

CIS433: Python Programming

Lecture 25: Documentation Strings

Yihsiang Liow

Agenda

- ♦ Study documentation attached to code

Documenting Functions

Try this:

```
def fib(n):  
    """  
    fib(n) returns the Fibonacci of n if n>=0.  
    Otherwise it returns 1.  
    """  
    # ... how do you do fibo again?  
    pass  
  
print(fib.__doc__)
```

Documenting Classes

Try this:

```
class C:
    """
```

```
This class is pretty ... useless
    """
```

```
    def g(self):
        """
```

```
Yeah ... and same for this method ...
        """
```

```
        pass
```

```
print(C.__doc__)
print(C.g.__doc__)
```

Optimized Bytecode

- You know that if you import a .py file, a corresponding .pyc file is generated.
- .pyc file contains bytecodes
- It is possible to compile optimized bytecodes. These files have .pyo extension.
- Right now .pyo files does not have a lot of optimization. The doc strings are removed in .pyo files

Block Comments

- You have already seen Python's line comment:

```
# Here's a hello world function  
def f():  
    print("hello world")
```

- To write block comments, use strings:

```
"""  
Here's a hello world  
function  
"""  
  
def f():  
    print("hello world")
```