Administrivia

• Quickie Homework 10
  – Python

• Term Programming Project
  – Did you email me about your proposed project yet?
Online documentation

Python 2.7.10 documentation

Welcome! This is the documentation for Python 2.7.10, last updated Nov 12, 2015.

Parts of the documentation:

- What's new in Python 2.7?
- Extending and Embedding
  tutorial for C/C++ programmers
- Tutorial
  start here
- Python/C API
  reference for C/C++ programmers
- Library Reference
  keep this under your pillow
- Installing Python Modules
  information for installers & sys-admins
- Language Reference
  describes syntax and language elements
- Distributing Python Modules
  sharing modules with others
- Python Setup and Usage
  how to use Python on different platforms
- FAQs
  frequently asked questions (with answers!)
- Python HOWTOs
  in-depth documents on specific topics
Python

- Chapter 3: Computing with Numbers
- Last time:
  - Number data types: int, float
  - Matters for things like division: /
    - 9/5 produces different result from 9/5.0

### Basic:

<table>
<thead>
<tr>
<th>operator</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
</tr>
<tr>
<td>**</td>
<td>exponentiation</td>
</tr>
<tr>
<td>%</td>
<td>remainder</td>
</tr>
<tr>
<td>abs()</td>
<td>absolute value</td>
</tr>
</tbody>
</table>

### Library: module

```python
import math  # Makes the math library available.
```

Table 3.1: Python built-in numeric operations.
Python

- `math.sqrt()`

complex numbers:
import cmath

<table>
<thead>
<tr>
<th>Python</th>
<th>Mathematics</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pi</code></td>
<td>$\pi$</td>
<td>An approximation of pi.</td>
</tr>
<tr>
<td><code>e</code></td>
<td>$e$</td>
<td>An approximation of $e$.</td>
</tr>
<tr>
<td><code>sin(x)</code></td>
<td>$\sin x$</td>
<td>The sine of x.</td>
</tr>
<tr>
<td><code>cos(x)</code></td>
<td>$\cos x$</td>
<td>The cosine of x.</td>
</tr>
<tr>
<td><code>tan(x)</code></td>
<td>$\tan x$</td>
<td>The tangent of x.</td>
</tr>
<tr>
<td><code>asin(x)</code></td>
<td>$\arcsin x$</td>
<td>The inverse of sine x.</td>
</tr>
<tr>
<td><code>acos(x)</code></td>
<td>$\arccos x$</td>
<td>The inverse of cosine x.</td>
</tr>
<tr>
<td><code>atan(x)</code></td>
<td>$\arctan x$</td>
<td>The inverse of tangent x.</td>
</tr>
<tr>
<td><code>log(x)</code></td>
<td>$\ln x$</td>
<td>The natural (base $e$) logarithm of x</td>
</tr>
<tr>
<td><code>log10(x)</code></td>
<td>$\log_{10} x$</td>
<td>The common (base 10) logarithm of x.</td>
</tr>
<tr>
<td><code>exp(x)</code></td>
<td>$e^x$</td>
<td>The exponential of x.</td>
</tr>
<tr>
<td><code>ceil(x)</code></td>
<td>$[x]$</td>
<td>The smallest whole number $\geq x$.</td>
</tr>
<tr>
<td><code>floor(x)</code></td>
<td>$[x]$</td>
<td>The largest whole number $\leq x$.</td>
</tr>
</tbody>
</table>

Table 3.2: Some math library functions.
Python

- `range(n)`: produces a list (sequence)  
  - `range(n)`: \([0,1,..,n-1]\)  
  - `range(start, n)`: \([\text{start, start+1},..,n-1]\)  
  - `range(start, n, step)`: \([\text{start, start+step},...,\text{last}]\)  
  - `range(n, stop, -step)`: counts down  
    
    ```python
    >>> for i in [0,1,2,3]:
    print i
    0
    1
    2
    3
    ```
    ```python
    >>> for odd in [1, 3, 5, 7, 9]:
    print odd * odd
    1
    9
    25
    49
    81
    ```
# futval.py

This program calculates the future value of a 10 year investment.

Enter the initial principal: 1000
Enter the annualized interest rate: .035
The value in 10 years is: 1410.59876062

```python
def main():
    print "This program calculates the future value of a 10 year investment."
    print "Enter the initial principal: ",
    principal = input("Enter the initial principal: ")
    apr = input("Enter the annualized interest rate: ")
    for i in range(10):
        principal = principal * (1 + apr)
    print "The value in 10 years is: ", principal

main()
```

How about to 2 decimal places?
Python

• page 60: see factorial.py
  – Python automatically converts to type long int from int (32 bit 2's complement), (page 64–65, section 3.5)

```python
>>> 2**31-1
2147483647
```

  – explicit type coercion:
    1. float()
    2. int()
    3. long()
    4. complex(r,i)
    5. complex(string)
Homework 10

• Pick any three of the programming exercises from page 72-75
  – Program them up in python
  – Submit your code and sample runs all in one pdf file
  – Due this Saturday by midnight