ch_7_assignment

March 21, 2023

Ch_7_assignment

```
[ ]: from IPython.core.interactiveshell import InteractiveShell
     InteractiveShell.ast_node_interactivity = 'all'
```

1 Using Method

1.1 Modules, Classes and Methods

```
[ ]: # A method is another kind of function that is attached to a particular type.
     str.capitalize("browning")
     str.center("Sonnet 43", 26)
     str.count("How do I love thee? Let me count the ways.", "the" )
```

```
[ ]: 'Browning'

[ ]: '        Sonnet 43         '

[ ]: 2
```

1.2 Calling Methods the Object-Oriented Way

```
[ ]: # Python provides a shorthand form for calling a method where the object␣
       ↪appears first and then the method call.
     str.capitalize('browning')
     print()
     'browning'.capitalize()
     print()
     str.center('Sonnet 43', 26)
     print()
     'Sonnet 43'.center(26)
     print()
     str.count('How do I love thee? Let me count the ways.', 'the')
     print()
     'How do I love thee? Let me count the ways.'.count('the')
     print()
```

```
('TTA' + 'G' * 3).count('T')
```

```
[ ]: 'Browning'
```

```
[ ]: 'Browning'
```

```
[ ]: '        Sonnet 43         '
```

```
[ ]: '        Sonnet 43         '
```

```
[ ]: 2
```

```
[ ]: 2
```

```
[ ]: 2
```

1.3 Exploring String Methods

```
[ ]: # Method calls look almost the same as function calls, except that in order to␣
      ↪call a method we need an object of the type associated with that method.
     'species'.startswith('a')
     print()
     'species'.startswith('spe')
     print()

     #There is also an endswith method.
     'species'.endswith('a')
     print()
     'species'.endswith('es')
```

```
[ ]: False
```

```
[ ]: True
[ ]: False
```

```
[ ]: True
```

```
[ ]: # Sometimes strings have extra whitespace at the beginning and the end
     compound = '    \n  Methyl \n butanol  \n'
     compound.lstrip()
     print()
     compound.rstrip()
     print()
     compound.strip()
```

```
[ ]: 'Methyl \n butanol  \n'
```

```
[ ]: '    \n  Methyl \n butanol'
```

```
[ ]: 'Methyl \n butanol'
```

```
[ ]: # String method swapcase changes lowercase letters to uppercase and uppercase␣
      ↪to lowercase.
     'Computer Science'.swapcase()
```

```
[ ]: 'cOMPUTER sCIENCE'
```

```
[ ]: # Here we show that we can substitute a series of strings into a format string.
     '"{0}" is derived from "{1}"'.format('none', 'no one')
     print()
     '"{0}" is derived from the {1} "{2}"'.format('Etymology', 'Greek', 'ethos')
     print()
     '"{0}" is derived from {2} "{1}"'.format('December', 'decam', 'Latin')
```

```
[ ]: '"none" is derived from "no one"'
```

```
[ ]: '"Etymology" is derived from the Greek "ethos"'
```

```
[ ]: '"December" is derived from Latin "decam"'
[ ]: # Next, using string method format, we'll specify the number of decimal places to round a number to.
     my_pi = 3.14159
     'Pi rounded to {0} decimal places is {1:.2f}.'.format(2, my_pi)
     print()
     'Pi rounded to {0} decimal places is {1:.3f}.'.format(3, my_pi)
     print()
     'Pi rounded to {} decimal places is {:.3f}.'.format(3, my_pi)
```

```
[ ]: 'Pi rounded to 2 decimal places is 3.14.'
```

```
[ ]: 'Pi rounded to 3 decimal places is 3.142.'
```

```
[ ]: 'Pi rounded to 3 decimal places is 3.142.'
```

```
[ ]: 'Computer Science'.swapcase().endswith('ENCE')
```

```
[ ]: True
```

1.4 What are those underscore?

```
[ ]: # Any method (or other name) beginning and ending with two underscores is considered special by Python.
     'TTA' + 'GGG'
     print()
     'TTA'.__add__('GGG')
```

```
[ ]: 'TTAGGG'
```

```
[ ]: 'TTAGGG'
```

```
[ ]: abs(-3)
     print()
     (-3).__abs__()
     print()
     -3 .__abs__()
     print()
     -(3 .__abs__())
     print()
     3 + 5
     print()
```

```
3 .__add__(5)
print()
3 > 5
print()
3 .__gt__(5)
print()
5 > 3
print()
5 .__gt__(3)
```

```
[ ]: 3
```

```
[ ]: 3
```

```
[ ]: -3
```

```
[ ]: -3
```

```
[ ]: 8
```

```
[ ]: 8
```

```
[ ]: False
```

```
[ ]: False
```

```
[ ]: True
```

```
[ ]: True
```

[ ]:

```python
import math
math.sqrt.__doc__

print()
print(math.sqrt.__doc__)
print()
help(math.sqrt)
```

[ ]:

```
'Return the square root of x.'
```

```

Return the square root of x.

Help on built-in function sqrt in module math:

sqrt(x, /)
    Return the square root of x.
```

Reference * Title: Physics Programming Lecture Note (INU) * Author: Jeongwoo Kim, Ph.D. * Availability: https://sites.google.com/view/jeongwookim

---