PyWeather

API Documentation

March 27, 2006

Contents

1 Package weather

a collection of meteorological conversion functions, station readers, and publishing modules

1.1 Modules

1.2 Variables

Name	Description
__version__	**Value:** `'0.7.0'`

2 Package weather.services

2.1 Modules

- **test**: Demonstrates how to call the wunderground publisher.
  (Section 3, p. 5)
- **wunderground**: WUnderground.com Publisher
  (Section 4, p. 6)

3 Module weather.services.test

Demonstrates how to call the wunderground publisher. See the wunderground module for more information.

3.1 Functions

main()

usage()

wunderground()

3.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'\n  python $0\n'`

4 Module weather.services.wunderground

WUnderground.com Publisher

Abstract: The class contained within this module allows python programs to publish weather conditions to the wunderground.com servers. That is, this class encapsulates the wire protocol wunderground.com supports and allows application developers to insulate themselves against changes in the wunderground.com wire protocol.

If the rtfreq parameter is passed to the Publisher constructor, posting of the current conditions will go to the "real time updater" service provided by wunderground.com. The rtfreq optional parameter passed to the constructor is a float that represents the number of seconds between observations.

Usage: >>> publisher = Publisher() >>> publisher.set(30.12, 28.52, 53.0, 44.6, 0.0, time.gmtime(), 0, 0, 0) >>> response = publisher.publish('MyUserName', 'MyPassword') >>> print '%s: %s' % (response.status, response.reason)

Notes on arguments to Publisher.set(): <float> pressure: in inches of Hg <float> dewpt: in Fahrenheit <float> humidity: between 0.0 and 100.0 inclusive <float> tempf: in Fahrenheit <time tuple> dateutc: 9 value time tuple in UTC (e.g. time.gmtime()) <float> windgust: in mph <float> winddir: in degrees, between 0.0 and 100.0 <string> clouds: unknown at this time (email me if you know!) <string> weather: unknown at this time (email me if you know!)

Developers Notes: It appears that even if you provide an invalid username and password, a status of 200, and a reason of "OK" is returned.

Author: Christopher Blunck (chris@wxnet.org) Date: 2006-03-27

4.1 Class Publisher

Publishes weather data to the wunderground.com servers. See module documentation for additional information and usage idioms.

4.1.1 Methods

__init__(*self*, *rtfreq*=`None`)

publish(*self*, *username*, *password*, *debug*=`False`)

set(*self*, *pressure*, *dewpoint*, *humidity*, *tempf*, *rainin*, *dateutc*, *windgust*, *windspeed*, *winddir*, *clouds*=`'NA'`, *weather*=`'NA'`)

4.1.2 Class Variables

Name	Description
REALTIME_SERVER	**Value:** `'rtupdate.wunderground.com'`
STD_SERVER	**Value:** `'weatherstation.wunderground.com'`
URI	**Value:** `'/weatherstation/updateweatherstation.php'`

5 Package weather.stations

5.1 Modules

- **davis** *(Section 6, p. 8)*
- **validate** *(Section 7, p. 9)*

6 Module weather.stations.davis

6.1 Class VantagePro

A class capable of reading raw (binary) weather data from a vantage pro console and parsing it into usable scalar (integer/long/real) values.

The data read from the console is in binary format, and must be converted to hex using a least-ordered nybble strategy. The hex is in fixed-length format, and values can be extracted using an offset and length strategy (e.g. outside humidity starts at position 56 and is 2 bytes long).

6.1.1 Methods

__init__(*self*, *device*, *start*='c4f4f4')

get_field(*self*, *start*, *len*)

returns the value in the field specified by the starting offset and of the length provided. this value is computed by first reversing the hex data, followed by conversion of the hex data to a long.

parse(*self*)

read and parse a set of data read from the console. after the data is parsed it is available in the fields variable.

7 Module weather.stations.validate

7.1 Class Validator

7.1.1 Methods

__init__(*self*, *fields*)

get_value(*self*, *field*, *default*)

validate(*self*)

8 Module weather.test_all

Module weather.test_all

8.1 Functions

main()

9 Package weather.units

9.1 Modules

- **astro** *(Section 10, p. 12)*
- **precip**: precipitation related conversions
  (Section 11, p. 13)
- **pressure**: pressure related conversion functions
  (Section 12, p. 14)
- **temp**: temperature related conversionfunctions
  (Section 13, p. 18)
- **test_pressure**: Unit tests the pressure module.
  (Section 14, p. 20)
- **test_temp**: Unit tests the temp module.
  (Section 15, p. 23)
- **test_wind**: Unit tests the pressure module.
  (Section 16, p. 25)
- **wind**: wind related conversion functions
  (Section 17, p. 27)

10 Module weather.units.astro

10.1 Functions

daylight(*lat*, *long*, *tz*, *day*, *month*, *year*)

degrees_to_radians(*degrees*)

radians_to_degrees(*radians*)

10.2 Variables

Name	Description
zenith	**Value:** `-0.014540000000000001`

precipitation related conversions

11.1 Variables

Name	Description
__author__	**Value:** 'Christopher Blunck'
__email__	**Value:** 'chris@wxnet.org'
__revision__	**Value:** '$Revision: 1.6 $'
__usage__	**Value:** ''

12 Module weather.units.pressure

pressure related conversion functions

12.1 Functions

atm_to_in32(*atm*)

Atmospheres (atm) to inches of mercury @32F (inHg32)

atm_to_in60(*atm*)

Atmospheres (atm) to inches of mercury @60F (inHg60)

atm_to_lb_sqin(*atm*)

Atmospheres (atm) to pounds/square inch (lb/in**2)

atm_to_mb(*atm*)

Atmospheres (atm) to millibars (mb)

atm_to_pa(*atm*)

Atmospheres (atm) to pascals (Pa)

hpa_to_inches(*hpa*)

hpa_to_mb(*hpa*)

Hectopascals (hPa) to millibars (mb)

in32_to_atm(*inches*)

Inches of mercury @32F (inHg32) to millibars (mb)

in32_to_lbs(*inches*)

Inches of mercury @32F (inHg32) to pounds/square inch (lb/in**2)

in32_to_mb(*inches*)

Inches of mercury @32F (inHg32) to millibars (mb)

in60_to_atm(*inches*)

Inches of mercury @60F (inHg60) to millibars (mb)

in60_to_lbs(*inches*)

Inches of mercury @60F (inHg60) to pounds/square inch (lb/in**2)

in60_to_mb(*inches*)

Inches of mercury @60F (inHg60) to atmospheres (atm)

kpa_to_mb(*hpa*)

Kilopascals (kPa) to millibars (mb)

lb_sqft_to_mb(*lbs*)

Pounds/square foot (lb/ft**2) to millibars (mb)

lb_sqin_to_atm(*lbs*)

Pounds/square inch (lb/in**2) to atmospheres (atm)

lb_sqin_to_mb(*lbs*)

Pounds/square inch (lb/in**2) to millibars (mb)

lb_sqin_to_mm32(*lbs*)

Pounds/square inch (lb/in**2) to inches of mercury @32F (inHg32)

lb_sqin_to_mm60(*lbs*)

Pounds/square inch (lb/in**2) to inches of mercury @60F (inHg60)

mb_to_atm(*mb*)

Millibars (mb) to atmospheres (atm)

mb_to_hpa(*mb*)

Millibars (mb) to hectopascals (hPa)

mb_to_in32(*mb*)

Millibars (mb) to inches of mercury @32F (inHg60)

mb_to_in60(*mb*)

Millibars (mb) to inches of mercury @60F (inHg60)

mb_to_kpa(*mb*)

Millibars (mb) to kilopascals (kPa)

mb_to_lb_sqft(*mb*)

Millibars (mb) to pounds/square foot (lb/ft**2)

mb_to_lb_sqin(*mb*)

Millibars (mb) to pounds/square inch (lb/in**2)

mb_to_mm32(*mb*)

Millibars (mb) to millimeters of mercury @32F (mmHg)

mb_to_mm60(*mb*)

Millibars (mb) to millimeters of mercury @60F (mmHg)

mb_to_n_sqm(*mb*)

Millibars (mb) to newtons/square meter (N/m**2)

mb_to_pa(*mb*)

Millibars (mb) to pascals (Pa)

mm32_to_mb(*mm32*)

Millimeters of mercury @32F (mmHg) to millibars (mb)

mm60_to_mb(*mm60*)

Millimeters of mercury @60F (mmHg) to millibars (mb)

n_sqm_to_mb(*nsqm*)

Newtons/square meter (N/m**2) to millibars (mb)

pa_to_atm(*pa*)

Pascals (Pa) to atmospheres (atm)

pa_to_mb(*pa*)

Pascals (Pa) to millibars (mb)

12.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`

continued on next page

Name	Description
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'this module should not be run via the command line'`

13 Module weather.units.temp

temperature related conversionfunctions

13.1 Functions

calc_dewpoint(*temp*, *hum*)

calculates the dewpoint via the formula from weatherwise.org return the dewpoint in degrees F.

calc_heat_index(*temp*, *hum*)

calculates the heat index based upon temperature (in F) and humidity.
http://www.srh.noaa.gov/bmx/tables/heat_index.html
returns the heat index in degrees F.

calc_humidity(*temp*, *dewpoint*)

calculates the humidity via the formula from weatherwise.org return the relative humidity

calc_wind_chill(*t*, *windspeed*, *windspeed10min*=`None`)

calculates the wind chill value based upon the temperature (F) and wind.
returns the wind chill in degrees F.

celsius_to_fahrenheit(*c*)

Degrees Celsius (C) to degrees Fahrenheit (F)

celsius_to_kelvin(*c*)

Degrees Celsius (C) to degrees Kelvin (K)

celsius_to_rankine(*c*)

Degrees Celsius (C) to degrees Rankine (R)

fahrenheit_to_celsius(*f*)

Degrees Fahrenheit (F) to degrees Celsius (C)

fahrenheit_to_kelvin(*f*)

Degrees Fahrenheit (F) to degrees Kelvin (K)

fahrenheit_to_rankine(*f*)

Degrees Fahrenheit (F) to degrees Rankine (R)

kelvin_to_celsius(k)

Degrees Kelvin (K) to degrees Celsius (C)

kelvin_to_fahrenheit(k)

Degrees Kelvin (K) to degrees Fahrenheit (F)

kelvin_to_rankine(k)

Degrees Kelvin (K) to degrees Rankine (R)

rankine_to_celsius(r)

Degrees Rankine (R) to degrees Celsius (C)

rankine_to_fahrenheit(r)

Degrees Rankine (R) to degrees Fahrenheit (F)

rankine_to_kelvin(r)

Degrees Rankine (R) to degrees Kelvin (K)

13.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'this module should not be run via the command line'`

14 Module weather.units.test_pressure

Unit tests the pressure module.

14.1 Functions

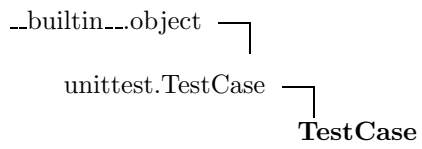
main()

usage()

14.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'\n  python $0\n'`

14.3 Class TestCase



14.3.1 Methods

setUp(*self*)

Hook method for setting up the test fixture before exercising it.

Overrides: unittest.TestCase.setUp extit(inherited documentation)

tearDown(*self*)

Hook method for deconstructing the test fixture after testing it.

Overrides: unittest.TestCase.tearDown extit(inherited documentation)

test__atm_to_in32(*self*)

test__atm_to_in60(*self*)

test__atm_to_lb_sqin(*self*)

test__atm_to_mb(*self*)

test__atm_to_pa(*self*)

test__hpa_to_mb(*self*)

test__in32_to_atm(*self*)

test__in32_to_lbs(*self*)

test__in32_to_mb(*self*)

test__in60_to_atm(*self*)

test__in60_to_lbs(*self*)

test__in60_to_mb(*self*)

test__kpa_to_mb(*self*)

test__lb_sqft_to_mb(*self*)

test__lb_sqin_to_atm(*self*)

test__lb_sqin_to_mb(*self*)

test__lb_sqin_to_mm32(*self*)

test__lb_sqin_to_mm60(*self*)

test__mb_to_atm(*self*)

test__mb_to_hpa(*self*)

test__mb_to_in32(*self*)

test__mb_to_in60(*self*)

test__mb_to_kpa(*self*)

test__mb_to_lb_sqft(*self*)

test__mb_to_lb_sqin(*self*)

test__mb_to_mm32(*self*)

test__mb_to_mm60(*self*)

test__mb_to_n_sqm(*self*)

test__mb_to_pa(*self*)

test__mm32_to_mb(*self*)

test__mm60_to_mb(*self*)

test__n_sqm_to_mb(*self*)

test__pa_to_atm(*self*)

test__pa_to_mb(*self*)

Inherited from object: __delattr__, __getattribute__, __hash__, __reduce__, __reduce_ex__, __setattr__
Inherited from type: __new__
Inherited from TestCase: __init__, __call__, __repr__, __str__, assert_, assertAlmostEqual, assertAlmostEquals, assertEqual, assertEquals, assertNotAlmostEqual, assertNotAlmostEquals, assertNotEqual, assertNotEquals, assertRaises, countTestCases, debug, defaultTestResult, fail, failIf, failIfAlmostEqual, failIfEqual, failUnless, failUnlessAlmostEqual, failUnlessEqual, failUnlessRaises, id, run, shortDescription

15 Module weather.units.test_temp

Unit tests the temp module.

15.1 Functions

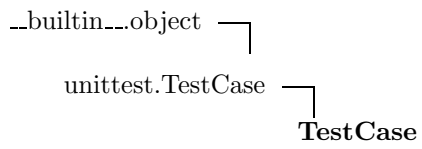
main()

usage()

15.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'\n  python $0\n'`

15.3 Class TestCase



15.3.1 Methods

setUp(*self*)

Hook method for setting up the test fixture before exercising it.

Overrides: unittest.TestCase.setUp extit(inherited documentation)

tearDown(*self*)

Hook method for deconstructing the test fixture after testing it.

Overrides: unittest.TestCase.tearDown extit(inherited documentation)

test__calc_heat_index(*self*)

test__calc_wind_chill(*self*)

test__celsius_to_fahrenheit(*self*)

test__celsius_to_kelvin(*self*)

test__celsius_to_rankine(*self*)

test__dewpoint(*self*)

test__fahrenheit_to_celsius(*self*)

test__fahrenheit_to_kelvin(*self*)

test__fahrenheit_to_rankine(*self*)

test__humidity(*self*)

test__kelvin_to_celsius(*self*)

test__kelvin_to_fahrenheit(*self*)

test__kelvin_to_rankine(*self*)

test__rankine_to_celsius(*self*)

test__rankine_to_fahrenheit(*self*)

test__rankine_to_kelvin(*self*)

Inherited from object: __delattr__, __getattribute__, __hash__, __reduce__, __reduce_ex__, __setattr__
Inherited from type: __new__
Inherited from TestCase: __init__, __call__, __repr__, __str__, assert_, assertAlmostEqual, assertAlmostEquals, assertEqual, assertEquals, assertNotAlmostEqual, assertNotAlmostEquals, assertNotEqual, assertNotEquals, assertRaises, countTestCases, debug, defaultTestResult, fail, failIf, failIfAlmostEqual, failIfEqual, failUnless, failUnlessAlmostEqual, failUnlessEqual, failUnlessRaises, id, run, shortDescription

16 Module weather.units.test_wind

Unit tests the pressure module.

16.1 Functions

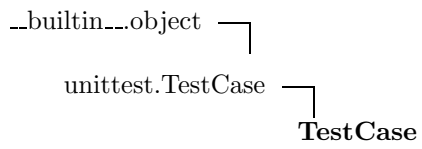
main()

usage()

16.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'\n  python $0\n'`

16.3 Class TestCase



16.3.1 Methods

setUp(*self*)

Hook method for setting up the test fixture before exercising it.

Overrides: unittest.TestCase.setUp extit(inherited documentation)

tearDown(*self*)

Hook method for deconstructing the test fixture after testing it.

Overrides: unittest.TestCase.tearDown extit(inherited documentation)

test__ft_min_to_mph(*self*)

test__ft_sec_to_knots(*self*)

test__ft_sec_to_mph(*self*)

test__km_hr_to_knots(*self*)

test__km_hr_to_mph(*self*)

test__knots_to_ft_sec(*self*)

test__knots_to_km_hr(*self*)

test__knots_to_m_sec(*self*)

test__knots_to_mph(*self*)

test__knots_to_nmph(*self*)

test__m_sec_to_knots(*self*)

test__m_sec_to_mph(*self*)

test__mph_to_ft_min(*self*)

test__mph_to_ft_sec(*self*)

test__mph_to_km_hr(*self*)

test__mph_to_knots(*self*)

test__mph_to_m_sec(*self*)

test__nmph_to_knots(*self*)

Inherited from object: __delattr__, __getattribute__, __hash__, __reduce__, __reduce_ex__, __setattr__
Inherited from type: __new__
Inherited from TestCase: __init__, __call__, __repr__, __str__, assert_, assertAlmostEqual, assertAlmostEquals, assertEqual, assertEquals, assertNotAlmostEqual, assertNotAlmostEquals, assertNotEqual, assertNotEquals, assertRaises, countTestCases, debug, defaultTestResult, fail, failIf, failIfAlmostEqual, failIfEqual, failUnless, failUnlessAlmostEqual, failUnlessEqual, failUnlessRaises, id, run, shortDescription

17 Module weather.units.wind

wind related conversion functions

17.1 Functions

ft_min_to_mph(*ft*)

Feet/minute (ft/min) to miles/hour (mph)

ft_sec_to_knots(*ft*)

Feet/second (ft/s) to knots (kt)

ft_sec_to_mph(*ft*)

Feet/second (ft/s) to miles/hour (mph)

km_hr_to_knots(*km*)

Kilometers/hour (kph) to knots (kt)

km_hr_to_mph(*km*)

Kilometers/hour (kph) to miles/hour (mph)

knots_to_ft_sec(*kts*)

Knots (kt) to feet/second (ft/s)

knots_to_km_hr(*kts*)

Knots (kt) to kilometers/hour (kph)

knots_to_m_sec(*kts*)

Knots (kt) to meters/second (m/s)

knots_to_mph(*kts*)

Knots (kt) to miles/hour (mph)

knots_to_nmph(*kts*)

Knots (kt) to nautical miles/hour (nmph)

m_sec_to_knots(*m*)

Meters/second (m/s) to knots (kt)

m_sec_to_mph(*m*)

Meters/second (m/s) to miles/hour (mph)

mph_to_ft_min(*mph*)

Miles/hour (mph) to feet/minute (ft/min)

mph_to_ft_sec(*mph*)

Miles/hour (mph) to feet/second (ft/s)

mph_to_km_hr(*mph*)

Miles/hour (mph) to kilometers/hour (kph)

mph_to_knots(*mph*)

Miles/hour (mph) to knots (kt)

mph_to_m_sec(*mph*)

Miles/hour (mph) to meters/second (m/s)

nmph_to_knots(*mph*)

Nautical miles/hour (nmph) to knots (kt)

17.2 Variables

Name	Description
__author__	**Value:** `'Christopher Blunck'`
__email__	**Value:** `'chris@wxnet.org'`
__revision__	**Value:** `'$Revision: 1.6 $'`
__usage__	**Value:** `'this module should not be run via the command line'`

Index