

pyHomework

`pyHomework` allows you to write quizzes, problem sets, and quizzes about problem sets in Python. Why would you want to do this?

- Writing code is funner than writing homework assignments.
- Answers to questions can be calculated in python.
- Updating numbers for a question is trivial, just update the variable storing the number.

`pyHomework` was born when I decided to start assigning standalone problem sets for my physics classes and give online quizzes (with Blackboard) over the problem set that could be graded automatically. I wanted a way to assign a “traditional” physics problem set but I could not grade every problem I assigned to a class of 30 - 40 students and needed a way to motivate the students to do the homework. Creating a problem set is not difficult, but giving a quiz over the problem set requires a set of solutions. Not only that, but the quiz questions do not need to restate the entire problem that was on the problem set, they just need to ask for some information about the problem’s solution. So the quiz questions need to reference problem numbers, and this turned out to be the difficult part. It is *very* easy to make a mistake and reference the wrong question number, which just confuses everybody involved. Even if you manage to get the references right the first time, it is difficult to add or change questions to the problem set in the future because the problem numbers may change and the quiz has to be updated. To top it all off, creating a quiz in Blackboard is a *pain*.

What eventually became `pyHomework` started off as a Blackboard quiz generator. I wrote a simple script that would read quizzes in YAML format and generate a text file that could be uploaded to Blackboard (the text file format is difficult to edit by hand). Now, it is possible to write a python program that will generate a problem set as a PDF document (using LaTeX) and a quiz file that can be uploaded to Blackboard. It tries to be flexible while automating much of the repetitive tasks associated with writing a homework assignment.

Dependencies

`pyHomework` depends on `pyErrorProp`

Examples

Here is an example of how I use `pyHomework` in my classes. First, there is a bit of boiler plate code that is required to setup a homework assignment. So I create a `BoilerPlate.py` file that does this setup.

```

import os, tempfile
from pyHomework import *

e = EquationsCollection()
s = e.s
c = ConstantsCollection(units, 3) # all constants to 3 sig figs
e.set_constants(c)

ass = HomeworkAssignment()
ass.add_quiz()
quiz = ass.get_quiz()

courseName = "UNKNOWN"
semester = "UNKNOWN"
module = "UNKNOWN"

def Setup(_file_):
    absfilename = os.path.abspath( _file_ )
    filename = os.path.basename( absfilename )
    absdirname = os.path.dirname( absfilename )

    #configure the title, header, and footer that will be generated
    #each of these have a default, so they can be left out
    global ass
    global quiz
    global courseName
    global semester
    global module

    courseName = GetCourseName(absfilename)
    semester = GetSemester(absfilename)
    module = GetModule(absfilename)

    ass.config( {"title" : "Module %s Homework" % module
                , "LH" : "%s - %s" % (courseName, semester)
                , "CH" : "FHSU"
                , "RH" : 'Mod %s' % module
                , "LF" : ""
                , "CF" : ""
                , "RF" : "Powered by \LaTeX" } )

def Build():
    # build a PDF and Bb quiz
    # make sure to get the basename before we change directories

```

```

global ass
global courseName,module

basename = ("%s-HW_%s" % (courseName,module)).replace(' ', '_')

workdir = '_' + module
if not os.path.exists( workdir ):
    os.makedirs( workdir )
os.chdir( workdir )

# # this will create the pdf
ass.build_PDF(basename+'.pdf')
# # this will create a yaml file that can be parsed by BbQuiz.py to generate a blackboard
ass.write_quiz_file(basename+'-Quiz.txt')

def GetSemester( filename ):
    '''Return the semester string, i.e. 'Spring 2015'. '''
    # some code to determine the semester
    return "Spring 2015"

def GetCourseName( filename ):
    '''Return the course name, i.e. 'Phys 212'. '''
    # some code to determine the course name
    return "Phys 212"

def GetModule( filename ):
    '''Return the module name, i.e. '01' or 'Electric_Charge'. '''
    # some code to determine the module name
    return "01"

```

This file defines two functions that we will use, `Setup()` and `Build()`, but it also imports modules we will use into the current namespace. So, here is an example of a homework assignment.

```

#!/bin/env python
from BoilerPlate import *

# the setup function from the BoilerPlate module
# it creates the assignment, equations, and constants instances
Setup(__file__)

with ass._add_question() as q:
    q.add_text(r'''
        What force is required to make a \SI{2.0}{\kilo\gram} accelerate at \SI{5.0}{\meter\per\second}
    ''')

```

```

'''
Mass = Q_(2,'kg')          # Q_ is an alias for the pint Quantity class from the UnitRe
Acceleration = Q_(2,'m/s^2')

with quiz._add_question() as qq:
    qq.add_text(r'''
    What is the force?
    ''')

    with qq._set_answer( NumericalAnswer ) as a:
        a.quantity = Mass * Acceleration

with ass._add_question() as q:
    q.add_text(r'''
    A \SI{3.0}{\kilo\gram} mass is dropped from a height of \SI{10}{\meter}.
    ''')

    Mass = Q_(3,'kg')
    Height = Q_(10,'m')

    PE = Mass*Q_(9.8,'m/s^2')*Height

    with q._add_part() as p:
        p.add_text(r'''
        What is the velocity of the mass at the moment it strikes the ground.
        ''')

        with quiz._add_question() as qq:
            qq.add_text(r'''
            What is the velocity?
            ''')

            with qq._set_answer( NumericalAnswer ) as a:
                a.quantity = np.sqrt( (2*PE)/Mass )

with q._add_part() as p:
    p.add_text(r'''
    How much work must be done by the normal force of the ground to stop the mass?
    ''')

    with quiz._add_question() as qq:
        qq.add_text(r'''
        How much work must be done?

```

```

'''

with qq._set_answer( NumericalAnswer ) as a:
    a.quantity = PE

# the build function from the BoilerPlate module
# this creates the PDF and Bb quiz file
Build()

```

Running this program will create a directory named '__01' that contains a PDF file of the homework assignment and a plain text file that can be uploaded to Blackboard.