

# OwlTest: Automated Feedback and Assessment

## Testing

It is very important to test your programs as you write them. You should carefully think about what you are trying to accomplish and write tests to make sure your program does what you expect. For the project in this course, that means that you should run your functions with a variety of inputs as you develop them and make sure they return the results you would expect. You shouldn't just try arbitrary inputs. You should think about where things might go wrong and try those inputs. Once you believe your function behaves in a reasonable manner, you are ready to submit it to the automated grader for feedback and assessment.

## OwlTest

OwlTest is a tool that provides a preliminary grade and feedback on your program. When you submit to OwlTest, there is no identifying information tying the submission to you. So, it is an anonymous way for you to run the automated grader on your code so that you can get feedback about how well your program runs. In the project description, there is an OwlTest link that will take you to a web page where you can submit your code. You simply paste a CodeSkulptor URL into the form or upload a file from your computer and then click the "Submit to OwlTest" button.

OwlTest is not particularly fast, so you should not use it instead of your own testing. However, it does run a suite of tests on all of your code to determine how well your code works. Each function is tested independently, so you will receive feedback for each function. This allows you to submit to OwlTest even if you haven't completed the entire assignment.

The OwlTest page tells you which functions will be tested and how many points each function is worth towards your final grade. For each graded function, OwlTest will either print nothing (meaning you passed all of the tests!) or it will print a line telling you how many points you lost for that function and giving you a sample test that you failed. For example:

```
[-15.0 pts] days_in_month(1, 1) expected 31 but received 0
```

You can interpret this as follows:

- **[-15.0 pts]** You lost 15 points on this function.

- **days\_in\_month(1, 1)** This is the test that was run. You can copy and paste this code directly into your Python file to run the same test!
- **expected 31** This was the expected output of the test (there are 31 days in January of year 1)
- **but received 0** This is what your code returned (the submitted function was broken and returned 0)

If you have more than one function that lost points, there will be multiple lines like this. Note that they might be in different orders each time you submit to OwlTest. The tests are run in an arbitrary order and can be different each time.

The best way to proceed after losing points in OwlTest is to cut and paste the test (in this case `days_in_month(1, 1)`) directly into your Python file and run it. Then, you can work on your code until it returns the expected value.

OwlTest uses [Pylint](#) to check that you have followed the coding style guidelines for this class. Deviations from these style guidelines will result in deductions from your final score. Please read the feedback from Pylint closely. If you have questions, feel free to consult the [guide to common Pylint errors](#) and the class forums. If you violate the coding style guidelines, you will see a deduction for style. For example,

**[-1.0 pts] 1 style warnings found (maximum allowed: 20 style warnings)**

This particular message indicates that you lost 1 point for code style. Click on the other tabs of the output to get more details about the style warnings.

Your total score for the submission is shown at the top of the output, as follows:

**Score: 52.3/100**

Remember, OwlTest is *anonymous*. Therefore, this grade is not (and can not be!) reported to Coursera. This is just the score you would receive if you were to submit that file to Coursera. Once you are happy with your score, you should submit your file to Coursera.

## Coursera LTITest

Coursera LTITest formally grades your submission and submits your grade to Coursera. It operates just like OwlTest, but you must access it by opening an "LTI Tool" in Coursera. This passes your identity to the grader. After grading your submission, the grade is then sent back to Coursera so that it can be recorded.

You can always tell whether you are using OwlTest or Coursera LTITest by the color of the background. OwlTest has a *pale yellow* background. Coursera LTITest has a *white* background.

To get to Coursera LTITest, you must go to the assignment within the course. That page will tell you that the course uses a third-party tool to enhance your learning experience. You will be required to agree to the Coursera Honor Code and then you will be able to click the "Open Tool" button. That will bring you to the Coursera LTITest page (with a white background). Coursera LTITest then operates just like OwlTest. You submit your code and it gives you the same feedback. But, it also submits your score to Coursera.

Each time you submit to Coursera LTITest, you should go back to the assignment within Coursera and re-open the tool. Do not just refresh the page or otherwise navigate to the URL.

## Submission History

This tool allows you to see all prior submissions that you have made. Coursera keeps track of your score. You may submit to an assignment as much as you would like until you get a grade you are happy with. However, Coursera will not show you what you submitted. The "Project Submission History" tool takes you to a page that shows all of the submissions you made to Coursera LTITest. You can look at each of them in order to remember the feedback you were given. If you click the "Your Code" tab, it will tell you the CodeSkulptor URL that you submitted and the entire file will be there for you to view, as well.

✓ Complete

