

Fizz Buzz homework

Work on an example implementation in class

`__name__` variable

The `__name__` variable is set to `__main__` when you are running the Python interpreter in interactive mode. When you run a python script the `__name__` variable is set to `__main__` in the script. When a module is imported, the `__name__` variable (defined in the scope of the module) is set to the name of the module (excluding the .py extension)

<https://docs.python.org/3/tutorial/classes.html#python-scopes-and-namespaces>

Python Functions and Modularity

The keyword `def` introduces a function *definition*. It must be followed by the function name and the parenthesized list of formal parameters. The statements that form the body of the function start at the next line, and must be indented.

1. Creating functions – `first_functions.py`
2. Functions with parameters – `fib1.py`
3. Functions without parameters – `get_time.py`
4. Functions with output – `helloworld1.py`
5. Functions without output – `none_type.py`
6. Functions which return a value – `fib2.py`
7. Functions which do not return a value (they actually return None) – `helloworld2.py`
8. `Print()` has output but no return value (it actually returns None)
9. Discuss namespaces, and how the namespace changes as you import a module – `name_test1.py` `name_test2.py`

Scope and namespaces

```
>>> def show_inner_scope():  
...     x = 3  
...     return dir()  
...
```

local vs. global variables

`scope_test1.py`

`scope_test2.py`

Reading from and Writing to files

“Iterable” objects. `with_open.py`

Reading (read and readlines)

```
>>> f = open('portfolio.csv', 'r')
>>> f.read()
'"Stock", "Date", "Shares", "Price"\n"IBM", "06/16/2017/", 100, 154.41\n"MSFT", "06/16/2017", 200, 69.37\n"YHOO", "06/16/2017", 175, 52.28\n"AAPL", "06/16/2017", 320, 143.14\n"ADR", "06/16/2017", 150, 37.21\n"DELL", "06/16/2017", 80, 13.65\n"KRX", "06/16/2017", 145, 2009.92\n"WMT", "06/16/2017", 100, 74.95\n"TGT", "06/16/2017", 200, 50.75\n"KSS", "06/16/2017", 75, 36.84\n"M", "06/16/2017", 140, 22.64\n"DDS", "06/16/2017", 235, 54.44\n"JCP", "06/16/2017", 300, 4.88\n"SHLD", "06/16/2017", 50, 6.76\n"LOW", "06/16/2017", 100, 79.57\n"HD", "06/16/2017", 200, 155.58\n"SNA", "06/16/2017", 150, 156.54\n"MAC", "06/16/2017", 80, 57.69\n"TYO", "06/16/2017", 220, 15.87\n"CAT", "06/16/2017", 300, 107.18\n"DE", "06/16/2017", 280, 127.16\n'
>>> f.read()
''
>>> f.seek(0)
0
>>> f.read()
'"Stock", "Date", "Shares", "Price"\n"IBM", "06/16/2017/", 100, 154.41\n"MSFT", "06/16/2017", 200, 69.37\n"YHOO", "06/16/2017", 175, 52.28\n"AAPL", "06/16/2017", 320, 143.14\n"ADR", "06/16/2017", 150, 37.21\n"DELL", "06/16/2017", 80, 13.65\n"KRX", "06/16/2017", 145, 2009.92\n"WMT", "06/16/2017", 100, 74.95\n"TGT", "06/16/2017", 200, 50.75\n"KSS", "06/16/2017", 75, 36.84\n"M", "06/16/2017", 140, 22.64\n"DDS", "06/16/2017", 235, 54.44\n"JCP", "06/16/2017", 300, 4.88\n"SHLD", "06/16/2017", 50, 6.76\n"LOW", "06/16/2017", 100, 79.57\n"HD", "06/16/2017", 200, 155.58\n"SNA", "06/16/2017", 150, 156.54\n"MAC", "06/16/2017", 80, 57.69\n"TYO", "06/16/2017", 220, 15.87\n"CAT", "06/16/2017", 300, 107.18\n"DE", "06/16/2017", 280, 127.16\n'
>>> f.seek(0)
0
>>> f.readline()
'"Stock", "Date", "Shares", "Price"\n'
>>> f.readline()
'"IBM", "06/16/2017/", 100, 154.41\n'
>>> f.readline()
'"MSFT", "06/16/2017", 200, 69.37\n'
>>> f.readline()
'"YHOO", "06/16/2017", 175, 52.28\n'
>>>
>>> f.seek(0)
0
>>> aline = True
>>> while aline:
...     aline = f.readline()
...     print(aline)
```

Writing (write and writelines)

```
>>> f = open('foo.txt', 'w')
>>> f.write('this is the first line\n')
23
>>> f.writelines(['aaa\n', 'bbb\n', 'ccc\n'])
>>> f.close()
>>>
[glenn@localhost]$ cat foo.txt
this is the first line
aaa
bbb
ccc
```

Python Modules and Imports

1. `if __name__ == '__main__':` this is not absolutely necessary in a module but putting this in your module is good Python coding practice
2. `random` – `random_gen.py` and `random_pw.py`
3. `import sys, dir(sys), sys` and `sys.argv` – `sys_argv.py`
4. `time` – `get_time.py`
5. A docstring is a string literal that occurs as the first statement in a module, function, class, or method definition. Such a docstring becomes the `__doc__` special attribute of that object. All modules should normally have docstrings, and all function and classes exported by a module should also have docstrings. – `do_nothing.py`
6. Import module
7. Import module as
8. From module import func
9. From module import *
10. Cover some of the `os` module functionality – `os_examples.py`

Command line arguments

Using `sys.argv` – `sys_argv.py`

Exceptions

`exceptions.py`

Homework for next week – Modified FizzBuzz

Write a program that:

- Takes three command line arguments `arg1`, `arg2`, `arg3`
- Casts them to integers
- Prints the numbers from 1 to `arg1`
- On multiples of `arg2`, instead of printing the number, print “Fizz”
- On multiples of `arg3`, instead of printing the number, print “Buzz”
- On multiples of `arg2` and `arg3`, print “FizzBuzz”
- Bonus – Exception Handling. Modify your script so that it fails gracefully with an error message if not enough arguments are supplied or if one of the arguments is not a positive integer

