(https://www.edx.org)

MITx: 6.00.1x Introduction to Computer Science and Programming Using Python

sandipan_dey (/dashboard) ▾

Courseware (/courses/MITx/6.00.1_4x/3T2014/courseware) Updates & News (/courses/MITx/6.00.1_4x/3T2014/info)

Calendar (/courses/MITx/6.00.1_4x/3T2014/89309559b0414f6d8cbef9e48ca19f4b/) Wiki (/courses/MITx/6.00.1_4x/3T2014/course_wiki)

Discussion (/courses/MITx/6.00.1_4x/3T2014/discussion/forum) Progress (/courses/MITx/6.00.1_4x/3T2014/progress)

PART IV: USER-SPECIFIED TRIGGERS (10/10 points)

Right now, your triggers are specified in your Python code, and to change them, you have to edit your program. This is very user-unfriendly. (Imagine if you had to edit the source code of your web browser every time you wanted to add a bookmark!)

Instead, we want you to read your trigger configuration from a `triggers.txt` file every time your application starts, and use the triggers specified there.

Consider the following example trigger configuration file:

```
        # subject trigger named t1
        t1 SUBJECT world

        # title trigger named t2
        t2 TITLE Intel

        # phrase trigger named t3
        t3 PHRASE New York City

        # composite trigger named t4
        t4 AND t2 t3

        # the trigger set contains t1 and
        t4
        ADD t1 t4
```

The example file specifies that four triggers should be created, and that two of those triggers should be added to the trigger list:

- A trigger that fires when a subject contains the word 'world' (`t1`).
- A trigger that fires when the title contains the word 'Intel' and the news item contains the phrase 'New York City' somewhere (`t4`).

The two other triggers (`t2` and `t3`) are created but not added to the trigger set directly. They are used as arguments for the composite `AND` trigger's definition (`t4`).

Each line in this file does one of the following:

- is blank
- is a comment (begins with a `#`)
- defines a named trigger
- adds triggers to the trigger list.

Each type of line is described below.

- **Blank:** blank lines are ignored. A line that consists only of whitespace is a blank line.
- **Comments:** Any line that begins with a `#` character is ignored.

- **Trigger definitions:** Lines that do not begin with the keyword `ADD` define named triggers. The first element in a trigger definition is the name of the trigger. The name can be any combination of letters without spaces, except for "ADD". The second element of a trigger definition is a keyword (e.g., `TITLE`, `PHRASE`, etc.) that specifies the kind of trigger being defined. The remaining elements of the definition are the trigger arguments. What arguments are required depends on the trigger type:
    - `TITLE` : a single word.
    - `SUBJECT` : a single word.
    - `SUMMARY` : a single word.
    - `NOT` : the name of the trigger that will be NOT'd.
    - `AND` : the names of the two other triggers that will be AND'd.
    - `OR` : the names of the two other triggers that will be OR'd.
    - `PHRASE` : a phrase.
- **Trigger addition:** A trigger definition should create a trigger and associate it with a name but should not automatically add that trigger to the running trigger list. One or more `ADD` lines in the .txt file will specify which triggers should be in the trigger list. An addition line begins with the `ADD` keyword. Following `ADD` are the names of one or more previously defined triggers. These triggers will be added to the the trigger list.

PROBLEM 11

We have implemented the function `readTriggerConfig(filename)` for you. We've written code to open the file and throw away all the lines that don't begin with instructions (e.g. comments and blank spaces), and then reads in the code that defines triggers and instantiates the triggers by making a call to the helper function `makeTrigger`. The function `readTriggerConfig` then returns a list of triggers specified in the configuration file.

First, read through the definition of `readTriggerConfig`. You should be able to understand everything this function is doing at this point in the course.

Next, implement the function `makeTrigger(triggerMap, triggerType, params, name)`. This helper function should build and return a trigger depending on its type. It also keeps track of triggers and names in a map. We have defined for you the specifications for this function to make it easier for you to write. Be sure you understand how `readTriggerConfig` is using this function; that will make implementation easier.

Once that's done, modify the code within the function `main_thread` to use the trigger list specified in your configuration file, instead of the one we hard-coded for you:

```
        # TODO: Problem 11
        # After implementing makeTrigger, uncomment the line below:
        # triggerlist = readTriggerConfig("triggers.txt")

                NOTE that you may have to change "triggers.txt" to the full
                filepath (eg, C:/Documents/triggers.txt) if the
                RSS reader window does not pop up. You can reset Canopy with
                Run -> Restart Kernel or CTRL key with the dot key.
```

After completing Problem 11, you can try running `ps7.py`, and depending on how your `triggers.txt` file is defined, various RSS news items should pop up for easy reading. The code runs an infinite loop, checking the RSS feed for new stories every 60 seconds.

Hint: If no stories are popping up, open up `triggers.txt` and change the triggers to ones that reflect current events (if you don't keep up, just pick a trigger that would fire on one of the current Google news (http://news.google.com/) stories).

Hint: You may find the str.join (http://docs.python.org/2/library/stdtypes.html#str.join) useful.

```
# Enter your code for makeTrigger in this box
def makeTrigger(triggerMap, triggerType, params, name):
    """
    Takes in a map of names to trigger instance, the type of trigger to make,
    and the list of parameters to the constructor, and adds a new trigger
    to the trigger map dictionary.

    triggerMap: dictionary with names as keys (strings) and triggers as values
    triggerType: string indicating the type of trigger to make (ex: "TITLE")
    params: list of strings with the inputs to the trigger constructor (ex: ["world"])
    name: a string representing the name of the new trigger (ex: "t1")

    Modifies triggerMap, adding a new key-value pair for this trigger.

    Returns a new instance of a trigger (ex: TitleTrigger, AndTrigger).
    """
```

Correct

Test results

Hide output

CORRECT

`Test: 1 make TitleTrigger`

Output:

```
Trigger file:
t1 TITLE NASA

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
TitleTrigger made with word: NASA
 --> Is trigger t1 in triggerMap? True
```

`Test: 2 make basic triggers`

Output:

```
Trigger file:
t1 TITLE NASA
t2 SUBJECT Mars
t3 SUMMARY rocket

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
TitleTrigger made with word: NASA
 --> Is trigger t1 in triggerMap? True
SubjectTrigger made with word: Mars
 --> Is trigger t2 in triggerMap? True
SummaryTrigger made with word: rocket
 --> Is trigger t3 in triggerMap? True
```

`Test: 3 make PhraseTriggers`

Output:

```
Trigger file:
t1 PHRASE NASA
t2 PHRASE Curiosity Mars rover
t3 PHRASE Neil deGrasse Tyson

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
PhraseTrigger made with phrase: NASA
 --> Is trigger t1 in triggerMap? True
PhraseTrigger made with phrase: Curiosity Mars rover
 --> Is trigger t2 in triggerMap? True
PhraseTrigger made with phrase: Neil deGrasse Tyson
 --> Is trigger t3 in triggerMap? True
```

Test: 4 make NotTriggers

Output:

```
Trigger file:
t1 TITLE Manitoba
t2 NOT t1
t3 PHRASE British Colombia
t4 NOT t3

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
TitleTrigger made with word: Manitoba
 --> Is trigger t1 in triggerMap? True
NotTrigger made with trigger: TitleTrigger("Manitoba")
 --> Is trigger t2 in triggerMap? True
PhraseTrigger made with phrase: British Colombia
 --> Is trigger t3 in triggerMap? True
NotTrigger made with trigger: PhraseTrigger("British Colombia")
 --> Is trigger t4 in triggerMap? True
```

Test: 5 make AndTriggers

Output:

```
Trigger file:
t1 TITLE Manitoba
t2 PHRASE British Colombia
t3 AND t1 t2

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
TitleTrigger made with word: Manitoba
 --> Is trigger t1 in triggerMap? True
PhraseTrigger made with phrase: British Colombia
 --> Is trigger t2 in triggerMap? True
AndTrigger made with triggers: TitleTrigger("Manitoba") and PhraseTrigger("British Colombia")
 --> Is trigger t3 in triggerMap? True
```

Test: 6 make OrTriggers

Output:

```
Trigger file:
t1 SUBJECT Harper
t2 PHRASE Prime Minister
t3 OR t1 t2

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
SubjectTrigger made with word: Harper
 --> Is trigger t1 in triggerMap? True
PhraseTrigger made with phrase: Prime Minister
 --> Is trigger t2 in triggerMap? True
OrTrigger made with triggers: SubjectTrigger("Harper") and PhraseTrigger("Prime Minister")
 --> Is trigger t3 in triggerMap? True
```

```
Test: 7 make composite triggers
```

Output:

```
Trigger file:
t1 SUBJECT Cameron
t2 SUBJECT Gillard
t3 SUBJECT Harper
t4 PHRASE Prime Minister
t5 AND t1 t4
t6 AND t2 t4
t7 OR t3 t4

**** RUNNING MAKETRIGGER ON TRIGGER FILE ****
SubjectTrigger made with word: Cameron
 --> Is trigger t1 in triggerMap? True
SubjectTrigger made with word: Gillard
 --> Is trigger t2 in triggerMap? True
SubjectTrigger made with word: Harper
 --> Is trigger t3 in triggerMap? True
PhraseTrigger made with phrase: Prime Minister
 --> Is trigger t4 in triggerMap? True
AndTrigger made with triggers: SubjectTrigger("Cameron") and PhraseTrigger("Prime Minister")
 --> Is trigger t5 in triggerMap? True
AndTrigger made with triggers: SubjectTrigger("Gillard") and PhraseTrigger("Prime Minister")
 --> Is trigger t6 in triggerMap? True
OrTrigger made with triggers: SubjectTrigger("Harper") and PhraseTrigger("Prime Minister")
 --> Is trigger t7 in triggerMap? True
```

Hide output

Check Save *You have used 4 of 30 submissions*

Show related resources ▼

Show Discussion New Post

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code (https://www.edx.org/edx-terms-service)

Privacy Policy (Revised 4/16/2014) (https://www.edx.org/edx-privacy-policy)

About (https://www.edx.org/about-us)

News (https://www.edx.org/news)

Contact (https://www.edx.org/contact)

FAQ (https://www.edx.org/student-faq)

edX Blog (https://www.edx.org/edx-blog)

Donate to edX (https://www.edx.org/donate)

Jobs at edX (https://www.edx.org/jobs)

Twitter (https://twitter.com/edXOnline)

Facebook (http://www.facebook.com/EdxOnl

Meetup (http://www.meetup.com/edX-Global-Community)

LinkedIn (http://www.linkedin.com/company

Google+ (https://plus.google.com/+edXOnli