

Load Magic for More Fluid Activities

Sarah M Brown

Division of Computer Sciences
University of California, Berkeley

CarpentryCon 2018

How do I logistically manage exercises in a workshop?

How do I logistically manage exercises in a workshop?

With Load Magic!

```
In [ ]: %load code/example
```

How do I logistically manage exercises in a workshop?

With Load Magic!

```
In [ ]: # %load code/example
```

```
a = 4
```

```
b = 3
```

```
c = a**2 + b**2
```

```
In [ ]: |
```

How I set it up

The screenshot shows the GitHub interface for the repository 'python-novice-gapminder-files' by user 'brownsarahm'. The repository has 26 commits, 7 branches, 0 releases, and 1 contributor. The 'testworkshop' branch is selected. A file list shows 'code' (updated), 'data' (initial draft), 'worksheets' (add), and '.gitignore' (preparation). A 'Clone with HTTPS' modal is open, showing the URL 'https://github.com/brownsarahm/python' and a 'Download ZIP' button. The footer includes the GitHub copyright notice and a list of links: Contact GitHub, API, Training, Shop, Blog, About.

Search or jump to... Pull requests Issues Marketplace Explore

brownsarahm / python-novice-gapminder-files Unwatch 1 Star 1 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

helper files in a separate repo for auto creation of zip link and more download options Edit

Add topics

26 commits 7 branches 0 releases 1 contributor

Branch: testworkshop New pull request Create new file Upload files Find file Clone or download

This branch is 1 commit ahead, 2 commits behind master.

brownsarahm preparation for testworkshop

code	updated excercises
data	initial draft of excercises
worksheets	add worksheets
.gitignore	preparation for testworkshop

3 months ago 2 months ago

Help people interested in this repository understand your project by adding a README. Add a README

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

<https://github.com/brownsarahm/python-novice-gapminder-files/archive/testworkshop.zip>

It helps students focus on the coding during exercises

Fewer syntax errors, less frustration

No window switching, lower cognitive load

Overall, more time thinking about and completing exercises

Load Magic can make prep and management better too

Come Chat at the Poster Session

```
1 ---
2 layout: exercise
3 name: notebook
4 episode: 02-basics
5 solution: "1. Only the last result is printed.
6
7 2. The 4 'items' are printed by the REPL, but not in the same way as the print
8 statement. The items in quotes are treated as separate strings, for the variables a and
9 b the values are printed. All four items are treated as a 'tuple' which are shown in
10 parentheses, a tuple is another datatype in Python that allows you to group things
11 together and treat as a unit. We can tell that it is a tuple because of the '()'
12
13 A complete set of Python operators can be found in the [official
14 documentation](https://docs.python.org/3.5/library/operator.html) . The documentaion
15 may appear a bit confusing as it initially talks about operators as functions whereas
16 we generally use them as 'inplace' operators. Section 10.3.1 provides a table which
17 list all of the available operators, not all of which are relevant to basic arithmetic."
18 ---
19
20 Example code:
21 ---
22 print("a =", a, "and b =", b)
23 print(a + b)      # addition
24 print(a * b)      # multiplication
25 print(a - b)      # subtraction
26 print(a / b)      # division
27 print(b ** a)      # exponentiation
28 print(2 * a % b)   # modulus - returns the remainder
29
30 1. Create a new cell and paste into it the assignments to the variables a and b and the
31 contents of the code above. Remove all of the calls to the print function so you only
32 have the expressions that were to be printed and run the code. What is returned?
33
34 2. Make a new cell with only '("a =", a, "and b =", b)' in it. How does this output
35 differ from when we used the print function?
```

/home/mb/Documents/carpentries/python-socialsci_episodes/old_ex.md

```
3 > 1. Create a new cell and paste into it the assignments to the variables a and b and
4 the contents of the code cell above with all of the print statements. Remove all of the
5 calls to the print function so you only have the expressions that were to be printed
6 and run the code. What is returned?
7 >
8 > 2. How remove all but the first line (with the 4 items in it) and run the cell again.
9 How does this output differ from when we used the print function?
10 >
11 > 3. Practice assigning values to variables using as many different operators as you
12 can think of.
13 >
14 > 4. Create some expressions to be evaluated using parentheses to enforce the order of
15 mathematical operations that you require
16 >
17 > ## Solution
18 >
19 > 1. Only the last result is printed.
20 > 2. The 4 'items' are printed by the REPL, but not in the same way as the print
21 statement. The items in quotes are treated as separate strings, for the variables a and
22 b the values are printed. All four items are treated as a 'tuple' which are shown in
23 parentheses, a tuple is another datatype in Python that allows you to group things
24 together and treat as a unit. We can tell that it is a tuple because of the '()'
25 >
26 > A complete set of Python operators can be found in the [official
27 documentation](https://docs.python.org/3.5/library/operator.html) . The documentaion
28 may appear a bit confusing as it initially talks about operators as functions whereas
29 we generally use them as 'inplace' operators. Section 10.3.1 provides a table which
30 list all of the available operators, not all of which are relevant to basic arithmetic.
31 >
32 > (: .solution)
33 (: .challenge)
```

Note the lack of >>