Jython

Introduction to Jython programming

Agenda

- Module 1 - Introduction to Jython
- Module 2 - Jython language and semantics
- Module 3 - Data types
- Module 4 - Regular expressions
- Module 5 - Functions, debugging, modules, and packages
- Module 6 - Objects, classes and exceptions
- Module 7 - Java integration
- Module 8 - Testing
- Module 9 - System programming
- Module 10 - Conclusion

Topics

- Numeric types
- Collection types
- Comprehension
- File operations
- Standard Library
- Input and Output
- Quiz
- Q & A

Jython

Numeric types

Types of numbers

- int
- long
- float
- complex

int

- Arbitrary length
  - Two types of integral numbers
    - Traditional fixed size integers
    - Long integers
      - Conversion performed automatically
- In Jython, ints are represented as Java integer types
  - 32-bit signed integers
  - Literals are string of digits
    - No decimal point
    - Optional "+" (positive) or "-" (negative) sign
    - If literal starts with "0o" or "OO", interpreted as octal
    - If literal starts with "x", "X", "0x", or "0X", interpreted as hexadecimal

PEP 3127: https://www.python.org/dev/peps/pep-3127/

long

- Integral integer of arbitrary length
  - **Not** Java long type
  - Instances of *`java.math.BigInteger`*
- Literals follow same rules as integer literals
  - Must end with "l" (ell) or "L"
- *`long()`* built-in function will coerce integer into a long type

float

- Represented as Java double type
- Numeric literals with a decimal point are considered floats
  - If literal ends with a decimal point, but no numerals following, it is a float
- `float()` built-in coerces an integer into a float type

Jython

complex

- Complex numbers represented by pair of Java doubles
  - real-part + imaginary-part
- Literals written as numeral followed by "j" or "J"
- The *`complex()`* built-in function creates a complex type from two numbers
  - The *`real()`* built-in function returns the real-part of the complex type
  - The *`imag()`* built-in function returns the imaginary part of the complex type

Arithmetic conversion

- If either argument is a complex number, the other is converted to complex
- otherwise, if either argument is a floating point number, the other is converted to floating point;
- otherwise, if either argument is a long integer, the other is converted to long integer;
- otherwise, both must be plain integers and no conversion is necessary.

Numeric functions

Function	Description
abs(number)	Return absolute value of number
cmp(obj1, obj2)	Returns 0 if obj1 == obj2 1 if obj1 > obj2 -1 if obj1 < obj2
coerce(num1, num2)	Return a tuple that represents the two args coerced to the same numerical type - the more precise of the two
divmod(x, y)	Return a tuple of the integer and remainder (x / y, x % y). The args (x and y) do not have to be integers.
hex(arg)	Returns a string of the hexadecimal value of the argument
oct(arg)	Returns a string of the octal value of the argument.
pow(base, power, [mod])	Return base ** power. Ig *mod* is present, returns (base ** power) % mod

math module

- Optimized math functions defined by the C standard
  - Cannot be used with complex numbers
    - Use the cmath module

https://docs.python.org/2.7/library/math.html

random module

- Pseudo-random number generators
  - Integers - uniform selection from a range
  - Sequences - uniform selection of a random element
  - Real - functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, the von Mises distribution is available.
- Not to be used for security purposes

https://docs.python.org/2/library/random.html

Other numeric types

Decimal

- Module provides support for decimal floating point arithmetic
  - Three concepts
    - Decimal number
    - Arithmetic context
    - Signals

Decimal number

- Immutable
- Contains
  - Sign
  - Coefficient digits (trailing 0 not truncated)
  - Exponent digits
- Special values (pos & negative)
  - Infinity
  - NaN
  - 0

Arithmetic context

- Environment to specify
  - Precision
  - Rounding rules
  - Limits on exponents
  - Flags to indicate results of operations
  - Trap enablers to handle exceptions

signals

- Exceptions encountered during computation

Module: fractions

- Support for rational number arithmetic

https://docs.python.org/2/library/fractions.html

Jython

Collection types

General types of collections

- Sequences
- Mappings
- Sets

Sequences

Sequences

- Strings
- Lists
- Tuples

strings

Characteristics

- Sequence of characters
- Immutable
- Create literals
  - Within single quote
  - Double quote
  - Triple quote
- Single (‘) and double (“) quotes are the same
- Escape quote using “\” character
  - Special escape

Characteristics

- Many string functions modify strings (e.g. replace()
  - Does not change the string - creates a copy

Common string functions

Function	Description
str.count('c')	Count occurrences of 'c' str
str.find('c')	Get position of 'c' in str
str.lower(), str.upper()	Change case of str to lowercase or uppercase
str.replace('a', 'b')	Replace all occurrences of 'a' with 'b' in str
str.strip(), str.lstrip(), str.rstrip()	Remove leading and trailing white space from str
str.split()	Split string at delimiter

https://docs.python.org/2/library/string.html

string searching

- Built in functions
- Regular expressions - next module

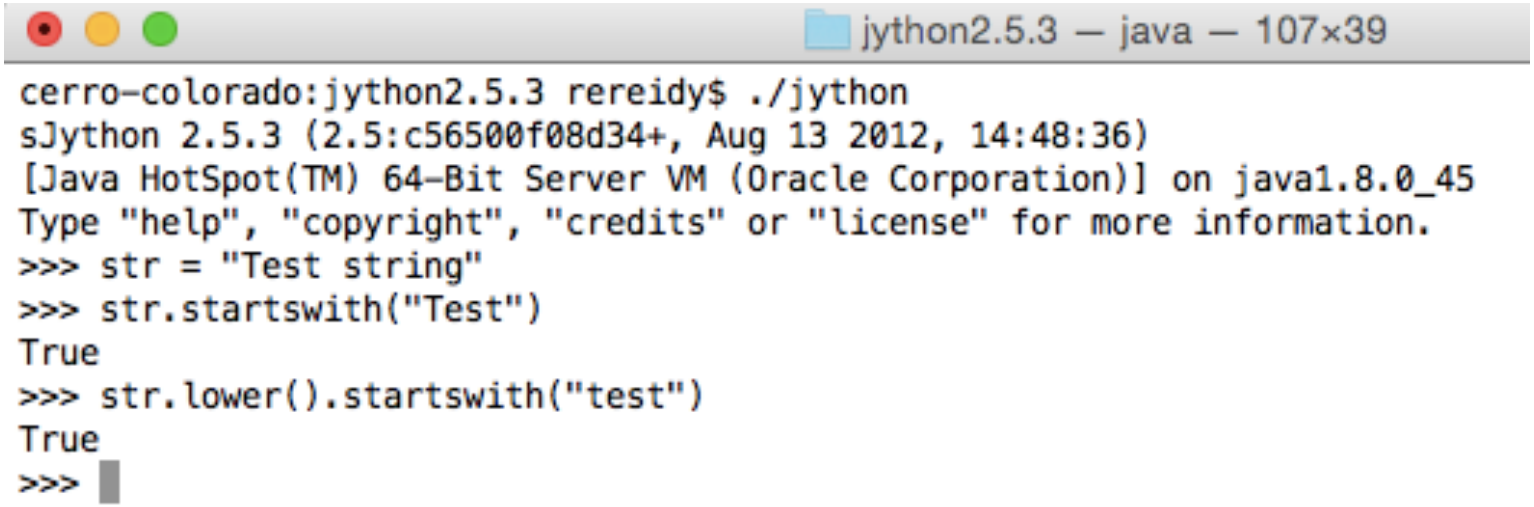
Built in functions

Common functions

- startswith()
- endswith()
- find()
- in
- index()
- partition()
- split()
- join()

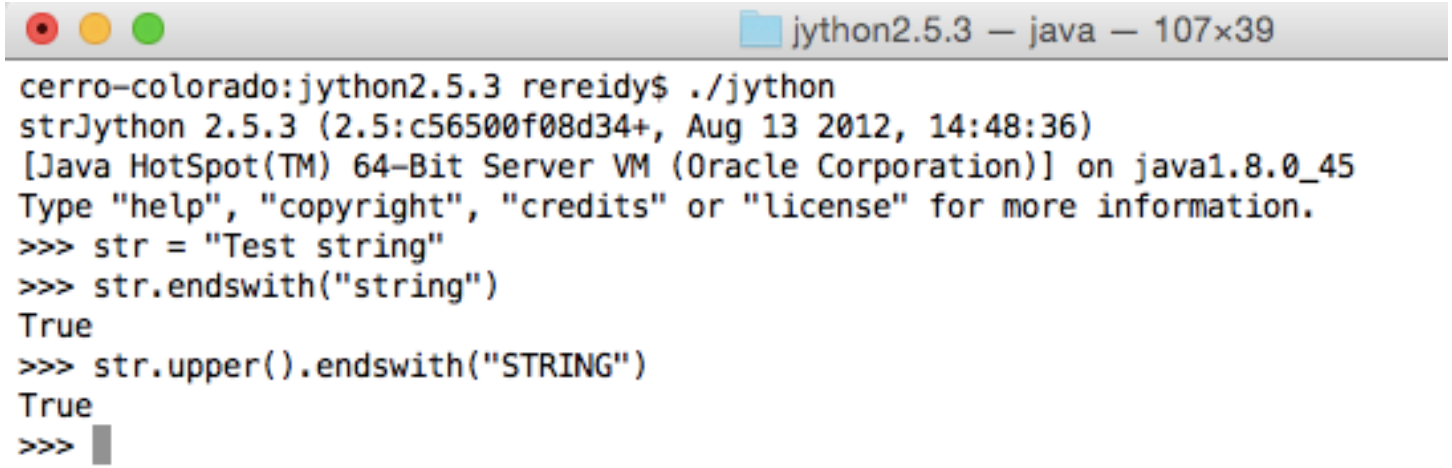
startswith()

String starts with substring



endswith()

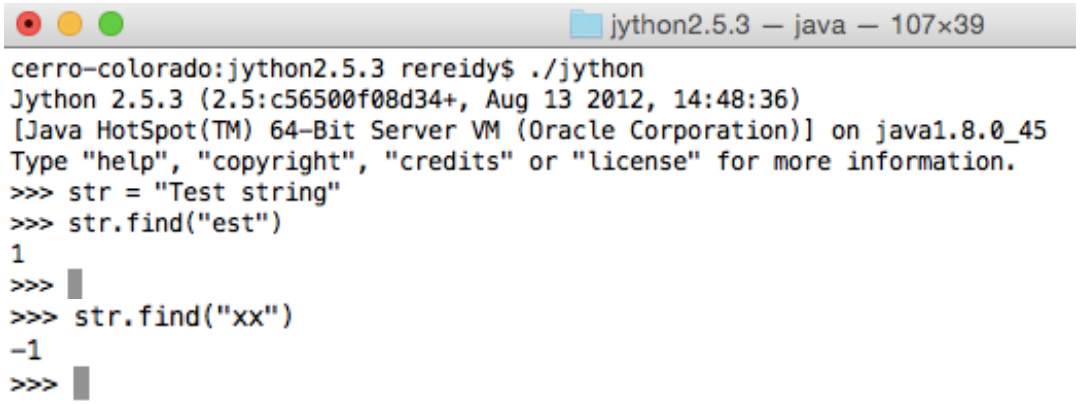
String ends with substring



find()

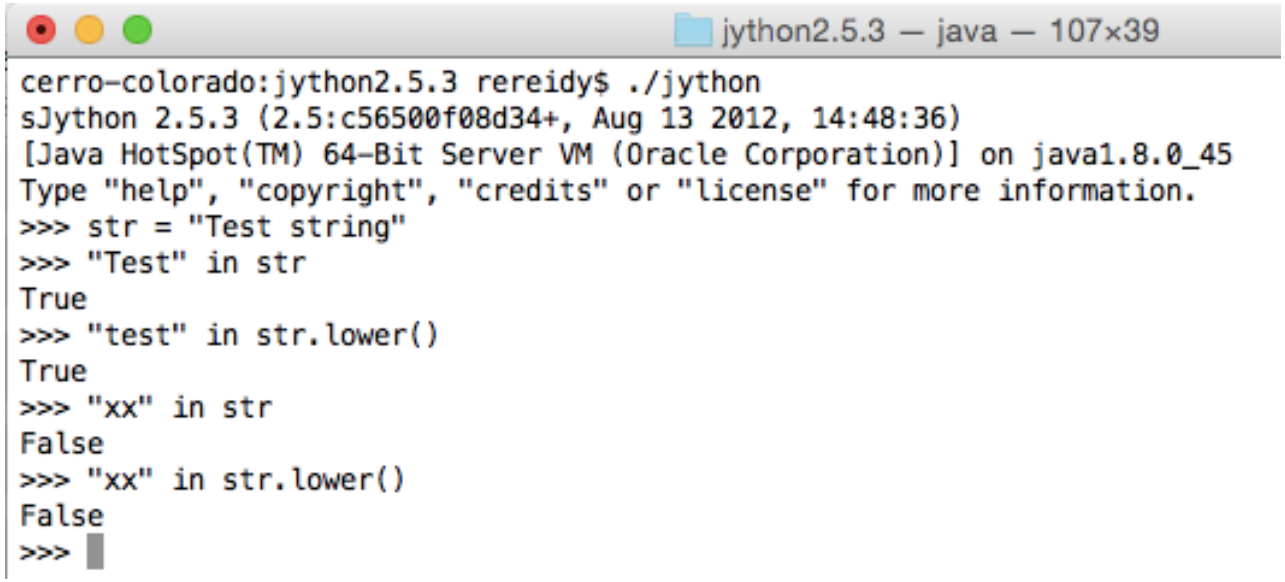
Find a substring in a string

- Returns lowest index in the string if found
  - -1 if not found
- Only use if you know the substring is in the string



in

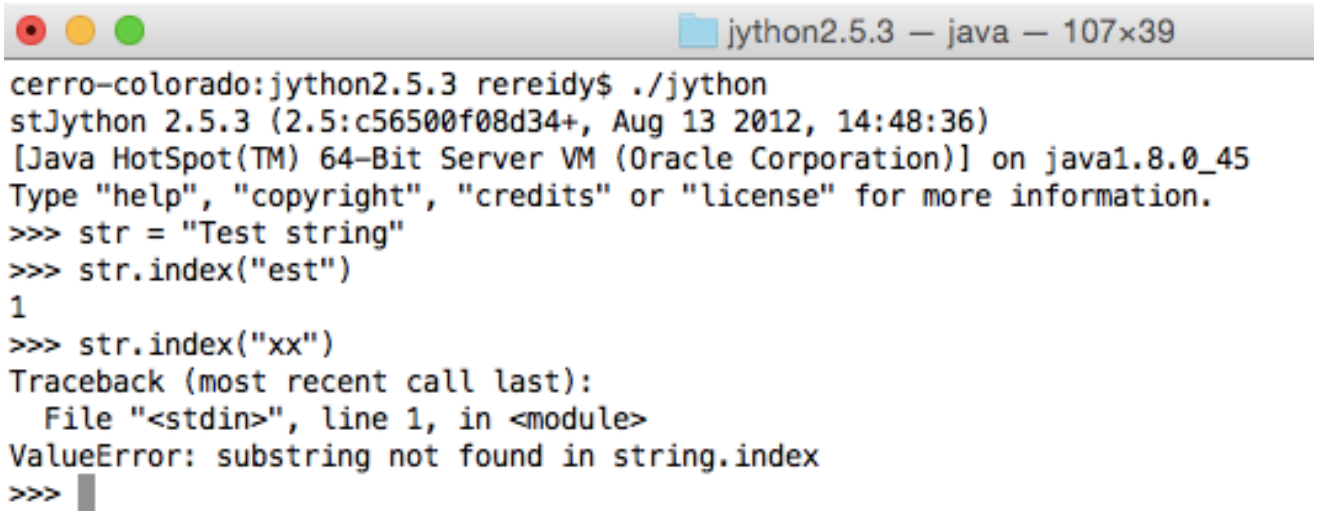
Substring in a string



index()

index()

- Returns the position in the string of the substring or throws an exception



partition()

Split at first occurrence of substring

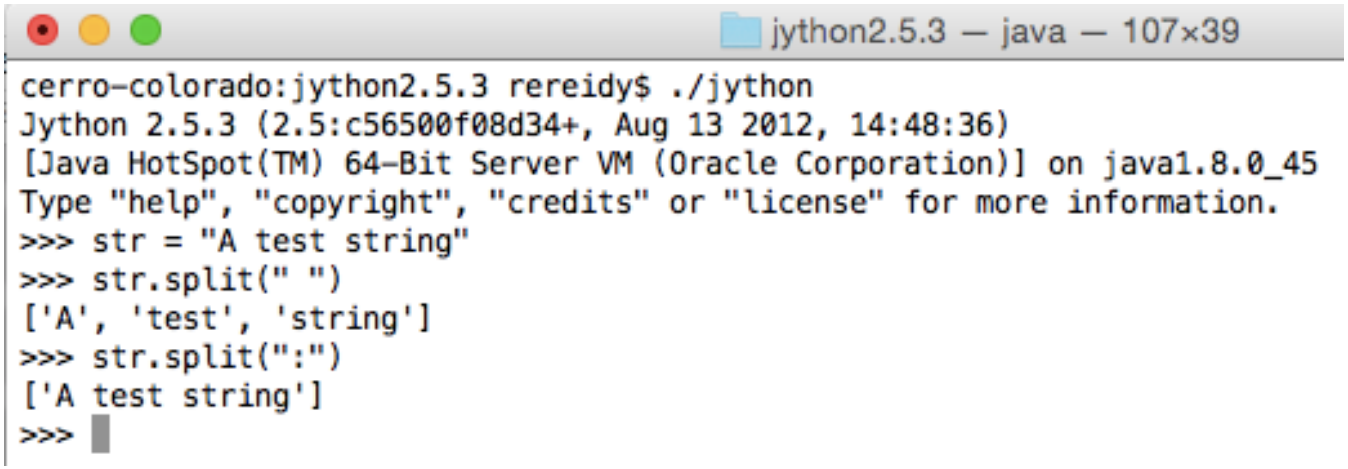
- Returns a three element tuple
  - Element[1] is the substring used to partition the string
  - If not found, return the string and 2 empty elements

```
jython2.5.3 — java — 107×39
cerro-colorado:jython2.5.3 rereidy$ ./jython
sJython 2.5.3 (2.5:c56500f08d34+, Aug 13 2012, 14:48:36)
[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)] on java1.8.0_45
Type "help", "copyright", "credits" or "license" for more information.
>>> str = "Test string"
>>> str.partition(" ")
('Test', ' ', 'string')
>>> type(str.partition(" "))
<type 'tuple'>
>>> str = "A test string"
>>> str.partition(" ")
('A', ' ', 'test string')
>>> str.partition(":")
('A test string', '', '')
>>>
```

split()

All substrings in string

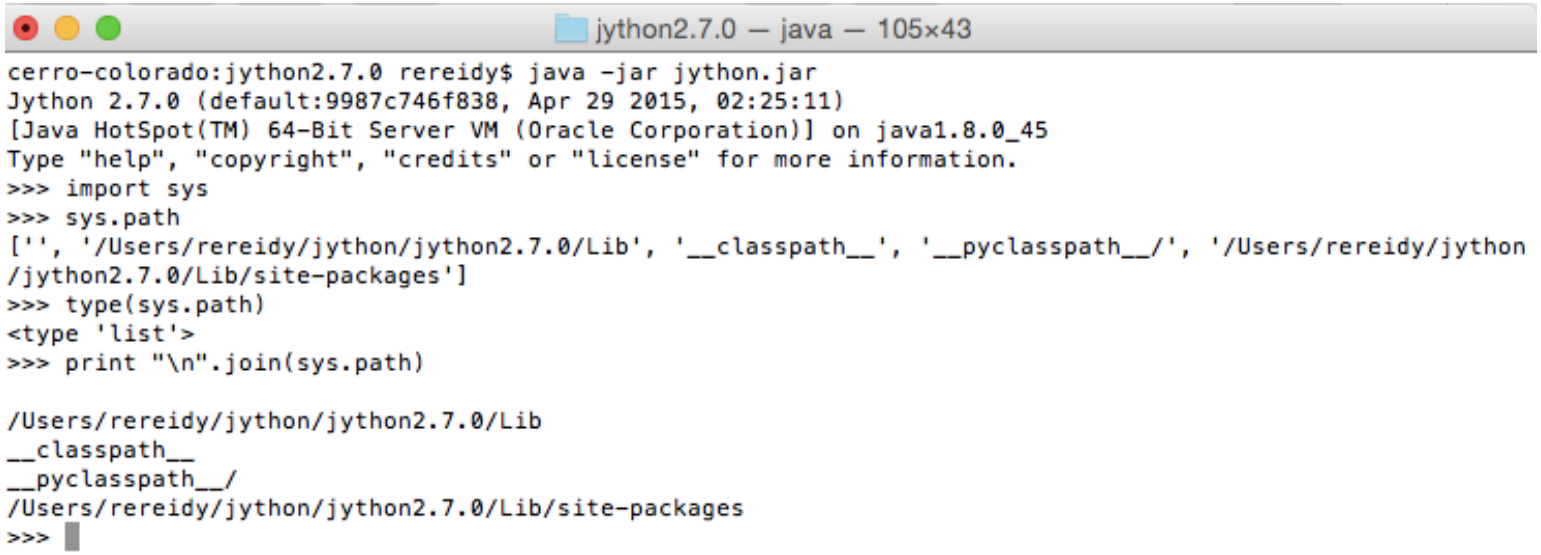
- Return a list of strings where the original string is delimited by the split character
  - If not found, return the original string as a list



join()

join()

- Flatten a list



Unicode string

- Unicode support for strings
- Create literal
  - Prefix with “u”

lists

Characteristics

- Ordered list of any data type
  - Numbers, mappings, lists, strings, int, float, long, etc.
- Mutable

Common list operations

Operation	Description
L = []	Empty list
L = [0, 1, 2]	Three items (indexes 0, 1, 2)
L = ['abc', ['def']]	Nexted list
L = list('test')	Make a list it items
L[i]	Index
L1 + L2 L * n	Concatenate
3 in L	Membership
for x in L:	Iteration

Common list methods

Method	Description
L.append() L.extend() L.insert()	Growing
L.index() L.count()	Searching
L.sort() L.reverse()	Sorting, reversing
del L[i] del L[i:j] L.pop() L.remove()	Shrinking
L[i:j] = [] L[i] = 1 L[i:j] = [4,5,6]	Index assignment, slice assignment

Tuple

Characteristics

- Similar to list
- Immutable - cannot be changed

Mappings

Dictionaries

- The only mapping type is a dictionary
- Also known as "associative array" or "hash"
  - key/value pairs
  - Random order
- Accessed by key
- Can grow in place

Common operations

Operation	Description
{}	Empty dictionary
dict(first='Ron', last='Reagan')	Alternate constructor
{'foo': 1, 'bar': 2}	Creat a two item dictionary
{'sandwich' : {'ham' : 1, 'cheese' : 2}}	Nesting
Ron' n D	Membership
D['value']	Access by key
D.values()	Method to return list of values
D.items()	Return keys and values

Common operations$_2$

Operation	Description
D.copy()	Copy dictionary
D.update(D_2)	Merge 2nd dictionary into first
D.pop(key)	Remove key from dictionary
len(D)	Number of stored entries
D[key] = value	Add/change key
del D[key]	Delete entries by key
list(D.keys())	Dictionary view (Python 2.6 and up)
{ x: x ** 2 for x in range(4)}	Dictionary comprehension

Sets

Characteristics

- Unordered collection of unique and immutable objects
  - Cannot embed lists, dictionaries, or other sets
- Similar to dictionaries with no values (keys only)
- Basis in mathematics
- Useful for
  - Filter duplicates from other collections
  - Tracking location or processing details
  - Great at dealing with large sets of data
- Create using the set() function

The future

- Python 2.6 introduces set comprehension (similar to list and dictionary comprehension)

Operations I

Operation	Description
set_1 in set_2	Membership
set_1 - set_2	Difference
set_1 | set_2	Union
set_1 & set_2	Intersection
set_1 ^ set_2	Symmetric difference (XOR)
set_1 > set_2 set_1 < set_2	Superset Subset

Operations$_2$

Method	Description
set$_1$.add(*seq*)	Insert one item
set$_1$.update(set(['a', 'b'])	Merge - in place union
set(set$_1$).intersection(set(set$_2$))	Intersection of sets
set$_1$.remove('a')	Delete an item from the set

collections module

Methods

- defaultdict
- deque
- namedtuple
  - More in later versions of Python (and Jython)

defaultdict

collections.defaultdict

- Part of the standard library
- Only available in Jython 2.7

deque

collections.deque

- Part of the standard library
- Only in Jython 2.7
- Double ended queue
  - Add and remove from either end of queue

namedtuple

- Part of the standard library
- Only in Jython 2.7
- Access fields of tuple by name using '.' notation
- Fields are invalid
  - Repeated names
  - Python keywords

Jython

Future methods

Counter

Counter

- Purpose
  - Convenient and rapid tallies
  - Subclass of dict for counting hashable objects
  - Can be accessed as a sequence or dictionary
  - Sorted in key order
  - Access by unknown key does not throw KeyError - returns 0

Fractions

Future type - Python 2.6 and up

- Support for rational numbers
  - "Rational, infinite-precision, real numbers"
- Can be constructed from
  - A pair of integers
  - A rational number
  - A float type
  - Decimal type
  - String

Uses

- gcd() - calculate the greatest common divisor of two numbers
- Fraction() - fraction constructor

Comprehension

Jython

Characteristics

- Shorthand method of building lists, dictionaries, and sets
- Substitution for
  - lambda
  - map
  - filter
  - reduce
    - These will be discussed later

list comprehension

Jython

dictionary comprehension

set comprehension

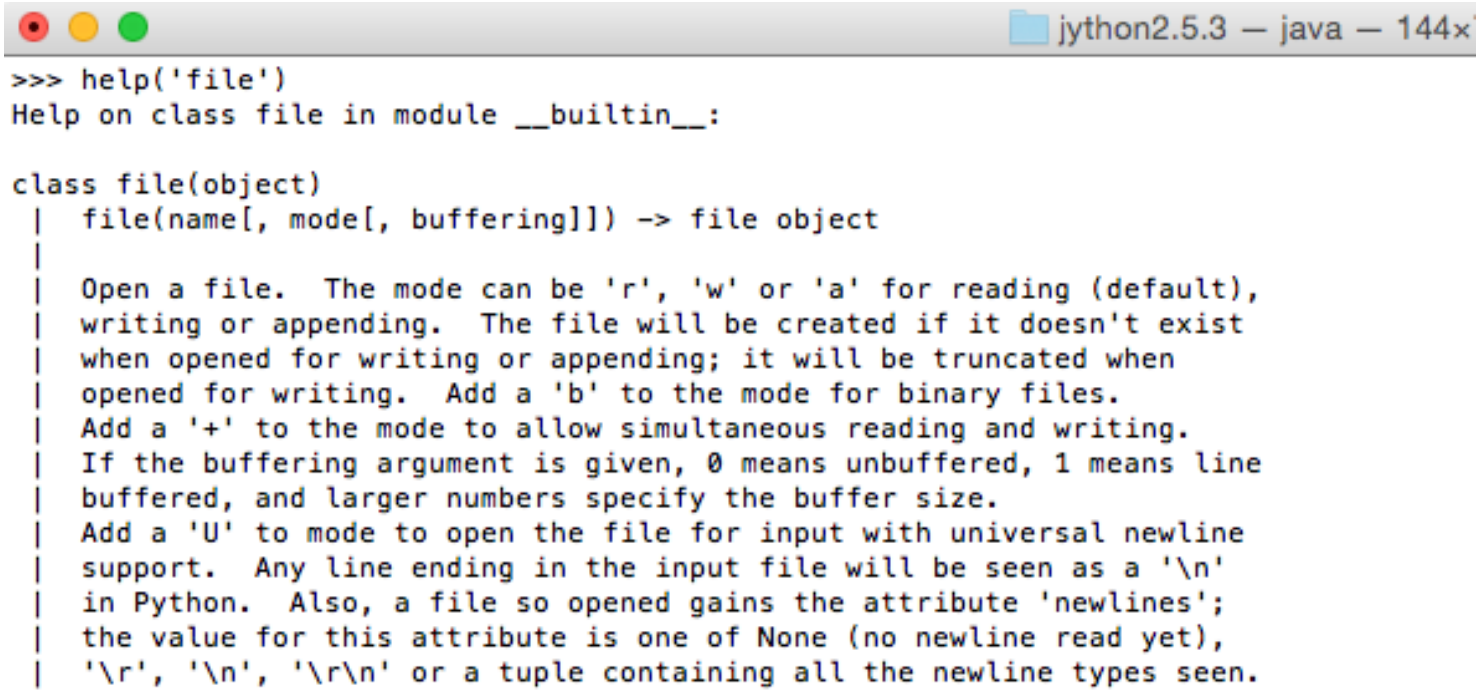
Jython

File operations

Common file operations

Operation	Decsription
close()	Close an open file
flush()	Flush internal file buffer
fileno()	Internal file desriptor
read([size])	Read a number of bytes (argument) or until EOF
readline([size])	Read an entire line until end of line. If size != 0, empty string returned on EOF
readlines()	Read until EOF
seek(offset [,whence])	Set the current file position
tell()	Return file current position
truncate([size])	Truncate file size (or to at most *size* bytes)
write(str)	Write the string to the file
writelines(seq)	Write a *seq* of lines to the file
closed	Current state of the file
mode	The I/O mode of the file
name	Name of the file opened

A file type is a built-in object



File operations

- The default mode for opening a file is “r”ead
- Open a file for “w”rite
- Open and “a”ppend

Working with binary files2

- Files can be opened in
  - Read - “rb”
  - Write - “wb”
  - Append - “ab”
- Can read text files in binary mode - rare
- When reading binary data (file type of “data”)
  - Use the struct module to pack() and unpack() the data in binary format

Standard Library

Reference library

- Built into the interpreter
  - Access to operations not part of the core language
    - Efficiency
    - OS primitives and system calls

Jython

Features

- Very extensive
- System access functionality
- Standardized solutions to common problems
  - Efficient algorithms, data structures
- Enhance portability

Built-ins

- Live in the built-in namespace (__builtins__)
- Always accessible (no need to import them)

Built in	URL
Functions	https://docs.python.org/2/library/functions.html
Constants	https://docs.python.org/2/library/constants.html
Types	https://docs.python.org/2/library/stdtypes.html
Exceptions	https://docs.python.org/2/library/exceptions.html

Module: os

- Portable methods for interacting with the operating system

Function	Description
name	Name of the operating system - returns "java" in Jython - use platform
environ, getenv(), putenv()	Mapping of environment string, get/put environment var
chdir()	Change directory
getcwd()	Get current working directory
strerror(code)	Print system error message that corresponds to the error code
listdir(path)	List of entries in the directory - arbitrary order - does not include'.' or '..'

https://docs.python.org/2/library/os.html

Module: sys

- Functions and variables that provide strong integration with the system

Function	Description
argv	Command line arguments passed to Python script
builtin_module_names	Tuple of all modules compiled into the interpreter
exc_info(), exc_clear()	Get exception information, clear exception
getsizeof()	Get size of an object - does not work in Jython - use java.lang.Instrumentation
path	List of directories searched during import
platform	Server platform identifier
ps1, ps2	Prompt strings (primary and secondary) used with the interactive shell

https://docs.python.org/2/library/sys.html

Module: shutils

- High-level operations on files and collections of files

Function	Description
copyfile(src, dst)	Copy file src to dest (no metadata)
copymode(src,dst)	Copy permission bits of src to dst
copytree(src,dst, …)	Recursive copy an entire directory tree
rmtree(path, …)	Delete an entire directory tree
move(src,dst)	Recursively move a file or directory

https://docs.python.org/2/library/shutil.html?highlight=shutils

Module: glob

- Pattern matching for paths (files and directories)
  - No tilde (~) expansion
  - Correct matching for “*”, “?”, and character ranges expressed with “[…]”

Exploring modules

dir() function

- Examine all functions implemented in a module
- All "special" methods and functionality

jython2.5.3 — java — 107×39

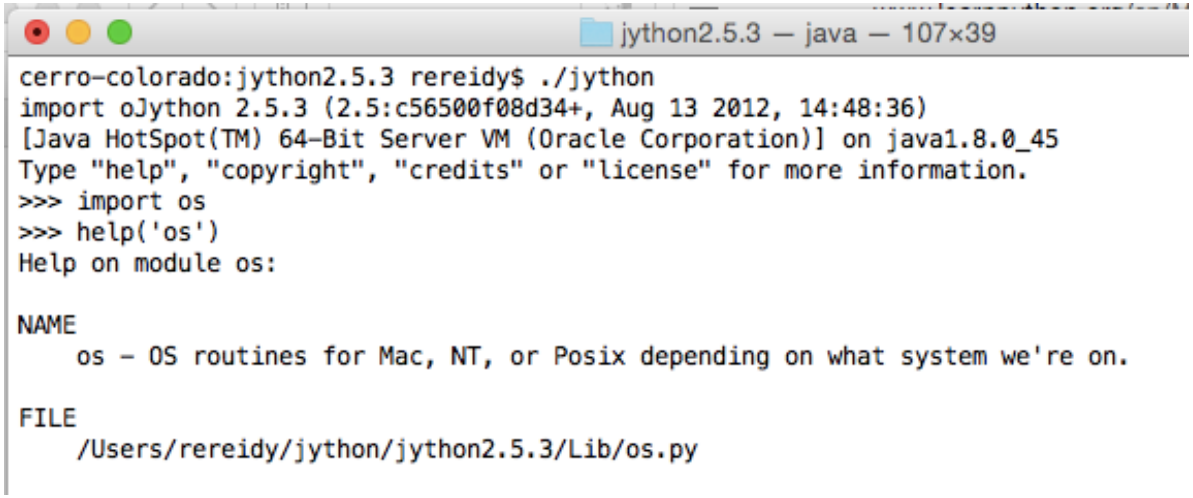
```
cerro-colorado:jython2.5.3 rereidy$ ./jython
Jython 2.5.3 (2.5:c56500f08d34+, Aug 13 2012, 14:48:36)
[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)] on java1.8.0_45
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> dir('os')
['__add__', '__class__', '__cmp__', '__contains__', '__delattr__', '__doc__', '__eq__', '__ge__', '__getatt
ribute__', '__getitem__', '__getnewargs__', '__getslice__', '__gt__', '__hash__', '__init__', '__le__', '__
len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__
rmul__', '__setattr__', '__str__', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expand
tabs', 'find', 'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'islower', 'isnumeric', 'isspace', 'i
stitle', 'isunicode', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rind
ex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', '
title', 'translate', 'upper', 'zfill']
>>>
```

https://docs.python.org/2/library/functions.html#dir

help() function

- Invokes the built-in help system documentation
  - Interactive shell only
    - Module, function, class, method, keyword

https://docs.python.org/2/library/functions.html#help



- You can also get help on specific methods

```
>>> help('os.chdir')
Help on reflectedfunction in os:

os.chdir = chdir(...)
    chdir(path)

    Change the current working directory to the specified path.

>>>
```

Jython

Demonstration of help() and dir()

Input and output

Streams

- Another word for files, directories
  - Many things in UNIX are represented as a stream of some kind
- Three built-in streams (in the sys module)
  - stdin - standard input - keyboard (file descriptor 0)
  - stdout - standard output - terminal (file descriptor 1)
  - stderr - standard error - terminal (but different channel than stdout) (file descriptor 2)

Console

Read from keyboard

- raw_input() - presents a prompt to the user (the optional arg), gets input from the user and returns the data input by the user in a string

Printing

- Use the print statement
  - Python 2.7 and up, print is a function - print()

```
from __future__ import print_function
```

- print always appends a newline
- print always prints to stdout

Other ways to print

- From the sys module
  `sys.stderr.write()`
  `sys.stdout.write()`
- Redirect print to stderr
  `print >> sys.stderr, …`
  - With the print function (print())
    `print("string", file=sys.stderr)`
    - This does not work in Jython 2.7 - does work in Python 2.7

Formatting

- C language like formatting
  - I have shown some examples already

```
print "this is %s - %d" %(var, i)
```

Format string	Description
%s %.ns %ns	String, left justify n characters; right justify n characters
%d	Integer
%r	Print internal representation of object
%f %n.mf	Floating point (default precision is) n numerals on left of decimal; m numerals on right
%x or %X	Hexidecimal

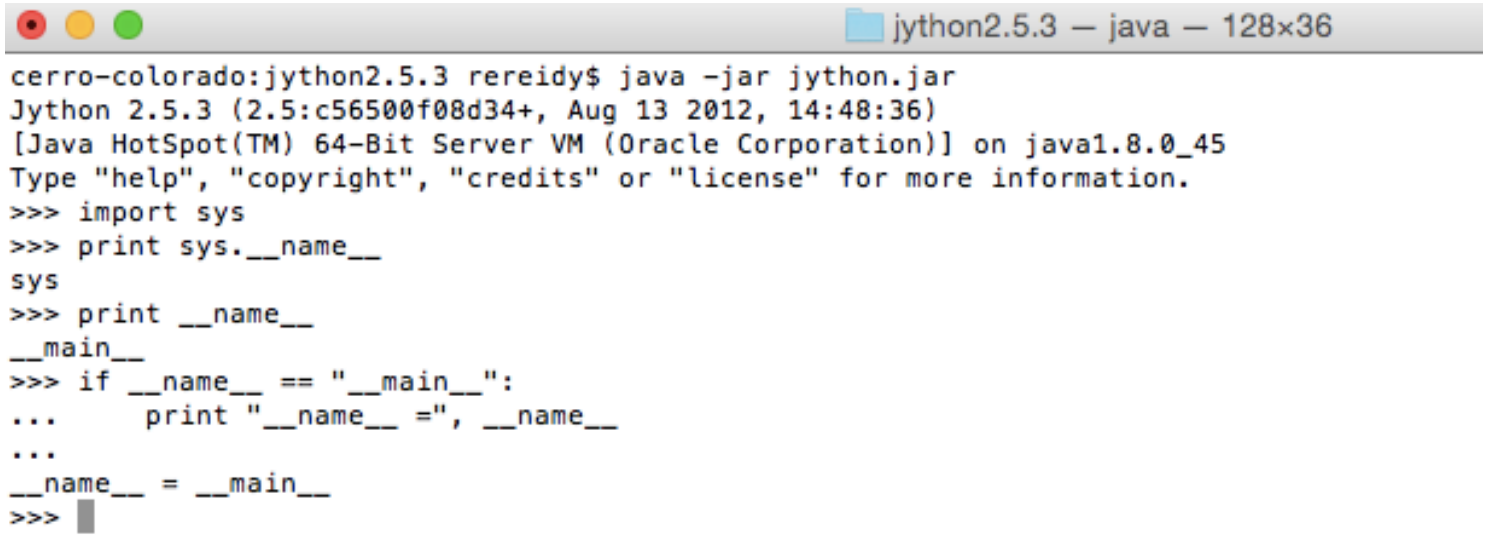
Module: pprint

- Print arbitrary data structures
  - Form that can be used as input to the interpreter

Special variables

Many special variables I

- *__name__* - name of the module file namespace
- *__main__* - name of module namespace when executing program



Many special variables$_2$

- "_" (single underscore) - variable used for short lived and/or ignored results from functions

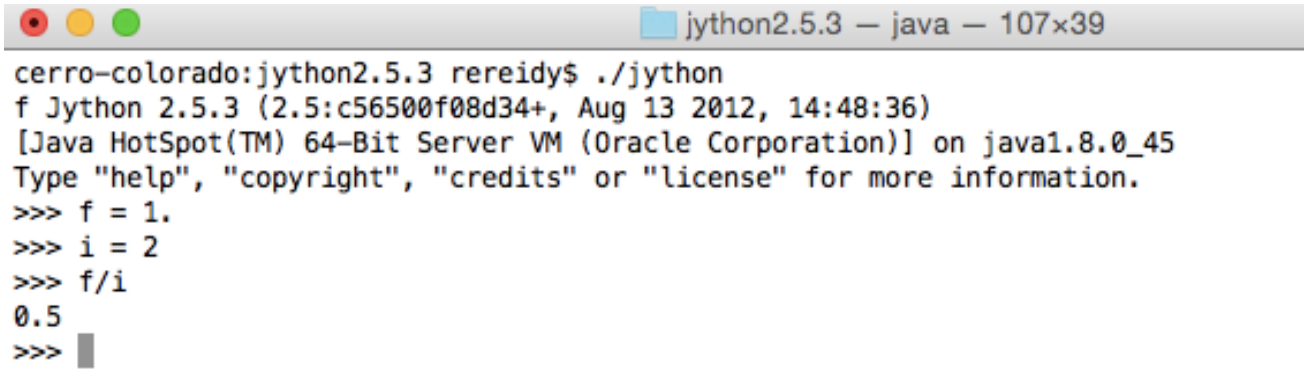
```
import subprocess

proc = subprocess.Popen(cmd, shell=True, stdin=subprocess.PIPE,
                                          stdout=subprocess.PIPE,
                                          stderr=subprocess.PIPE,
                                          close_fds=True
                        )
(_, cout, cerr) = (proc.stdin, proc.stdout, proc.stderr)
```

- Others specific to classes

Quiz

- Given f = 1. and i = 2
  1. What is the value of "f / i"?
  2. What is the answer if you coerce to a long (using long())?
     - Explain both answers




```
>>> long(f/i)
0L
>>>
```

Jython

2. What are the collection types?

1. List
2. Dictionary
3. String
4. Tuple
5. Set

1. How does Jython represent a long integer?

A. As an instant of java.math.BigInteger

Q & A

Exercises

Jython

1. Explore the numeric functions. Save your work.

Function
abs(number)
cmp(obj1, obj2)
coerce(num1, num2)
divmod(x, y)
hex(arg)
oct(arg)
pow(base, power, [mod])

Jython

2. Many of the string functions have “r” and “l” variations. Write code to explore the “r” variations of split() (rsplit()) and partition() (rpartition()).

3. Write a program to create a list and change it in place (do not create a temporary list)
4. Write a program to append items into a list
5. Sort the list
6. Sort the list in reverse order

Jython

9. Write a program to read a file on your computer and display the results with line numbers.

10. Write a program that uses the log10() functions from the decimal, math, and cmath modules. Calculate log10() values for 20 floating point values.

Notes:
1. The decimal package is only available in Jython 2.7. This part of the test can be skipped for Jython 2.5.

2. cmath.log10() returns a complex type, so use "%r" (not "%f") to print the results.

a. Use range(1, 20) and convert the results to floats. Use for loops and comprehension methods to generate the numbers.

b. Benchmark the three methods:

```
import time
start_tm = time.time()
…
elapsed_tm = time.time() - start_tm
```

11. Write a program to read the contents of the "README.txt" file in the "jython2.5.3" directory. Print only lines that start with the "*" character.