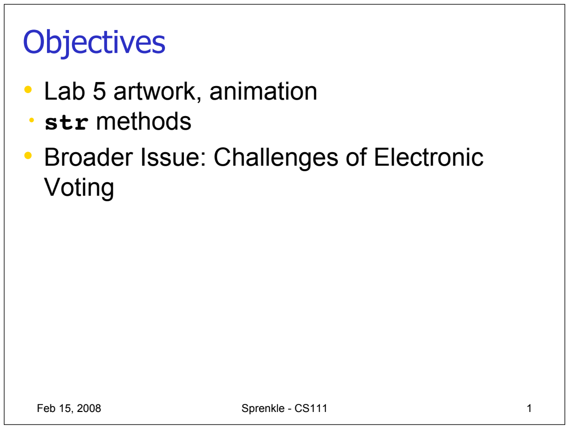
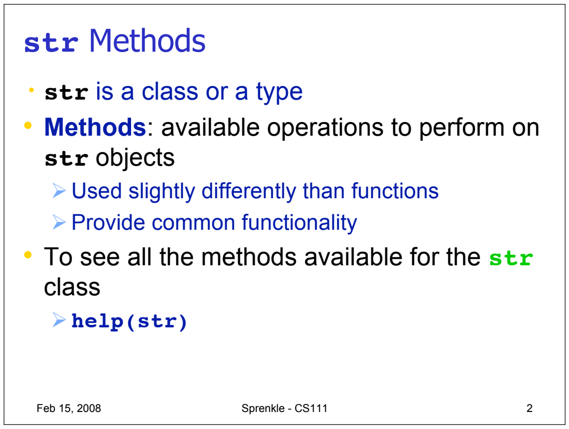


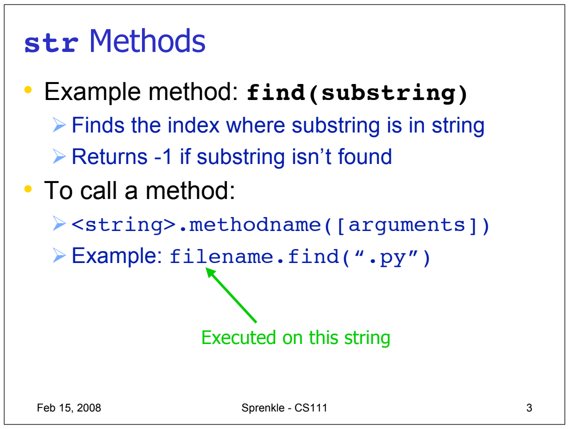
Objectives

- Lab 5 artwork, animation
- **str** methods
- Broader Issue: Challenges of Electronic Voting



str Methods

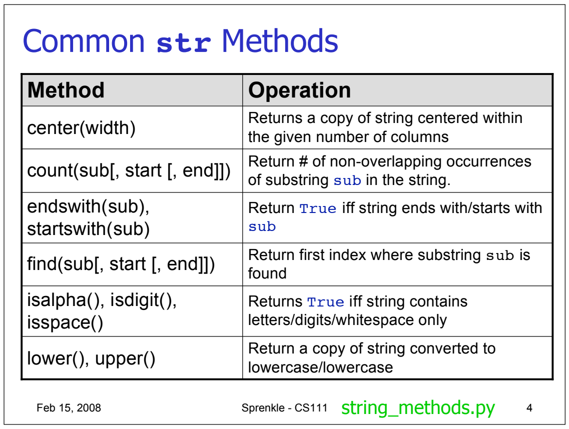
- **str** is a class or a type
- **Methods**: available operations to perform on **str** objects
  - Used slightly differently than functions
  - Provide common functionality
- To see all the methods available for the **str** class
  - **help(str)**



str Methods

- Example method: **find(substring)**
  - Finds the index where substring is in string
  - Returns -1 if substring isn't found
- To call a method:
  - `<string>.methodname([arguments])`
  - Example: `filename.find(“.py”)`

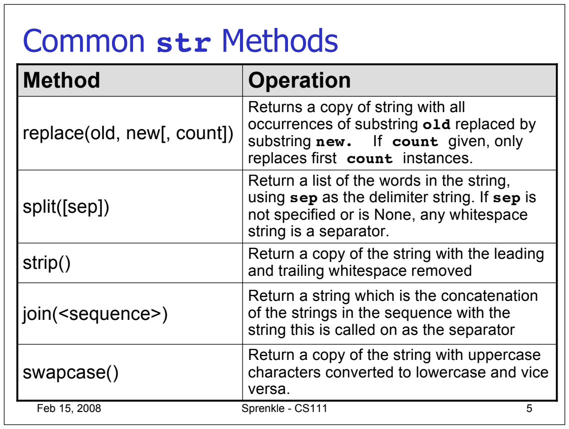
Executed on this string



Common **str** Methods

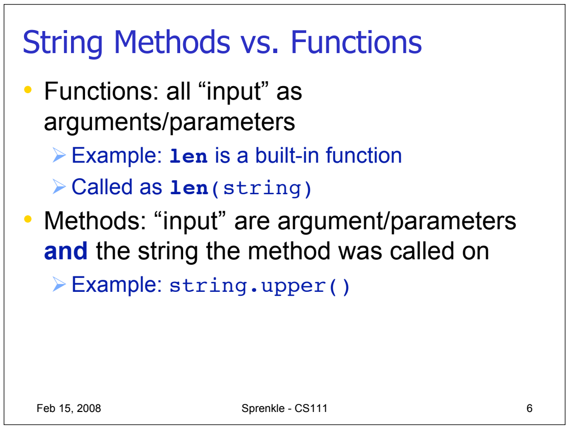
Method	Operation
center(width)	Returns a copy of string centered within the given number of columns
count(sub[, start [, end]])	Return # of non-overlapping occurrences of substring `sub` in the string.
endswith(sub), startswith(sub)	Return `True` iff string ends with/starts with `sub`
find(sub[, start [, end]])	Return first index where substring `sub` is found
isalpha(), isdigit(), isspace()	Returns `True` iff string contains letters/digits/whitespace only
lower(), upper()	Return a copy of string converted to lowercase/lowercase

string_methods.py



Common **str** Methods

Method	Operation
replace(old, new[, count])	Returns a copy of string with all occurrences of substring **old** replaced by substring **new.** If **count** given, only replaces first **count** instances.
split([sep])	Return a list of the words in the string, using **sep** as the delimiter string. If **sep** is not specified or is None, any whitespace string is a separator.
strip()	Return a copy of the string with the leading and trailing whitespace removed
join(<sequence>)	Return a string which is the concatenation of the strings in the sequence with the string this is called on as the separator
swapcase()	Return a copy of the string with uppercase characters converted to lowercase and vice versa.



String Methods vs. Functions

- Functions: all “input” as arguments/parameters
  - Example: **len** is a built-in function
  - Called as **len**`(string)`
- Methods: “input” are argument/parameters **and** the string the method was called on
  - Example: `string.upper()`

Using **str** Methods

- Modify our search program to find out if the entered string has the .py extension

```
PYTHON_EXT = ".py"

filename = raw_input("Enter a filename: ")

if filename[-(len(PYTHON_EXT)):] == PYTHON_EXT:
        # Appropriate output
if PYTHON_EXT in filename:
        # Appropriate output
```

Are You Smarter Than a 5th Grader?

- Problem in spelling from the show: How many a's are in abracadabra?
  - Solve using **str** methods

Using **str** Methods

- Modify binaryToDecimal.py to verify that the entered string contains only numbers

- 2nd modification: How could we make sure that it contains only 0s and 1s?

Implementing Wheel of Fortune

- Simplifications: no money, no buying vowels, no keeping track of previous guesses, one player
- Functionality
  - Displaying puzzle appropriately
  - Gets guesses from user
    - Either letters or solve the puzzle
  - Reports number of the guess in the puzzle
  - Displays puzzle with guesses filled in
- Think about …
  - User input robustness?
  - Any special cases?

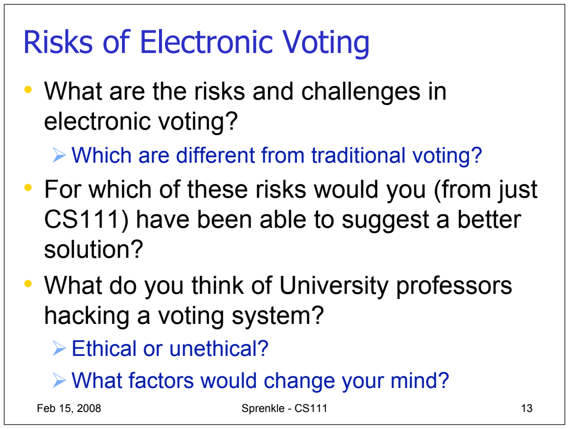
wheeloffortune.py

Extra Credit Opportunity

- Modify Wheel of Fortune to …
  - Include prize money
    - Spinning wheel
  - Handles buying vowels vs consonants
  - Multiple players, losing turns
- Due Friday, February 29

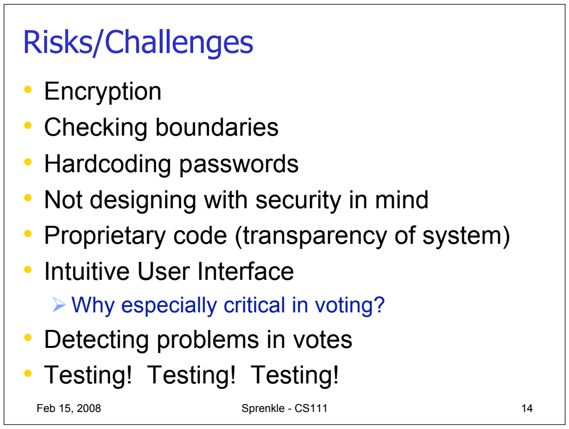
Broader Issue: Electronic Voting

- Select from one of three articles
  - The Risks of Electronic Voting
  - Princeton Scientists Create Vote-Stealing Program
  - Voting with (Little) Confidence



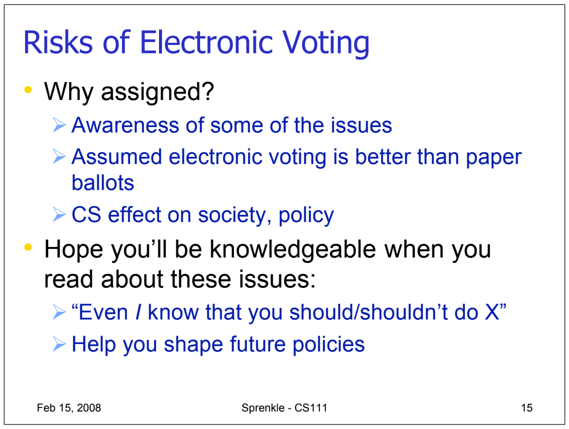
Risks of Electronic Voting

- What are the risks and challenges in electronic voting?
  - Which are different from traditional voting?
- For which of these risks would you (from just CS111) have been able to suggest a better solution?
- What do you think of University professors hacking a voting system?
  - Ethical or unethical?
  - What factors would change your mind?



Risks/Challenges

- Encryption
- Checking boundaries
- Hardcoding passwords
- Not designing with security in mind
- Proprietary code (transparency of system)
- Intuitive User Interface
  - Why especially critical in voting?
- Detecting problems in votes
- Testing! Testing! Testing!



Risks of Electronic Voting

- Why assigned?
  - Awareness of some of the issues
  - Assumed electronic voting is better than paper ballots
  - CS effect on society, policy
- Hope you'll be knowledgeable when you read about these issues:
  - "Even *I* know that you should/shouldn't do X"
  - Help you shape future policies