Client-Side Options (Part 1)

Robert M. Dondero, Ph.D.
Princeton University

Objectives

So far:

Client	Browser + HTML + JavaScript
Server	Python + Flask Java + Servlets Java + Spring JavaScript + Express

Objectives

- We will cover:
  - Desktop app programming
  - A Penny **desktop** client

Objectives

- More specifically…
- We will cover:
  - “High-level” desktop app programming
- We will not cover:
  - “Low-level” desktop app programming

Motivation

- **Question**: Why study desktop app programming?
- **Answers**:
  - Mainstream “advanced programming”
  - Illustrates event-driven programming
  - Illustrates concurrent programming
  - Important!

Agenda

- **Aside: Function def expressions**
- Aside: Multithreaded programming
- Java Swing
- Baseline
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Function Def Exprs

In **JavaScript**

With a function definition **statement**:

```
function compareLengths(word1, word2) {
    return word1.length - word2.length;
}
…
words.sort(compareLengths);
…
```

With a function definition **expression**:

```
…
words.sort(
    function(word1, word2) {
        return word1.length - word2.length;}
);
…
```

Function Def Exprs

In **JavaScript**
With an **arrow** function definition **expression:**

```
…
words.sort(
    (word1,word2) => word1.length-word2.length);
…
```

Arrow functions are a syntactic shortcut

Function Def Exprs

In **Python**

Lambda Expressions

Alonzo Church

Function Def Exprs

- In **Python**

```
lambda param1, param2 …: expression
```

  - The keyword `lambda`
  - (optionally) Parameters separated by commas
  - A colon
  - A single expression that uses the parameters

Function Def Exprs

In **Python**

Without a lambda expression:

```
def compare_lengths(word1, word2):
    return len(word1) - len(word2)
…
words.sort(compare_lengths)
…
```

With a lambda expression:

```
…
words.sort(
    lambda word1,word2: len(word1)-len(word2))
…
```

Function Def Exprs

- In **Java**:
  - ***Java lambda expressions***
    - New to Java SE 8

Function Def Exprs

In **Java**
Without a lambda expression:

```
class LengthComparator implements Comparator<String>
{
   public int compare(String word1, String word2)
   {
      return word1.length() - word2.length();
   }
}
...
String[] words;
...
Arrays.sort(words, new LengthComparator());
...
```

Function Def Exprs

In **Java**
With a lambda expression:

```
...
String[] words;
...
Arrays.sort(words,
    (String word1, String word2) ->
       word1.length() - word2.length() );
...
```

Function Def Exprs

In **Java**
With a lambda expression:

```
...
String[] words;
...
Arrays.sort(words,
    (word1, word2) -> word1.length() - word2.length()
);
...
```

Sometimes can omit parameter types

Function Def Exprs

- In **Java**
  - ***Functional interface***
    - An interface that declares a single method
  - It's OK to use a lambda expression in lieu of **an object of a class that implements a functional interface**

Function Def Exprs

- In **Java**
  - Java lambda expression observations
    - Function def expressions in a language that doesn't have functions!
    - Handy!
    - Inelegant?

Agenda

- Aside: Function def expressions
- **Aside: Multithreaded programming**
- Java Swing
- Baseline
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Multithreaded Programming

- Recall **spawning.py**

```
$ python spawning.py
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
main thread terminated
$
```

```
$ python spawning.py
blue
blue
blue
blue
red
red
red
red
red
red thread terminated
blue
main thread terminated
blue thread terminated
$
```

Multithreaded Programming

- See **Spawning1.java**

```
$ java Spawning1
main thread terminated
blue
blue
blue
blue
blue
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

```
$ java Spawning1
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

Multithreaded Programming

- **Problem**:
  - Java does not allow multiple inheritance
  - What if class PrinterThread already inherits from some class other than Thread?
- **Solution**: …

Multithreaded Programming

- See **Spawning2.java**

```
$ java Spawning2
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
blue thread terminated
red thread terminated
$
```

```
$ java Spawning2
blue
blue
main thread terminated
red
red
red
red
red
blue
blue
blue
blue thread terminated
red thread terminated
$
```

Multithreaded Programming

- **Note**:
  - `bluePrinterRunnable` is an object of a class that implements an interface that defines one method
  - `bluePrinterRunnable` can be replaced with a lambda expression
  - Same for `redPrinterRunnable`
- **And so…**

Multithreaded Programming

- See **Spawning3.java**

```
$ java Spawning3
main thread terminated
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
$
```

```
$ java Spawning3
main thread terminated
blue
blue
blue
red
red
red
red
red
red thread terminated
blue
blue
blue thread terminated
$
```

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- **Java Swing**
- Baseline
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Java Swing

Cross-platform desktop app libraries:

Desktop App Library	Language	Platform
GTK	C, C++	Linux, Windows, macOS
Qt	**C, C++**	**Linux, Windows, macOS**
wxWidgets	C, C++	Linux, Windows, macOS
Flutter	C, C++, Dart	Linux, WIndows, macOS, Android, iOS
PyQt	Python	Linux, Windows, macOS
Tkinter	Python	Linux, WIndows, macOS
AWT	Java	Linux, Windows, macOS
Swing	Java	Linux, Windows, macOS
JavaFX	Java	Linux, Windows, macOS
…	…	…

Java Swing

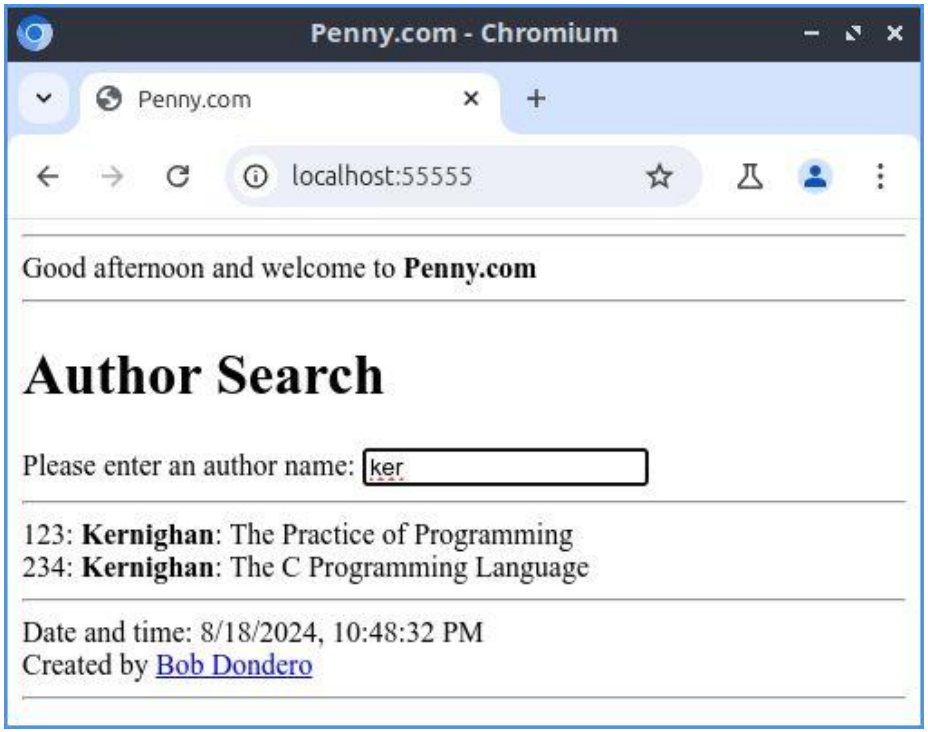
- Why study Java Swing
  - (Instead of some other GUI library)
  - I know it!
  - Popular
  - Bridges to Android programming

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- **Baseline**
- Penny desktop client
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Baseline

- See **PennyJson** app
  - (Almost) same as **PennyFlaskSpa** app from previous lecture



Baseline

- See **PennyJson** app (cont.)
  - runserver.py
  - penny.sql, penny.sqlilte
  - database.py
  - index.html
    - Irrelevant for this lecture
  - **penny.py**
    - Added IO_DELAY via an environment variable
      - Simulates slow DB
      - Shows how client performs with I/O bound server

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - **Version 1: Simple**
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Penny Desktop v1

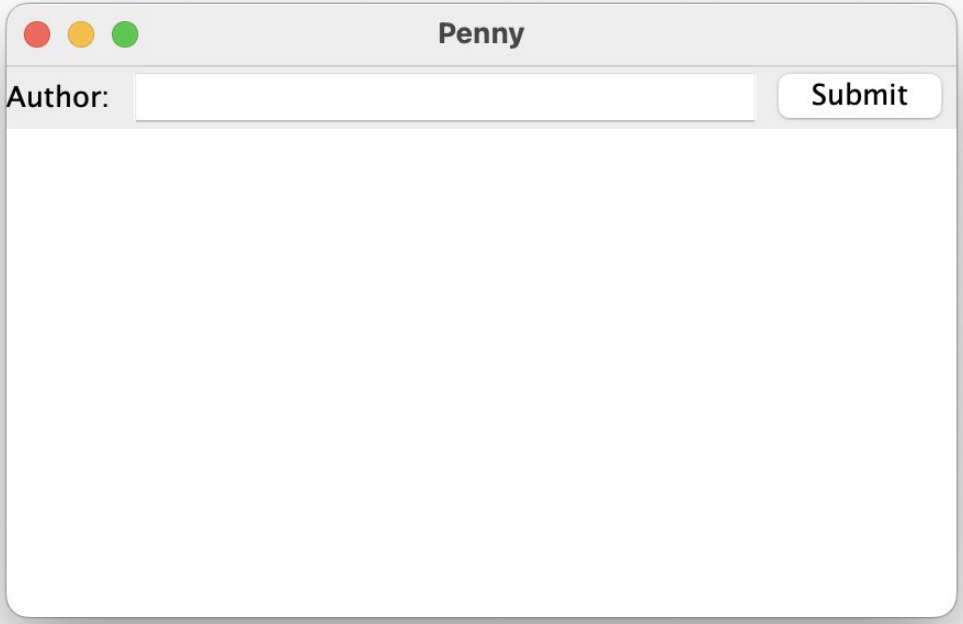
- Penny desktop v1

```
$ export IO_DELAY=1
$ cd PennyJson
$ python runserver.py 55555
```

```
$ cd PennySwing1
$ ./runclient http://localhost:55555
```

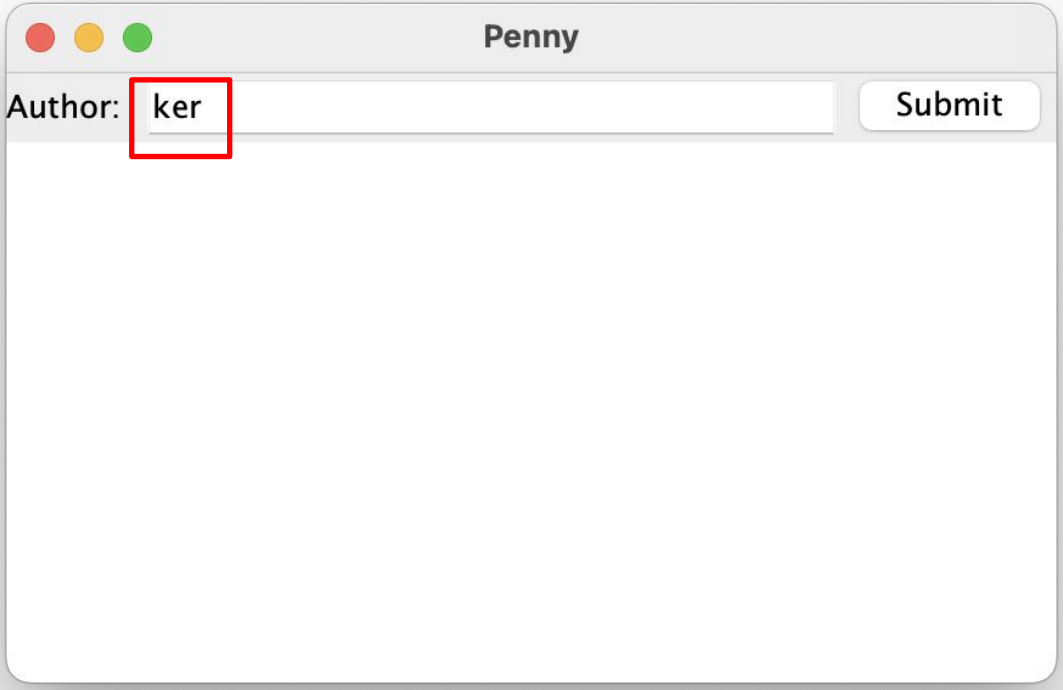
Penny Desktop v1

- Penny desktop v1 (cont.)



Penny Desktop v1

- Penny desktop v1 (cont.)

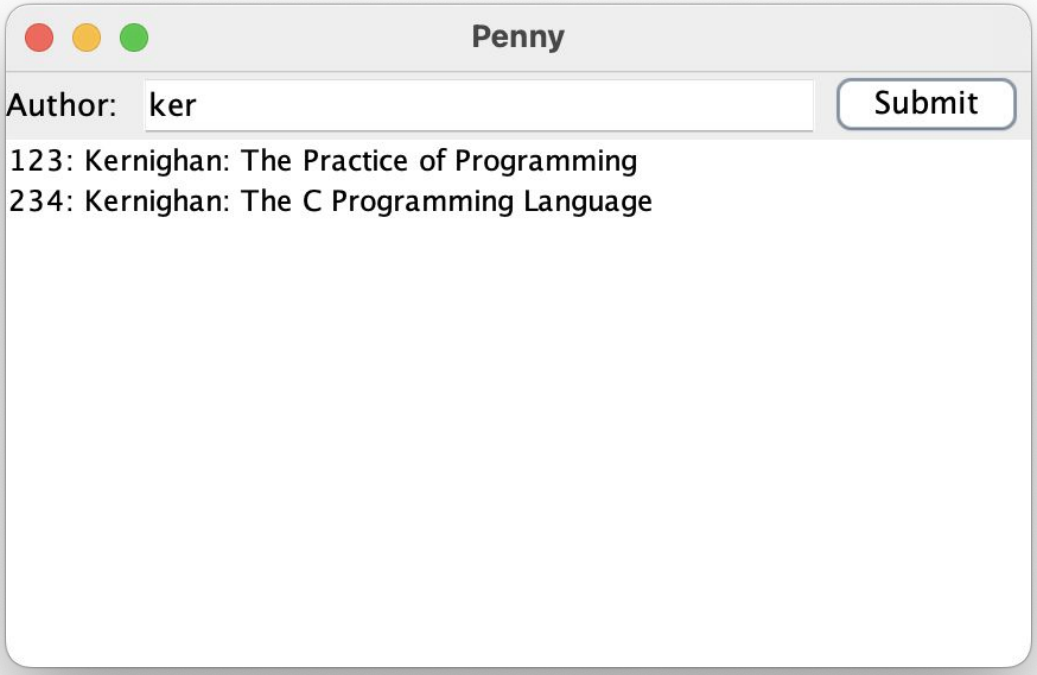


Penny Desktop v1

- Penny desktop v1 (cont.)

Penny Desktop v1

- Penny desktop v1 (cont.)



After a
1 second
delay

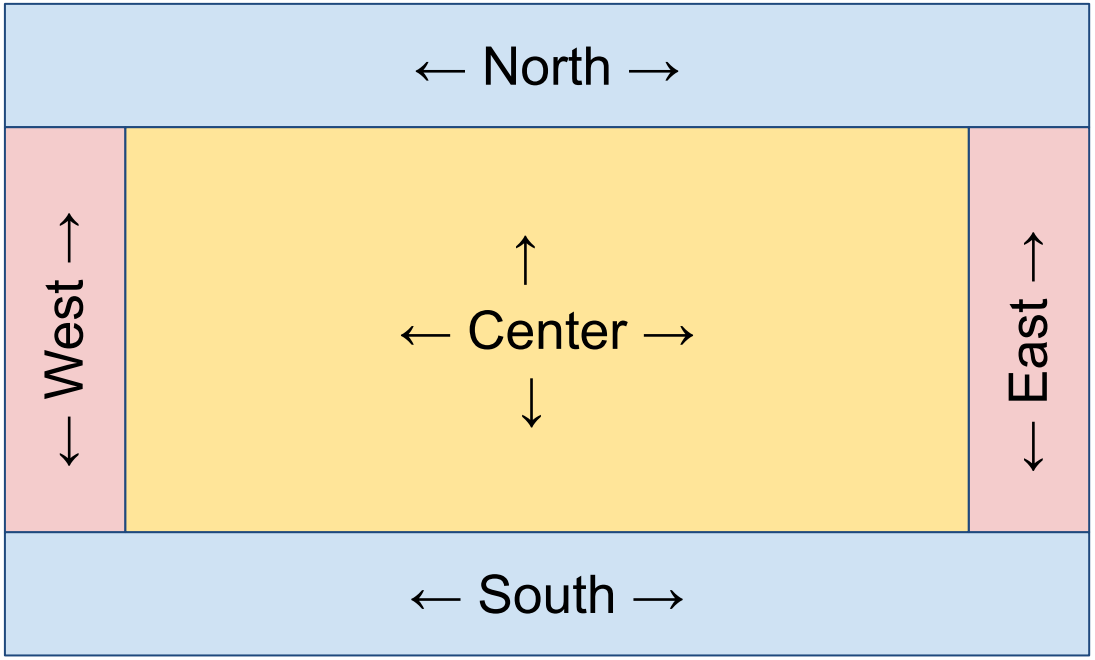
Penny Desktop v1

- Penny desktop v1 (cont.)
  - See **PennySwing1 app**

```
PennySwing1
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
```

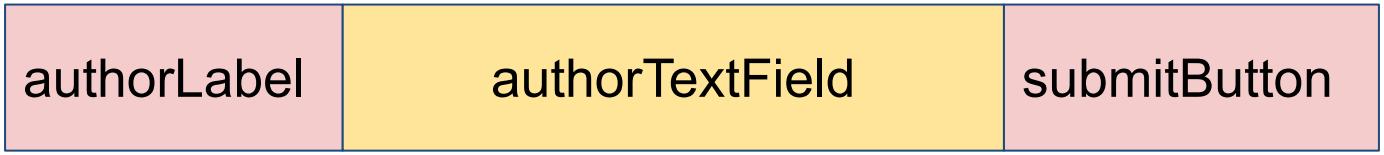
Penny Desktop v1

The Java Swing `BorderLayout` layout manager

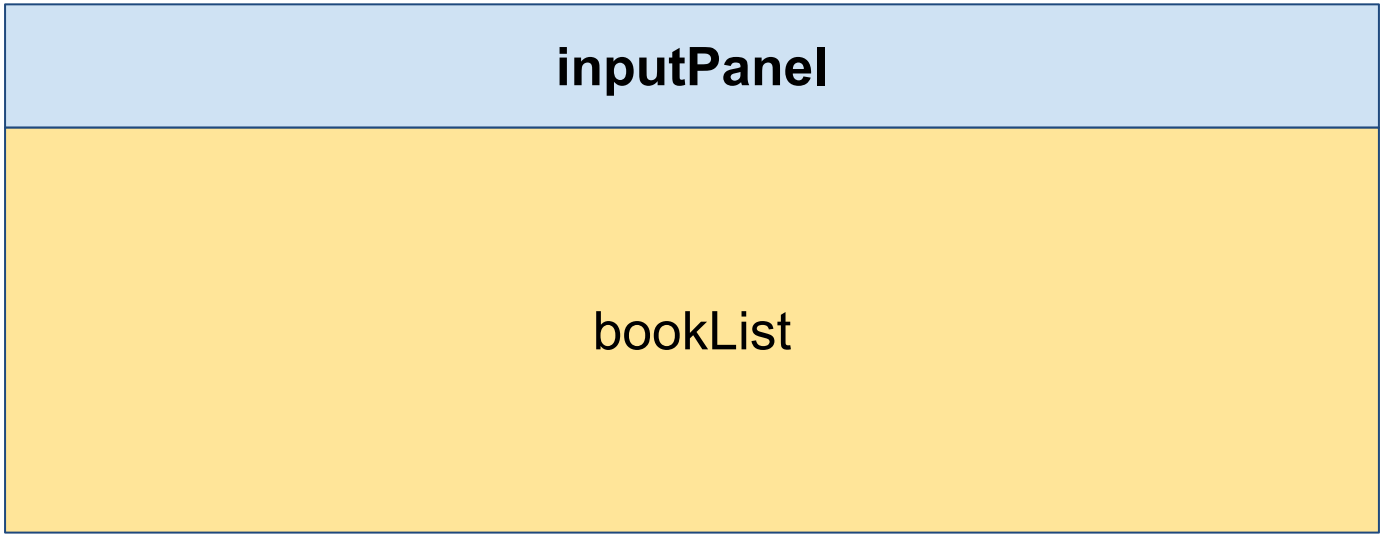


Penny Desktop v1

inputPanel



frame



Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - **Version 2: Immediate**
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Penny Desktop v2

- Recall Penny desktop v1
  - **(maybe) Problem**:
    - Inconsistent window state after typing author and before clicking Submit button
  - **Solution: redesign**...
    - Eliminate Submit button
    - GUI refreshes with each keystroke

Penny Desktop v2

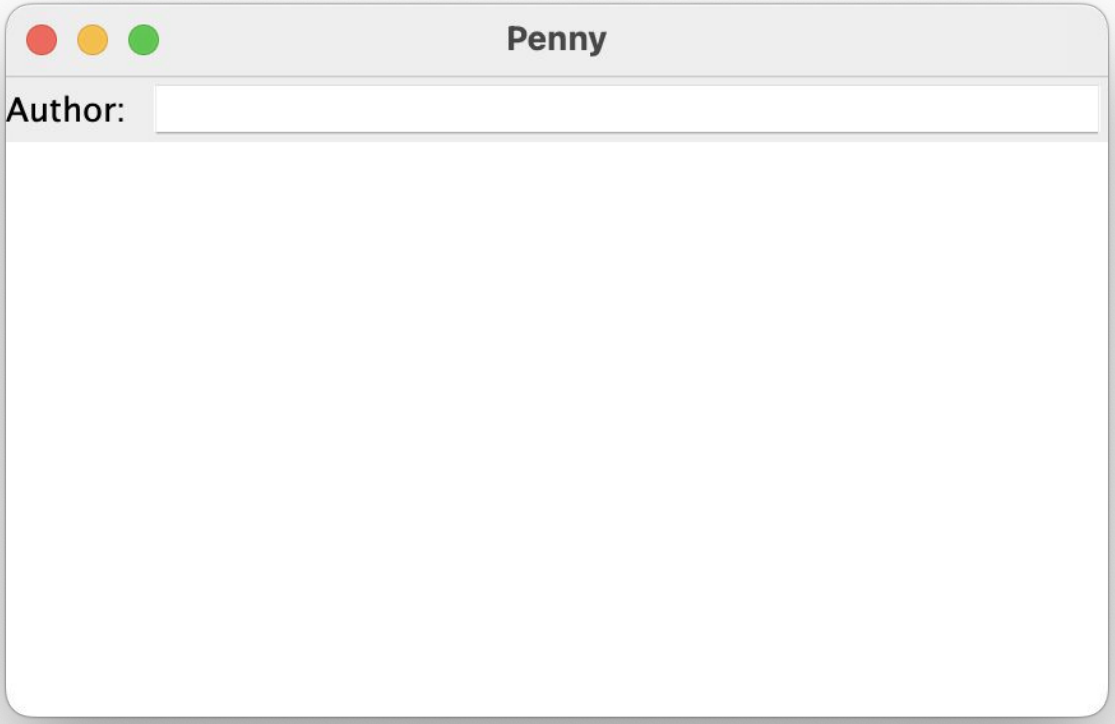
- Penny desktop v2

```
$ export IO_DELAY=1
$ cd PennyJson
$ python runserver.py 55555
```

```
$ cd PennySwing2
$ ./runclient http://localhost:55555
```

Penny Desktop v2

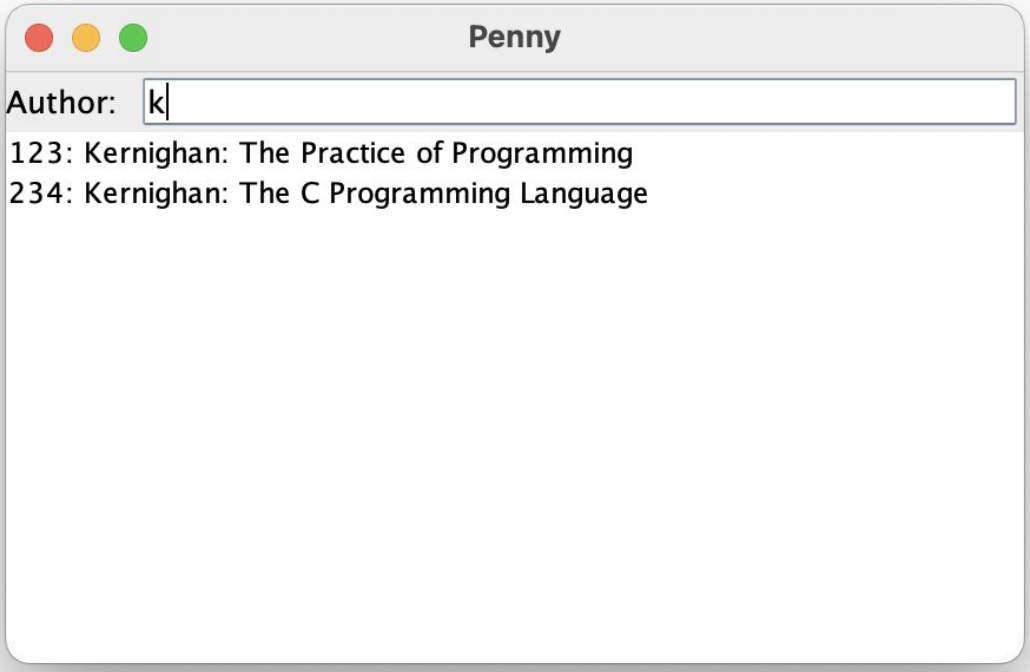
- Penny desktop v2 (cont.)



No Button

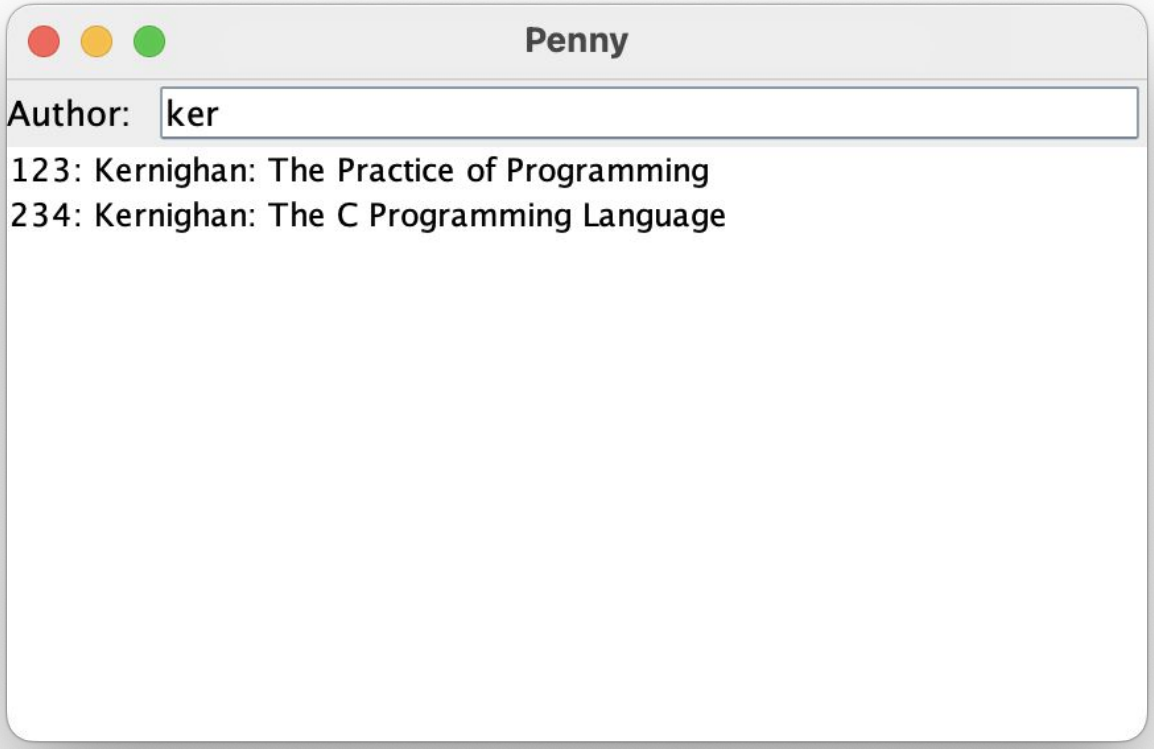
Penny Desktop v2

- Penny desktop v2 (cont.)



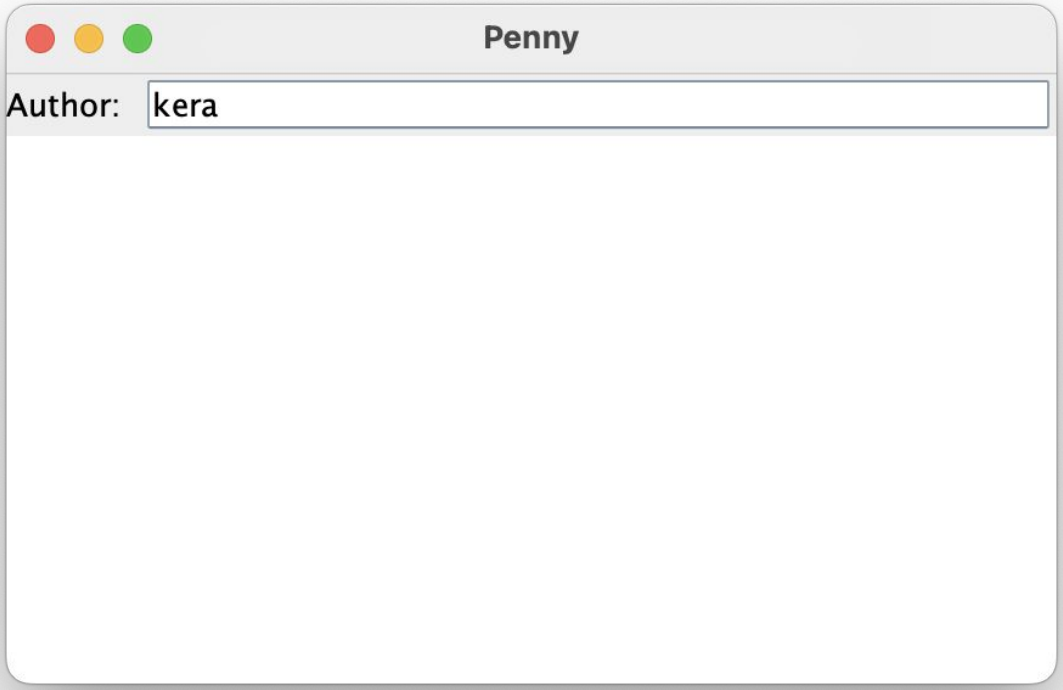
Penny Desktop v2

- Penny desktop v2 (cont.)



Penny Desktop v2

- Penny desktop v2 (cont.)



Penny Desktop v2

- Penny desktop v2 (cont.)
  - See **PennySwing2** app

```
PennySwing2
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
```

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - **Version 3: Threads**
  - Version 4: ThreadsProper
  - Version 5: Stop
  - Version 6: Debounce

Penny Desktop v3

- Recall Penny desktop v2
  - **Problem:**
    - Serious GUI lag
  - **Solution: redesign…**
    - Use threads!!!

Penny Desktop v3

- Penny desktop v3
  - See **PennySwing3** app

```
PennySwing3
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
                     WorkerThread.java
```

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - **Version 4: ThreadsProper**
  - Version 5: Stop
  - Version 6: Debounce

Penny Desktop v4

- Recall Penny desktop v3
- **Problem**
  - Worker thread updates GUI widgets
  - Only GUI thread should update GUI widgets
- **Solution**
  - Worker thread asks GUI thread to update GUI widgets

Penny Desktop v4

- Penny desktop v4
  - See **PennySwing4** app

```
PennySwing4
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
                     WorkerThread.java
```

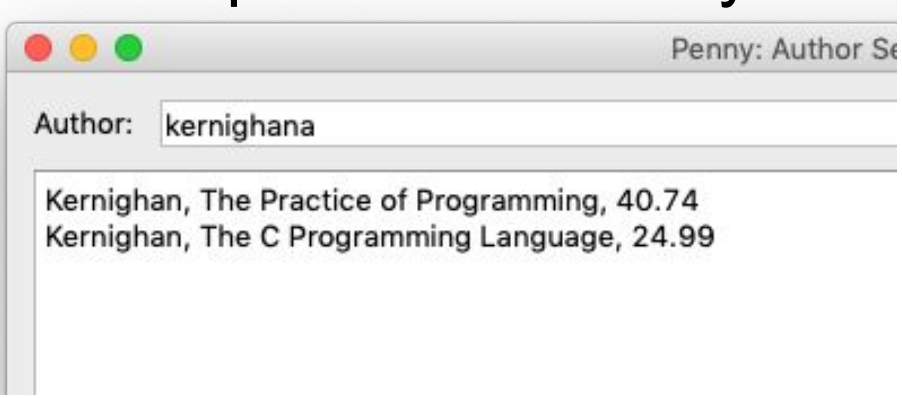
Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - **Version 5: Stop**
  - Version 6: Debounce

Penny Desktop v5

- Recall Penny desktop v4
- **Problem:**
  - Server will respond to requests in arbitrary order

old screen shot



- **Solution:**
  - Abort previous request

Penny Desktop v5

- Penny desktop v5 (cont.)
  - See **PennySwing5** app

```
PennySwing5
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
                     WorkerThread.java
```

Agenda

- Aside: Function def expressions
- Aside: Multithreaded programming
- Java Swing
- Baseline
- **Penny desktop client**
  - Version 1: Simple
  - Version 2: Immediate
  - Version 3: Threads
  - Version 4: ThreadsProper
  - Version 5: Stop
  - **Version 6: Debounce**

Penny Desktop v6

- Recall Penny desktop v5
- **Problem**:
  - Server could be overwhelmed with requests
- **Solution:**
  - ***Debounce*** the requests
- **Bonus:**
  - Reduces (but does not eliminate) the need to abort old requests!

Penny Desktop v6

- Penny desktop v6 (cont.)
  - See **PennySwing6** app

```
PennySwing6
   pom.xml
   runclient
   runclient.bat
   src
      main
         java
            edu
               princeton
                  penny
                     PennyDesktop.java
                     PennyDesktopRunnable.java
                     WorkerThread.java
```

Summary

- Note:
  - If a web application delivers JSON (instead of HTML), then...
  - The client reasonably can be:
    - A browser (running HTML/JavaScript code delivered by the server)
    - A desktop client

Objectives

So far:

Client	Browser + HTML + JavaScript Java + Swing
Server	Python + Flask Java + Servlets Java + Spring JavaScript + Express

Summary

- We have covered:
  - Desktop programming
  - A Penny **desktop** client
    - A desktop client that works with the Penny server
- See also:
  - **Appendix 1**: Java Multithreaded Programming

Appendix 1:
Java Multithreaded Programming

Java Multithreaded Programming

- Recall **race.py**

```
$ python race.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python race.py
1
2
3
4
-1
5
6
-3
-5
-7
-9
7
8
9
10
Final balance: 10
$
```

```
$ python race.py
1
2
3
4
5
6
7
8
9
6
4
10
2
0
-2
Final balance: -2
$
```

Java Multithreaded Programming

- See **Race.java**

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
-6
-8
9
10
Final balance: 10
$
```

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
9
10
-6
-8
Final balance: -10
$
```

Java Multithreaded Programming

- Recall **lockinuser.py**

```
$ python lockinuser.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python lockinuser.py
1
2
3
4
2
0
-2
-4
-6
-5
-4
-3
-2
-1
0
Final balance: 0
$
```

Java Multithreaded Programming

- See **LockInUser.java**

```
$ java LockInUser
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ java LockInUser
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- Recall **lockinresource.py**

```
$ python lockinresource.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python lockinresource.py
1
2
3
1
-1
-3
-5
-7
-6
-5
-4
-3
-2
-1
0
Final balance: 0
$
```

Java Multithreaded Programming

- See **LockInResource.java**

```
$ java LockInResource
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ java LockInResource
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- Recall **conditions.py**

```
$ python conditions.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python conditions.py
1
2
3
4
5
3
1
2
3
4
5
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- See **Conditions.java**

```
$ java Conditions
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 10
$
```

```
$ java Conditions
1
2
3
4
2
0
1
2
3
4
5
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- Multithreaded programming across languages

	Language Support	Library Support
Python	no	yes
C	no	yes
Java	yes	yes
JavaScript	no	no