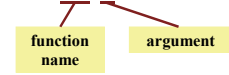


## Announcements

- Reading for next Tuesday: Chapter 3
- Please read the *First Edition* of *Think Python*
- Course Website:  
<http://www.cs.cornell.edu/courses/cs1110/2017sp/>

## Function Calls

- Python supports expressions with math-like functions
  - A function in an expression is a **function call**
  - Will explain the meaning of this later
- Function expressions have the form **fun(x,y,...)**



- **Examples** (math functions that work in Python):
  - `round(2.34)` Arguments can be any expression
  - `max(a+3,24)`

## Built-in Functions vs Modules

- The number of built-in functions is small
  - <http://docs.python.org/2/library/functions.html>
- Missing a lot of functions you would expect
  - **Example:** `cos()`, `sqrt()`
- **Module:** file that contains Python code
  - A way for Python to provide optional functions
  - To access a module, the `import` command
  - Access the functions using module as a *prefix*

## Example: Module math

```
>>> import math
>>> math.cos(0)
1.0
>>> cos(0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'cos' is not defined
>>> math.pi
3.141592653589793
>>> math.cos(math.pi)
-1.0
```

To access math functions

Functions require math prefix!

Module has variables too!

### Other Modules

- **io**
  - Read/write from files
- **random**
  - Generate random numbers
  - Can pick any distribution
- **string**
  - Useful string functions
- **sys**
  - Information about your OS

## Reading the Python Documentation

**Function name**

**Possible arguments**

**Module**

**What the function evaluates to**

<http://docs.python.org/library>

## Interactive Shell vs. Modules

- Launch in command line
- Type each line separately
- Python executes as you type
- Write in a text editor
  - We use Komodo Edit
  - But anything will work
- Load module with `import`

## Using a Module

### Module Contents

```
# module.py
```

**Single line comment**  
(not executed)

```
""" This is a simple module.
It shows how modules work"""
```

**Docstring** (note the Triple Quotes)  
Acts as a multiple-line comment  
Useful for *code documentation*

```
x = 1+2
x = 3*x
```

**Commands**  
Executed on import

## Using a Module

### Module Contents

### Python Shell

```
# module.py
```

```
>>> import module
>>> x
```

Traceback (most recent call last):  
File "<stdin>", line 1, in <module>  
NameError: name 'x' is not defined

```
x = 1+2
x = 3*x
```

**"Module data" must be**  
prefixed by module name

```
>>> module.x
9
```

**Prints docstring and**  
module contents

```
>>> help(module)
```

## Modules Must be in Working Directory!

Module you want is in this folder

Have to navigate to folder **BEFORE** running Python

## Modules vs. Scripts

### Module

### Script

- Provides functions, variables
  - Example:** temp.py
- import it into Python shell
 

```
>>> import temp
>>> temp.to_fahrenheit(100)
212.0
>>>
```

This is the command line prompt
- Behaves like an application
  - Example:** helloApp.py
- Run it from command line:
 

```
C:\> python helloApp.py
```

Hello World!

Files look the same. Difference is how you use them.

## Next Time: Defining Functions

### Function Call

- Command to **do** the function
- Can put it anywhere
  - In the Python shell
  - Inside another module

```
>>> import plusone
>>> plusone.plus(1)
2
>>> plusone.plus(2)
3
>>> plusone.plus(3)
4
>>>
```

Can call as many times as you want

### Function Definition

- Command to **do** the function
- Belongs inside a module

```
plusone.py
1 # plusone.py
2 # Walker M. White (wmw2)
3 # August 30, 2015
4 """Module with a function definition"""
5
6 def plus(n):
7     """Returns the value of n+1"""
8     return (n+1)
9
```

But only define function **ONCE**

## Functions and Modules

- Purpose of modules is **function definitions**
  - Function definitions are written in module file
  - Import the module to call the functions
- Your Python workflow (right now) is
  - Write a function in a module (a .py file)
  - Open up the Terminal/Command Prompt
  - Move to the directory with this file
  - Start Python (type python)
  - Import the module
  - Try out the function