

# Python Odds & Ends

Dec 2, 2015

# Today

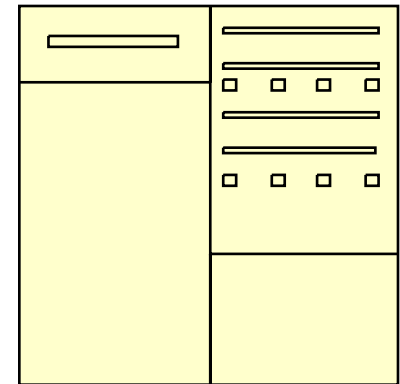
- Web Interaction and Forms
- Graphical User Interfaces
- Final Project Workshop

# Today

- **Web Interaction and Forms**
- Graphical User Interfaces
- Final Project Workshop

# Basic Web Interaction, ~1995

http://www.weather.com/pvd



The Weather Channel

[The Weather Channel](#)  
[City of the Day](#)

[Focus in on  
Local Radar](#)

[New! The Idea  
Exchange!](#)

[Latest on Marco](#)



[What's Your  
Temperature?](#)

[The Global Picture](#)

```
<HTML>
<HEAD>
...
</HEAD>
...
</HTML>
```

Check out the latest on rain and snow with local radar updated every 30 minutes! [Go to a state](#) and choose local radar for a city.

 Pick :

# Basic Web Interaction, 2015



<HTML>  
<SCRIPT>

javascript program here,  
which is executed

</SCRIPT>

<HEAD>

**Google Results**

</HEAD>

<BODY>

**Text, text, text, text**

</BODY>...

</HTML>

<HTML>

**<SCRIPT>**

**javascript program here**

**</SCRIPT>**

<HEAD>

</HEAD>

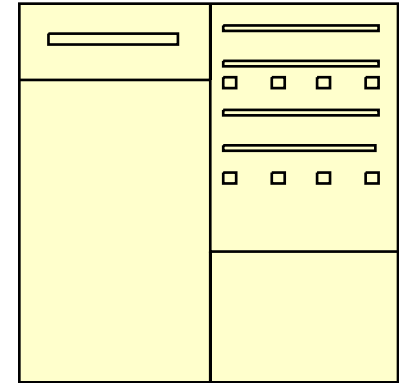
<BODY>

</BODY>...

</HTML>

# Interaction Through Forms

<http://www.azlyrics.com/>



F G H I J K L M N O P Q R S T

**Welcome to AZLyrics!**

It's a place where all searches end!

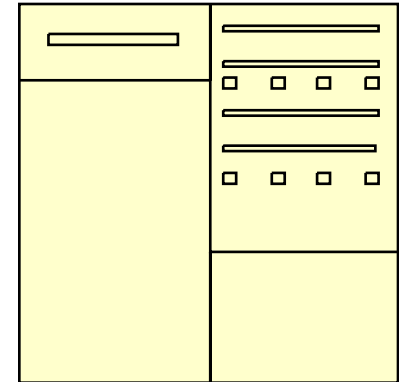
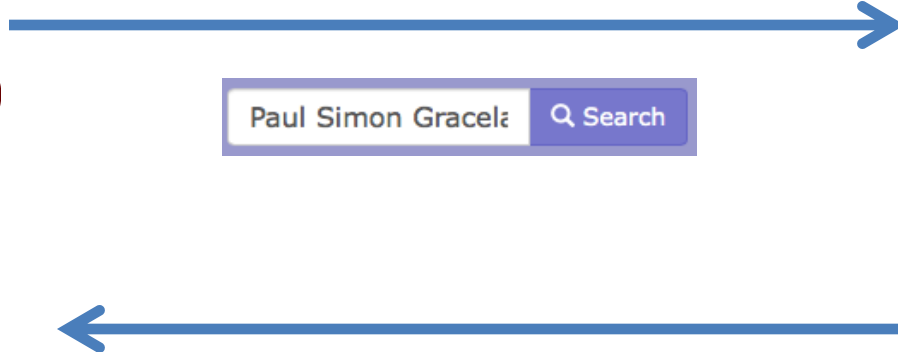
We have a large, legal, every day growing universe of lyrics where stars of all genres and ages shine.

to AZLyrics!

...

# Fancier web Interaction

http://search.azlyrics.com/search.php?q=Paul+Simon+Graceland



**Song results:**  
[1-1 of 1 total songs found]

1. **Graceland** by **PAUL SIMON**  
The Mississippi Delta was shining Like a National guitar I am following the river D  
war I'm going to **Graceland Graceland** In Memphis Tennessee I'm going to **Gra**  
are going to **Graceland** My traveling ...

```
<HTML>  
<HEAD>  
  <ul>  
    <li>Simon  
  </li>  
</ul>  
<BODY>  
  <p>Graceland...</p>  
</BODY>...  
</HTML>
```

# Details

- “Inspect Element” on the search box on [www.azlyrics.com](http://www.azlyrics.com)

```
<form class="navbar-form navbar-right search" method="get" action="http://search.azlyrics.com/search.php" role="search">
  <div class="input-group">
    <input type="text" class="form-control" placeholder name="q" id="q">
    <span class="input-group-btn">...</span>
  </div>
</form>
```



Expand this

- Look at the call to “search.azlyrics.com”
  - With input parameter “q” being the song name



# Details

- “Inspect Element” on the search box on [www.azlyrics.com](http://www.azlyrics.com)

```
<form class="navbar-form navbar-right search" method="get" action="http://search.azlyrics.com/search.php" role="search">
  <div class="input-group">
    <input type="text" class="form-control" placeholder name="q" id="q">
  <span class="input-group-btn">
    <button class="btn btn-primary" type="submit">
      <span class="glyphicon glyphicon-search"></span>
      " Search"
    </button>
  </span>
```

- Look at the call to “search.azlyrics.com”
  - With input parameter “q” being the song name

# Equivalent Python Code

```
import urllib.parse
import urllib.request

url = 'http://search.azlyrics.com/search.php?'
params = {'q': 'Paul Simon Graceland'}

paramsEncode = urllib.parse.urlencode(params)

remoteFile= urllib.request.urlopen(url + paramsEncode)
contents = remoteFile.read()
remoteFile.close()
```

# Results!

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns=http://www.w3.org/1999/xhtml...

<head>
<title>AZLyrics - Search</title>
...
<b>Song results:</b>
<small>[1-1 of 1 total <span class="text-lowercase">Songs</span> found]</small>
<td class="text-left visitedlyr">
1. <a href="http://www.azlyrics.com/lyrics/paulsimon/graceland.html">Graceland</a>
by <b>PAUL SIMON</b>
<div>
The Mississippi Delta was shining
Like a National guitar ...
```

*Which is displayed as...*

# Results!

**Song results:**  
[1-1 of 1 total songs found]

1. **Graceland** by **PAUL SIMON**  
The Mississippi Delta was shining Like a National guitar I am following the river D  
war I'm going to **Graceland Graceland** In Memphis Tennessee I'm going to **Gra**  
are going to **Graceland** My traveling ...

*OK, now we grab the song using the link to it*

# Method

- Use *lxml* or equivalent to transform the string you read into an “XML Tree” and...
  - Use an *xpath* to grab the piece you want
  - Or grab a specific node, and then walk around its children
- Tougher for simple cases, really helpful for complex ones
- You can do this (*Lecture 3-3*)

# Today

- Web Interaction and Forms
- **Graphical User Interfaces**
- Final Project Workshop

# Graphical User Interfaces

- This is the way most of us *interact* with the computer
  - Windows containing text forms and buttons
  - Dialog boxes
- Not good for everything:
  - Say you want to convert files from PPT to PDF
    - But you have 10000 of them
    - Better make a Python program, right?
- But great for most cases

# easygui

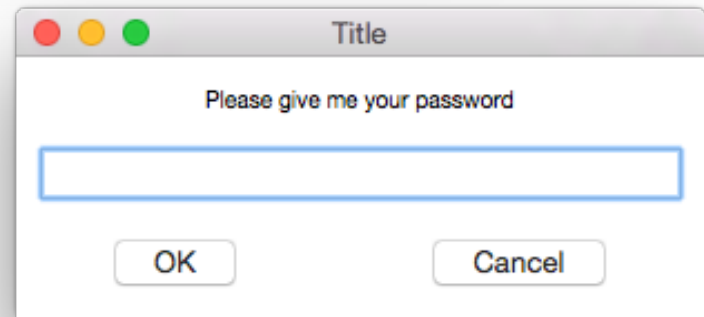
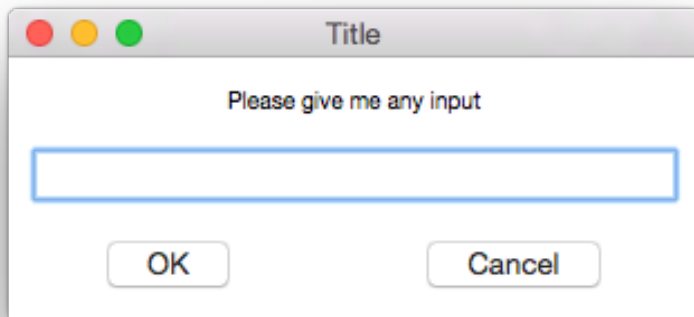
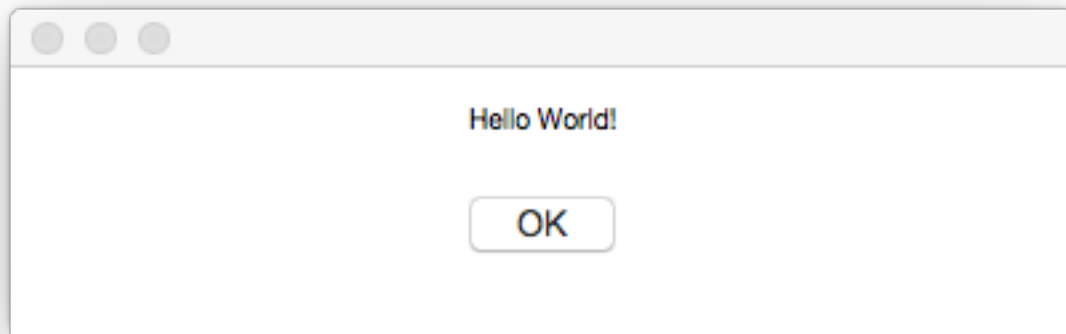
- Python module for... well... easy... GUIs
- To install on the MacOS
  - Open your “Terminal”
    - `“sudo easy_install easygui”`
- To install on Windows
  - Open your “Command Prompt”
    - `“cd \Python27\Lib\site-packages`
    - `“.\easy_install easygui”`



# Basic Boxes

```
from easygui import *  
  
# Message Box  
msgbox("Hello World!")  
  
# Enter Box  
message = "Please give me any input"  
title = "Title"  
something = enterbox(message, title)  
  
# Password Box  
message = "Please give me your password"  
title = "Title"  
password = passwordbox(message, title)
```

# Basic Boxes



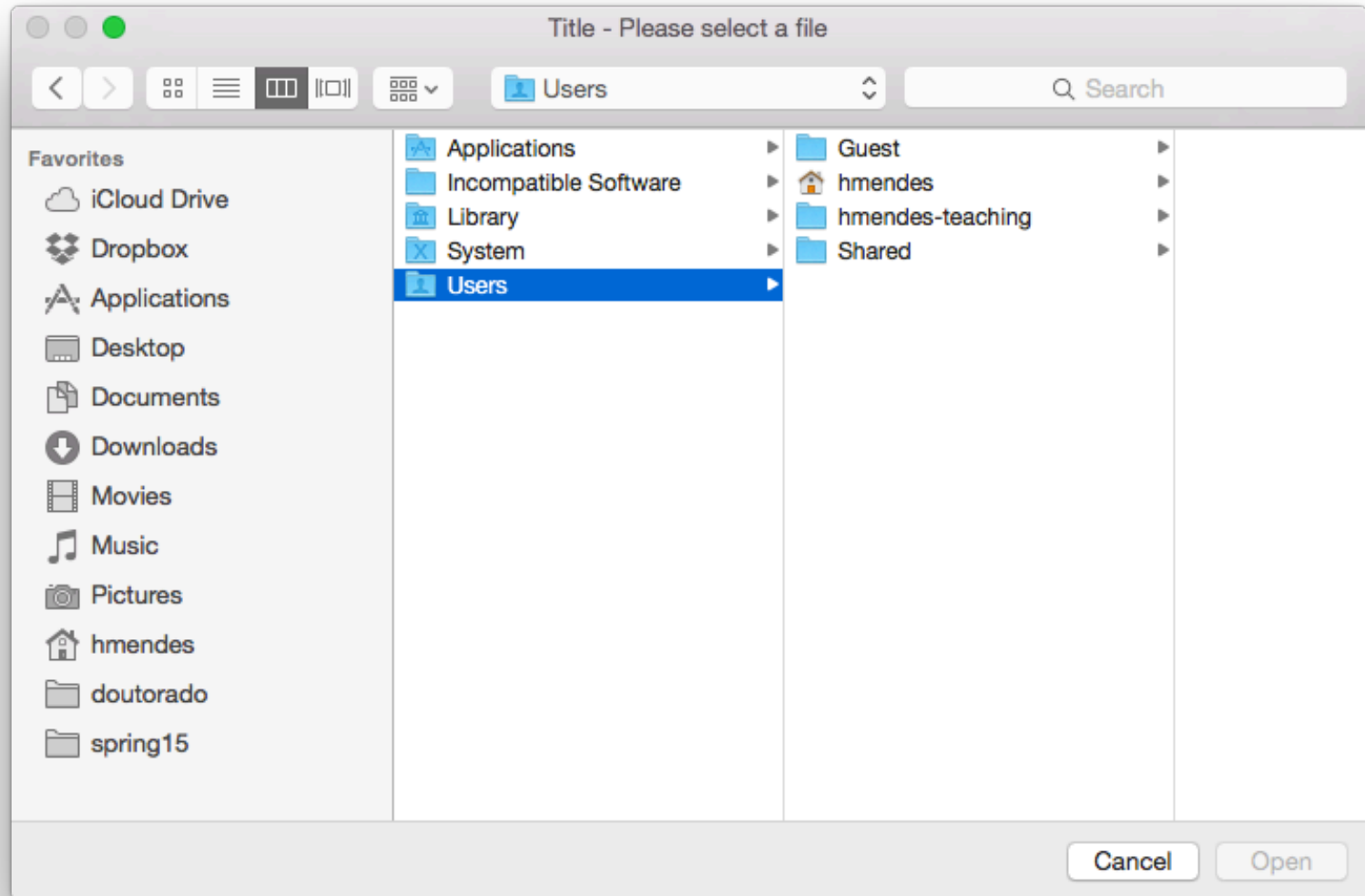
# A Fancier Way to Select Files

```
# File Box
message = "Please select a file"
title = "Title"

filetypes = filetypes = ["*.txt", "*.htm", "*.html"]
filename = fileopenbox(message, title, "*", filetypes)

print(filename)
```

# A Fancier Way to Select Files



# Choices & Questions

## # Choice Box

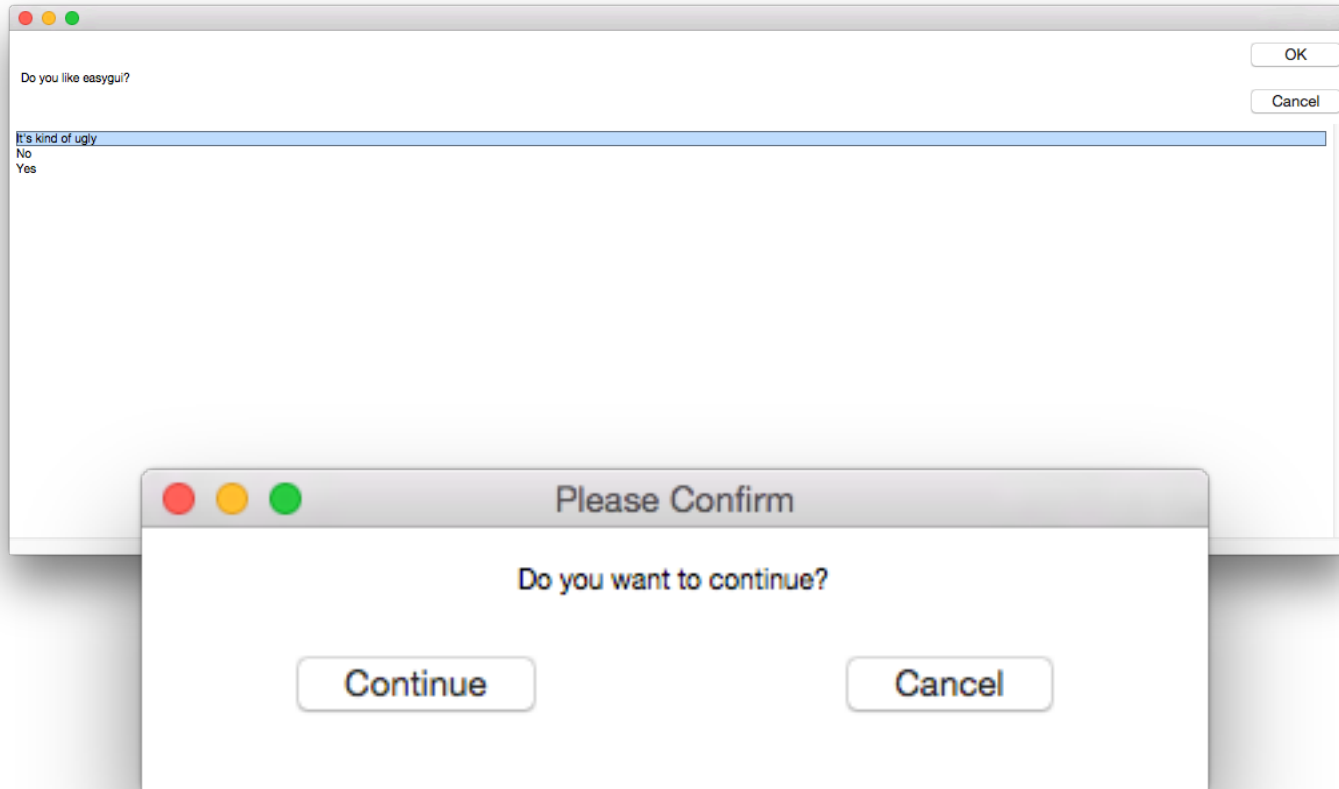
```
choices = ["Yes", "No", "It's kind of ugly"]
reply = choicebox("Do you like easygui?", choices=choices)

print(reply)
```

## # Continue/Cancel Box

```
message = "Do you want to continue?"
title = "Please Confirm"
if ccbox(message, title):
    print("Do actions for continue")
else:
    print("Do actions for cancel")
```

# Choices & Questions



# User Input Forms

```
message = "Enter your personal information"  
title = "Credit Card Application"  
fieldNames = ["Name", "Street Address", "City", "State", "ZipCode"]  
fieldValues = []  
fieldValues = multenterbox(message ,title, fieldNames)  
  
print("Here are the captured fields:", fieldValues)
```

# User Input Forms

Credit Card Application

Enter your personal information

Name

Street Address

City

State

ZipCode

OK Cancel



# User Input Forms

```
message = "Enter your personal information"
title = "Credit Card Application"
fieldNames = ["Name", "Street Address", "City", "State", "ZipCode"]
fieldValues = []
fieldValues = multenterbox(message ,title, fieldNames)

print "Here are the captured fields:", fieldValues
```

*But we still have to **validate** the data!*

```
message = "Enter your personal information"
title = "Credit Card Application"
fieldNames = ["Name", "Street Address", "City", "State", "ZipCode"]
fieldValues = []
fieldValues = multenterbox(message ,title, fieldNames)

# Keep asking for things until all the fields have been filled
if fieldValues == None:
    done = True
else:
    done = False

while not done:
    # Check for empty fields
    someIsEmpty = False
    for i in range(len(fieldNames)):
        if fieldValues[i].strip() == "":
            someIsEmpty = True
    if someIsEmpty:
        fieldValues = multenterbox("Please fill all fields",
                                   title, fieldNames, fieldValues)

        if fieldValues == None:
            done = True
    else:
        done = True
Print("Here are the captured fields:", fieldValues)
```

# There are *so many* alternatives...

- Other GUI modules
  - PyGUI
  - WxPython
  - PyQt
- Some GUI modules use “native” windows from your operating system
  - That is, they look exactly as you’d expect