Part 4

- Importing into the Global Namespace
- Using function without specifying the namespace before the function call
- Why we shouldn’t import *
- A better way to import our modules/functions

The Global Namespace

- So far we have been using

`import module_name` or

`import module_name as alias`

- This creates a separate namespace.
- We could import into the global namespace. Then we can use the functions directly without referring to any namespace at all.

```
from validate_module import *
```

- Let’s check it out.

```python
from validate_module import *

while True:
    user_input = input("Enter a number: ")
    user_input = convert_to_int(user_input)
    if (is_integer(user_input)):
        break
```

```
Validate User Input Imported.
Enter a number: No
Enter a number: Why should I?
Enter a number: hmm
Enter a number: ok
Enter a number: 42!!!!!
Enter a number: 4 2
Enter a number: 42

Process returned 0 (0x0)
execution time : 10.187 s
Press any key to continue . . .
```

import *

- You should not import *

A better import solution

```
from validate_module import is_integer,
convert_to_int
```

- Now it is quite clear where the functions have been imported from
- We could take this one step further and create aliases
- Let’s see an example

```python
from validate_module import is_integer as is_int,
convert_to_int as to_int
while True:
    user_input = input("Enter a number: ")
    user_input = to_int(user_input)
    if (is_int(user_input)):
        break
```

```
Validate User Input Imported.
Enter a number: No
Enter a number: Why should I?
Enter a number: hmm
Enter a number: ok
Enter a number: 42!!!!!
Enter a number: 4 2
Enter a number: 42

Process returned 0 (0x0)
execution time : 10.187 s
Press any key to continue . . .
```

So far…

- We have created a simple module
- We have created a main program and imported the module
  - import validate_module
  - import validate_module as check
  - from validate_module import *
  - from validate_module import is_integer, convert_to_int
  - from validate_module import is_integer as is_int, convert_to_int as to_int
- DocStrings
- Help
- Namespaces

End of Part 4

- Time for another break
- Check out the next video when ready and I'll discuss the Standard Library and how to install and import non standard libraries