

# Strings

## Python for Ecologists

Marcia Snyder, Tom Purucker, Tao Hong, Jonathan  
Flaishans

Ecological Society of America Workshop  
Minneapolis, MN

`purucker.tom@gmail.com`

August 1, 2013

# Accessing documentation in Python

- Use `dir()` and `help()` commands to access documentation

```
import sys
dir(sys)
help(sys)
help(sys.argv)
help(str)
dir(str)
help(len)
```

- or just Google it – 'python len'

# dunder methods

- "dunder" is short for double underline, e.g., `__init__`
- also known as special or magic methods
- `+` means different things for numbers and for strings

`3 + 5`

`"Hello_" + "world"`

- `__add__` for each type handles the different cases

# len v len()

- len - refers to the object

`help(len)`

- len() calls the function with a supplied argument

`len('Geospiza_magnirostris')`

# Single and Double quotes

- No preference between single and double quotes

```
aint = "ain't"
```

```
aint = 'ain"t'
```

# Strings

- Defining a string - single quote

```
darwins_finch = 'Geospiza_magnirostris'  
darwins_finch
```

- Quoting within the string - double quote

```
darwin_quote = " 'Great_is_the_power_of_steady_misrepresentation' "  
darwin_quote  
darwin_quote.lower()  
darwin_quote[13:18]  
darwin_quote[0] #Python is zero-based
```

- Slicing strings- extracts up to, but not including the second index

```
darwin_quote[0:1]  
darwin_quote[0:6]  
darwin_quote[:12]  
darwin_quote[12:]  
darwin_quote[:12] + darwin_quote[12:]
```

# Multi-line strings

- Multiline strings - triple quote

```
long_darwin_quote = '''There is grandeur in this view  
of life, with its several powers, having been originally  
breathed into a few forms or into one; and that, whilst  
this planet has gone cycling on according to the fixed  
law of gravity, from so simple a beginning endless  
forms most beautiful and most wonderful have been, and  
are being, evolved.'''  
long_darwin_quote  
print(long_darwin_quote)  
len(long_darwin_quote)
```

## C-like string formatting

```
>>> "%s_%s" % ( 'Hello' , 'Portland' )  
'Hello_Portland'
```



## Exercise 2- Run the script exer02\_strings.py

```
class TestStrings(unittest.TestCase):
    def test_strings(self):
        # Create the variable 'hola' and assign 'hello world'
        #*****

        self.assertEqual(hola, """hello world""")
        self.assertIsInstance(hola, str)

        # Create a string, 'hola2' that equals 'hola' multiplied by 2
        #*****

        self.assertEqual(hola2, 'hello_worldhello_world')

        # Create a triple quoted string
        # 'darwin_quote2' that has the following content:
        # Darwin said, "There is grandeur in this view of things."
        #*****

        self.assertEqual(darwin_quote2, 'Darwin_said,_"There_is_grandeur_in_this_view_of_things."')

        # Assign the method names of a string to a variable 'string_methods'
        # use 'dir()' to list them
        #*****

        self.assertEqual(string_methods, ['__add__', ...])

        # Create a variable where_is_gra that has holds the index of the
        # substring "gra" in the string darwin_quote2. (Find a string method to
        # figure it out)
        #*****

        self.assertEqual(where_is_gra, 23)
```