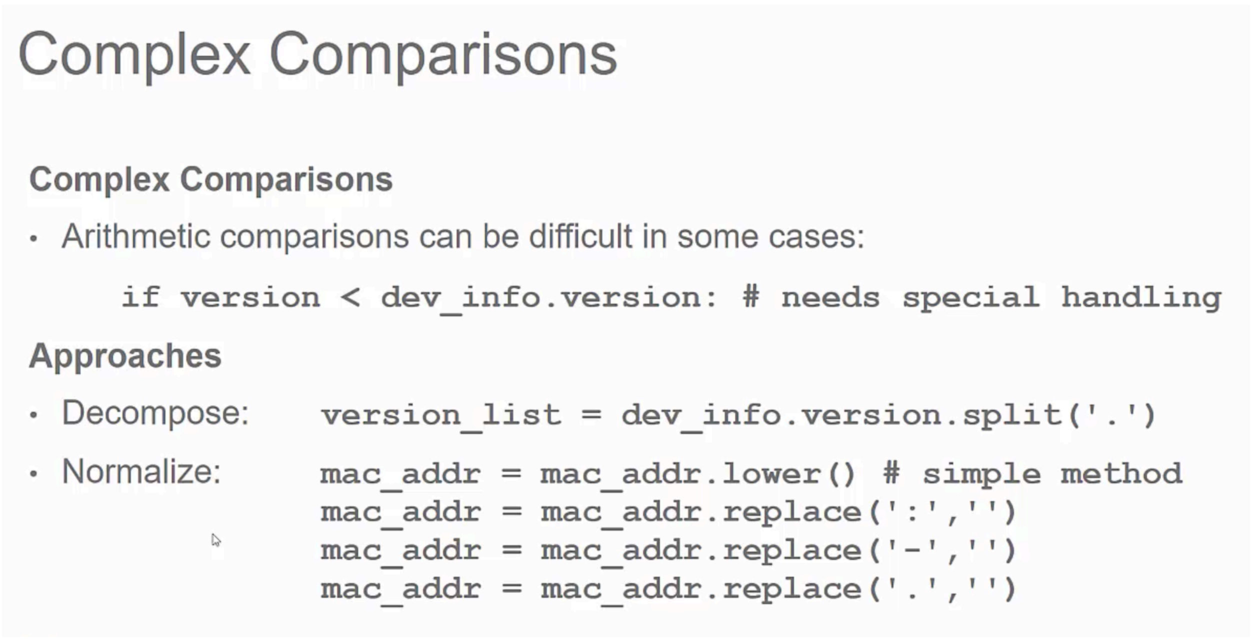

# Complex Comparisons

## Complex Comparisons

- Arithmetic comparisons can be difficult in some cases:

```
if version < dev_info.version: # needs special handling
```

## Approaches

- Decompose:

```
version_list = dev_info.version.split('.')
```

- Normalize:

```
mac_addr = mac_addr.lower() # simple method
mac_addr = mac_addr.replace(':','')
mac_addr = mac_addr.replace('-','')
mac_addr = mac_addr.replace('.','')
```

There are certain comparisons that would be beneficial in a networking environment, but are impractical using basic comparison methods. For example:

- **Versions:** Version strings can vary widely, and therefore it is not possible to do a straight string comparison to determine if one version is less than, or greater than, the other. For example, some version strings have a major version and a minor version that are separated by dots; some have major, minor, and build number, and so on.
- **Addresses:** IP addresses cannot be compared lexicographically either—an IP address of 198.1.1.1 will evaluate as less than IP address 2.1.1.1.

There are two approaches that can be used to handle complex comparisons. The two approaches can be used together or separately.

- **Decompose:** One strategy is to take an otherwise complex string, and separate it into its constituent parts. For example, with a version number, the example shows splitting the version string into a list, using the dot '.' to separate the items. In this way, it would be possible to examine major versions against each other, then the minor versions.

```
version_list = dev_info.version.split('.')
```

- **Normalize:** Another strategy would be to take the item in question and put it into a normalized form. A good example would be for dealing with MAC addresses, which can be presented in CLIs with colons (':'), or hyphens ('-') every two bytes. Sometimes they are even represented with dots ('.') every four bytes. Normalizing by changing all characters to lower case, then removing the hyphens, colons, or dots, helps to make all MAC address forms 'equal', so that they can be compared against one another, no matter where they came from.

  ```
  mac_addr = mac_addr.lower() # get mac address into lower case
  mac_addr = mac_addr.replace(':','') # strip out colons
  mac_addr = mac_addr.replace('-','') # strip out dashes
  mac_addr = mac_addr.replace('.','') # strip out dots
  ```

  The result is that the mac address has had all the special characters removed, which is good for computer programs and their ability to compare data without confusing and inconsistent visual aids.