name of the property.
Value: The value of the property, it can be any data type (string, int, bool, date, etc) or an object or array.
Array: A collection of data. The data could be objects, strings, boolean values, integers, etc. Written as `[]`
Part of a Person record:

```
{                                        The opening { of the object
      “Id”: 1234,                        The Id property, an integer
      “FirstName”: “Ted”,                The First Name property, a string
      “LastName”: “Decker”,              The Last Name property, a string
      “PhoneNumbers”:                    The Phone Numbers Property, an Array of PhoneNumbers
      [                                  The opening [ of the phone number array
            {                            The opening { of the first phone number object
                  “Id”: 3344,            The Phone Number Id Property, an integer
                  “Number”: “0123456789” The Number Property, a string
            }                            The closing } of the first phone number object
      ]                                  The closing ] of the Phone Numbers Array
}                                        The closing } of the object
```

Each property, except the last listed, will have a comma after it
The same goes for values in an array

Quick Tips

Create a New Array in Lava

Sometimes the filtering you can do in Lava with **Where** isn't enough, you need a more complex set of rules (like if a date property is within a certain range) in this case you might want to loop through your values and then use the **AddToArray** to only add the items meeting your criteria to the final array you will print. But how do you create a new variable of an array type?

```
{% assign myArray = ‘[]’ | FromJSON %}
```

Create an empty array with the **FromJSON** Lava Filter.

Attribute Values Are Not Properties Directly On The Entity

An attribute like a Person's Allergy cannot be accessed like First Name on a Person Record because Allergy is not a property of Person. It is important to remember this because when using Lava Filters like **Where** you will not have access to attributes, only properties. It is also the reason we have **Sort** and **SortByAttribute. Sort** *only works* with properties.

You Generally Cannot Use . Notation to Chain Properties In Lava Filters or Entity Commands

```
{{ Person.PrimaryFamily | Where:'Person.MaritalStatusValue.Value','Married' }}
```

That does not work.

```
{% person where:'MaritalStatusValue.Value == "Married"' %}
    {{personItems | Size}}
{% endperson %}
```

Also does not work.