Welcome Back

- Lets look at implementing a module
- We will also discuss getting help on our module and namespaces

Validate Module

- Create a new module called validate_module.py
- Create a docstring that explains the purpose of the module
- Create a function that checks if a value is an integer
- Create a docstring for the function
- Get the function to return True if the value is an integer otherwise return false

```
""" A bunch of functions to validate user
    input."""
print("Validate User Input Imported.")
def is_integer(value):
  """ Returns True if the value is an integer
      otherwise it will return False """
  data_type = str(type(value))
  if ("int" in data_type):
    return True
  return False
```

Using our Module

- Import our module into a new program
- ask the user to enter a number
- call our is_integer function
- store return value in a variable called an_int
- output “yep, an integer” or “Nope, not an integer” based on the value that was returned.

```
import validate_module
user_value = input("Enter a number:")
an_int = validate_module.is_integer(user_value)
if (an_int):
    print("yep, an integer")
else:
    print("Not an integer")
```

Getting help on Modules

- help on a module that is not familiar
- Having a peek into the namespace of the module

```
import validate_module
help(validate_module)
print(dir(validate_module))

help(validate_module.is_integer)
```

The Default Stuff in the Namespace

```
[
'__builtins__’,
'__cached__’,
'__doc__’,
'__file__’,
'__loader__’,
'__name__’,
'__package__’,
'__spec__’,
'is_integer’]
```

Question

- Given our program so far
- If the user enters the value 42
- What will be the result?

Implementing a Module End

- Time to take a short break?

- Have a think about the question. Do you already know what will happen?

- Join again by clicking the next video when you are ready