Regular Expression Basics

March 15, 2012

What We've Learned

TYPES
Integers, Floats
Strings, Booleans

What We've Learned

TYPES
Integers, Floats
Strings, Booleans

STATEMENTS
Expressions,
Assignments,
`Print`, `For`, `If`

What We've Learned

TYPES
Integers, Floats
Strings, Booleans

STATEMENTS
Expressions,
Assignments,
`Print`, `For`, `If`

Data Structures
Lists, Dictionaries

What We've Learned

TYPES
Integers, Floats
Strings, Booleans

STATEMENTS
Expressions,
Assignments,
`Print`, `For`, `If`

Data Structures
Lists, Dictionaries

Function Definitions

What We've Learned

TYPES
Integers, Floats
Strings, Booleans

STATEMENTS
Expressions,
Assignments,
`Print`, `For`, `If`

Data Structures
Lists, Dictionaries

Function Definitions

Files

Review: Python Dictionaries

- REMEMBER: Keys Are Unique
- REMEMBER: Dictionaries are Unordered

Function/Syntax	Input	Output	Example
`keys()`	None	List of keys	`>>> freq.keys()` `['the', 'cat']`
`values()`	None	List of values	`>>> freq2.values()` `[3, 2]`
`<key> in <dict>`	None	Boolean	`>>> 'the' in freq2` `True`
`del(<dict>[<key>])`	Dict. Entry	None	`>>> del(freq2['cat'])`

Making Things **Interactive**

- Run `echo()`

```
def echo():
    myInput = raw_input('write something… ')
    print 'You wrote:',myInput
    return
```

Making Things **Interactive**

- Run `echo()`

```
def echo():
    myInput = raw_input('write something… ')
    print 'You wrote:',myInput
    return
```

Do Task 2

Break: Project 1

Go to the Projects page on the class website

- Make sure that your website shows up properly.
- Look through a few other people's projects

On to Regular Expressions

Flexible ways to match strings in text

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.
- Find all words that begin with the letter `g`

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.
- Find all words that begin with the letter `g`
- Find words that begin with `a` OR `g` and end in `ing`

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.
- Find all words that begin with the letter `g`
- Find words that begin with `a` OR `g` and end in `ing`
- Find words with one or more consecutive `n`'s

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.
- Find all words that begin with the letter `g`
- Find words that begin with `a` OR `g` and end in `ing`
- Find words with two or more consecutive `n`'s
- Many, *many* more questions

On to Regular Expressions

Flexible ways to match strings in text

- Find all occurrences of the string `hello`.
- Find all words that begin with the letter `g`
- Find words that begin with `a` OR `g` and end in `ing`
- Find words with two or more consecutive `n`'s
- Many, *many* more questions

Use special Characters! Remember what '\' means?

Test String for Regular Expressions

- Run `nameGame` with your name

Test String for Regular Expressions

- Run `nameGame` with your name
  - Names that start with vowels vs. consonants

Test String for Regular Expressions

- Run `nameGame` with your name
  - Names that start with vowels vs. consonants

Do Task 3

Test String for Regular Expressions

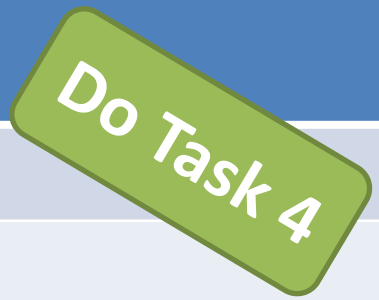
- Run `nameGame` with your name
  - Names that start with vowels vs. consonants

Special Syntax	Meaning
[ ]	Match ANYTHING between brackets
\w	Match any letter
\s	Match any whitespace
\d	Match any digit
+	Match **one or more** of the previous things.

Test String for Regular Expressions

- Run `nameGame` with your name
  - Names that start with vowels vs. consonants

Special Syntax	Meaning
[ ]	Match ANYTHING between brackets
\w	Match any letter
\s	Match any whitespace
\d	Match any digit
+	Match **one or more** of the previous things.



Regular Expressions

Just the beginning… see the 'PythonRE' link for lots more:

Pattern	Description
^	Matches beginning of line.
$	Matches end of line.
.	Matches any single character except newline. Using m option allows it to match newline as well.
[...]	Matches any single character in brackets.
[^...]	Matches any single character not in brackets
re*	Matches 0 or more occurrences of preceding expression.
re+	Matches 1 or more occurrence of preceding expression.
re?	Matches 0 or 1 occurrence of preceding expression.
re{ n}	Matches exactly n number of occurrences of preceding expression.
re{ n,}	Matches n or more occurrences of preceding expression.
re{ n, m}	Matches at least n and at most m occurrences of preceding expression.
a| b	Matches either a or b.
(re)	Groups regular expressions and remembers matched text.
…	…