Using Ok

CS 61A uses a program called `ok` to test and submit homework assignments, labs, and projects.

Every programming assignment will include a `.zip` archive that contains the following:

- Starter code
- A copy of `ok`

After extracting the contents of the archive, you can begin your assignment.

Signing in with Ok

To get started, open your terminal, and `cd` into the right directory (you should see a file called `ok` when you `ls` in the right directory).

Try the following command:

```
python3 ok
```

This runs the tests (don't worry if the tests fail at first; you haven't written code yet!).

The first time you run `ok`, you will be asked for your bCourses email. This should be your Berkeley email address (ending with `@berkeley.edu`). If you have more than one, please use the one that you log into your bConnected Google account with.

If you don't have a Berkeley email address, you can create one here (https://mybconnected.berkeley.edu/my/account/index) by logging in to your CalNet account.

After typing in your email, your web browser will open an authentication page. Click "Accept" to authenticate `ok`.

Troubleshooting

Not enrolled

If you see an error message that indicates you are not enrolled in the course, make sure you are using the email address that would appear on the course roster. If the email you used is correct, you may continue to use the email; your submissions will still be saved. You must contact your TA for that email address to be enrolled.

Wrong email

If you typed your email incorrectly, you can re-authenticate with the following command:

```
python3 ok --authenticate
```

Can't authenticate/browser issues/redirections to `127.0.0.1` /etc.

Try running with `--no-browser` :

```
python3 ok --authenticate --no-browser
```

Also, double-check your internet connection. If you're on campus, try using AirBears2 (https://ist.berkeley.edu/airbears/).

Crashed or did not load

Make sure you have the 64-bit version of Python 3 installed. You can check whether you have the incorrect 32-bit version by running the following command in your terminal.

```
python3 -c "import struct,platform;print(8 * struct.calcsize('P'), platform.python_versio
```

Testing with Ok

After writing some code, you can test your code with `ok` in various ways.

Test specific questions

To test a specific question, use the `-q` option with the name of the question:

```
python3 ok -q <question>
```

Test all questions

You can run all the tests with the following command:

```
python3 ok
```

Display all tests

By default, only tests that **fail** will appear. If you want to see how you did on all tests, you can use the `-v` option:

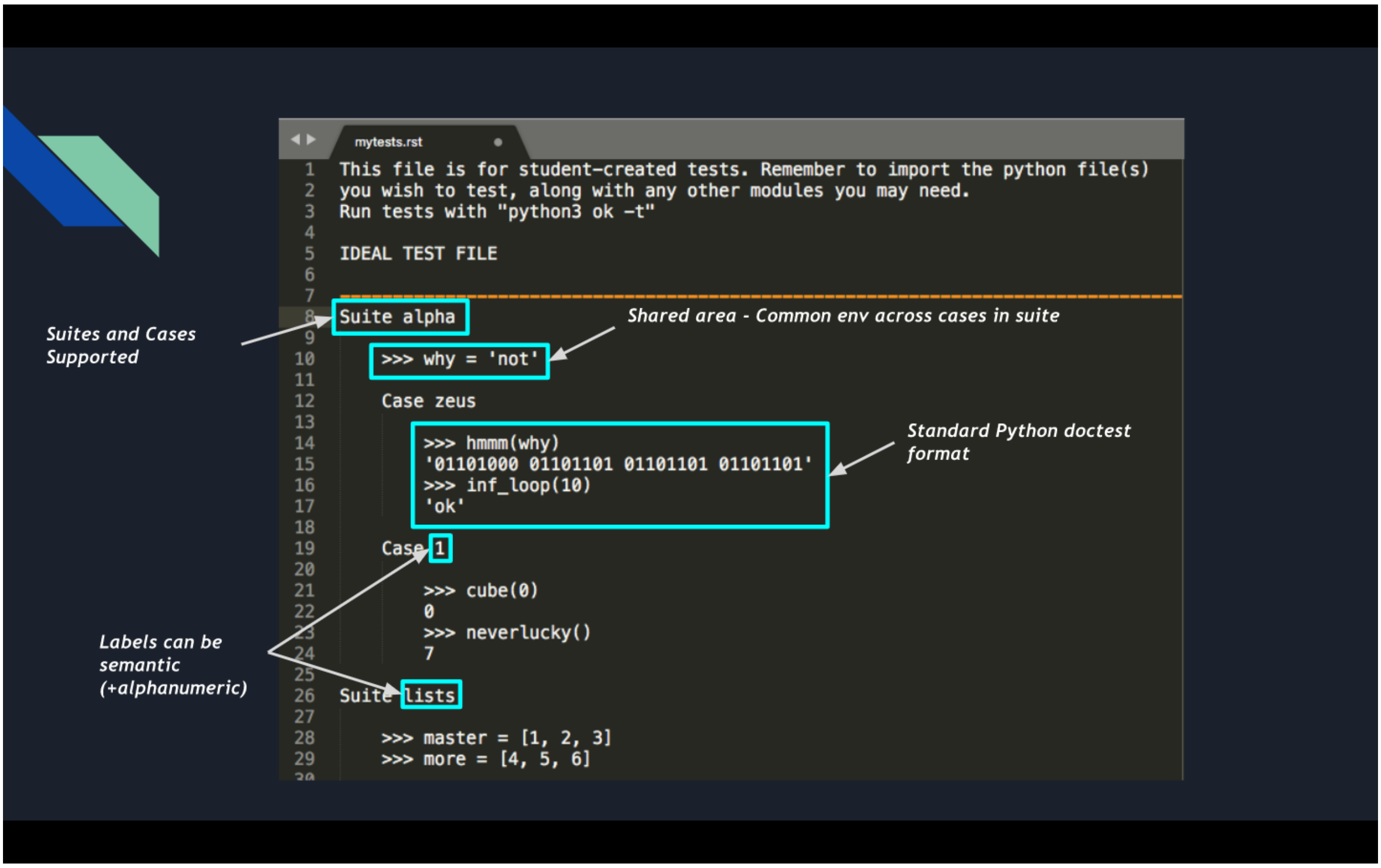
```
python3 ok -v
```

Test locally

If you do not want to send your progress to our server or you have any problems logging in, add the `--local` flag to block all communication:

```
python3 ok --local
```

Adding your own tests



mytests.rst
Suites and Cases Supported
Shared area - Common env across cases in suite
Standard Python doctest format
Labels can be semantic (+alphanumeric)

```
This file is for student-created tests. Remember to import the python file(s)
you wish to test, along with any other modules you may need.
Run tests with "python3 ok -t"

IDEAL TEST FILE

Suite alpha

    >>> why = 'not'

    Case zeus
        >>> hmmm(why)
        '01101000 01101101 01101101 01101101'
        >>> inf_loop(10)
        'ok'

    Case 1

        >>> cube(0)
        0
        >>> neverlucky()
        7

Suite lists

    >>> master = [1, 2, 3]
    >>> more = [4, 5, 6]
```

You can write your own tests and run them using `ok`. By default, a test file will be named `mytests.rst`. You may use a different name, but you will need to specify it when running tests.

Running your own tests

To run all your tests in `mytests.rst` with verbose results:

```
python3 ok -t -v
```

If you put your tests in a different file or split your tests up into multiple files:

```
python3 ok -t your_new_filename.rst
```

To run just the tests from suite 1 case 1 in `mytests.rst`:

```
python3 ok -t --suite 1 --case 1
```

You might have noticed that there's a "test coverage" percentage for your tests (note that coverage statistics are only returned when running all tests). This is a measure of your test's code coverage (https://en.wikipedia.org/wiki/Code_coverage).

To receive guidance on which lines you should test to increase your coverage:

```
python3 ok -t -cov
```

Code coverage won't include `ok` tests, so the coverage percentage might be higher in reality.

While code coverage is a useful tool, you should not get fixated on this number. It is better to write tests that help you complete the problem and make life easier instead of achieving a higher coverage.

Submit assignment

When you are ready to submit, run `ok` with the `--submit` option:

```
python3 ok --submit
```

After submitting, `ok` will display a submission URL, with which you can view your submission on okpy.org (https://okpy.org).

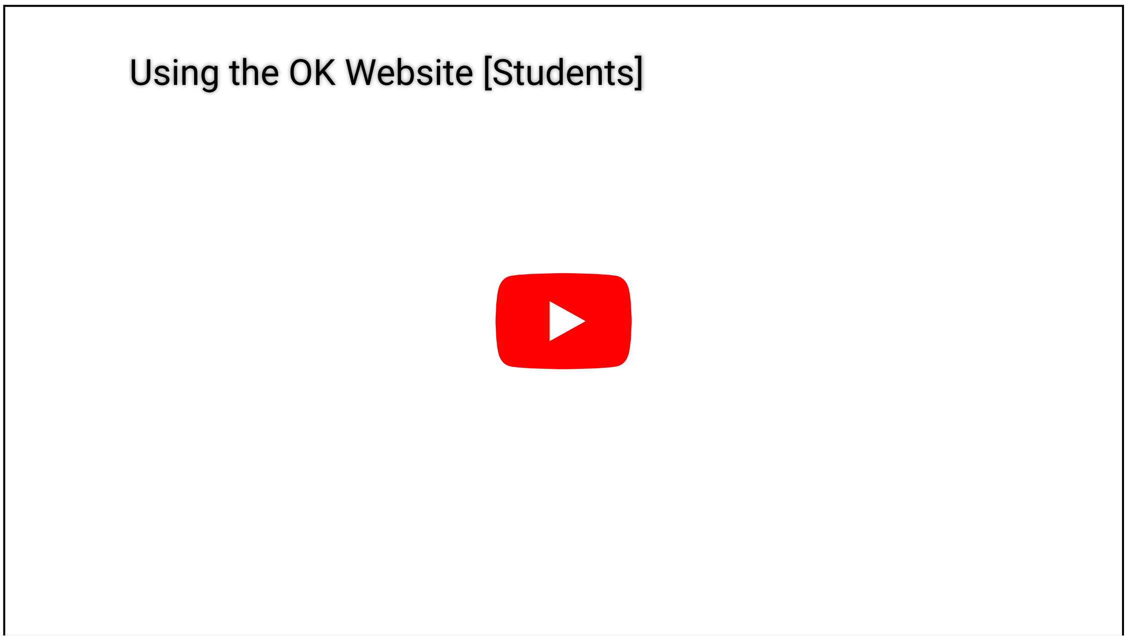
Viewing submissions

You can go to okpy.org (https://okpy.org) to check your submissions and backups. Make sure you sign in with the same bCourses email that you authenticated with.

FAQ

Adding partners on okpy.org

This video (<a href='https://youtu.be/3qx_RfKbvuQ'> https://youtu.be/3qx_RfKbvuQ </a>) walks through adding partners on okpy.org (https://okpy.org).



CS 61A (/~cs61a/fa20/)

Weekly Schedule (/~cs61a/fa20/weekly.html)

Office Hours (/~cs61a/fa20/office-hours.html)

Staff (/~cs61a/fa20/staff.html)

Resources (/~cs61a/fa20/resources.html)

Studying Guide (/~cs61a/fa20/articles/studying.html)

Debugging Guide (/~cs61a/fa20/articles/debugging.html)

Composition Guide (/~cs61a/fa20/articles/composition.html)

Policies (/~cs61a/fa20/articles/about.html)

Assignments (/~cs61a/fa20/articles/about.html#assignments)

Exams (/~cs61a/fa20/articles/about.html#exams)

Grading (/~cs61a/fa20/articles/about.html#grading)