Strings and Objects
Logistics

- Second Midterm is on April 12.
- Will cover all material up to now.
- Your midterm grades have been released, including up through assignment 5.
- Must have C or better to continue with Computer Science courses.
- Friday is last day to withdraw from class.
String Functions
String Functions

- We already know a few functions that can be used in conjunction with Strings

- For example, the `len()` function can be used to count the # of characters in a String

- As you can imagine, there are other functions that you can use when working with Strings to make things easier for you as a programmer!
Getting the largest and smallest character in a string

You can use two built in Python functions to obtain the maximum and minimum characters in a string (based on their ASCII codes) – Example:

```python
a = max("python")
b = min("python")

print ("max: ", a)
print ("min: ", b)
```

```plaintext
>> y
>> h
```
Slicing a String
Sometimes you may find it necessary to extract a portion of a string from another string.

You can use “slicing” notation in Python to extract a span of characters from a string into a new string. We call this new String a "substring". For example:

```python
full_name = "John Smith"
first_name = full_name[0:4]

print (first_name)

>> John
```
String Slicing Notation

- When you ask Python to slice a string you need to use bracket notation to specify the index range of the characters you wish to extract.

- The syntax for this notation is as follows:

  \[
  \text{substring} = \text{bigstring}[\text{start} : \text{end} : \text{step}]
  \]

- You must supply at least a start or an ending index value.

- Substrings contain all characters starting at the start value specified and continue up to (but do not include) the ending value.

- Omitting a starting or ending index value will cause Python to assume you want to start at the beginning of the string (if you omit a start value) or you want to continue slicing to the end of the string (if you omit the end value).

- This should look a lot like the range function!
String Slicing Notation

What will the following code print?

```python
word = "Superman sings in the shower."

print (word[0:8])
print (word[9:14])
print (word[:5])
print (word[9:])
print (word[-7:])
print (word[0:len(word):3])
```
String Slicing Notation

What will the following code print?

```python
word = "Superman sings in the shower."

print (word[0:8])  # Superman
print (word[9:14])  # sings
print (word[:5])    # Super
print (word[9:])    # sings in the shower.
print (word[-7:])   # shower.
print (word[0:len(word):3])  # Seasgit or
```
Programming Challenge: Username Generation

- You just accepted a position at NYU’s ITS department and your first task is to write a program that generates student Net IDs for all incoming freshmen.

- Net IDs are generated as follows:
  - The first two characters of a student’s first name
  - The first two characters of a student’s last name
  - The last three characters of a student’s N#

- Write a program that asks the user for these three pieces of information (first name, last name, and N#) and generate their Net ID.

- Note that if a student’s first name or last name is less than 2 characters then you should use the entire first name or last name.
String Operators
String Operators

- We already know that the "+" and "*" operators can be used in conjunction with a String.
- The "+" operator can be used to "concatenate" two Strings together.
- The "*" operator can be used to repeat a String a certain number of times (specified as an integer).
Testing Strings with in and not in

- The "in" operator is a Boolean operator that you can use to test to see if a substring exists inside of another string. Example:

```python
word = "Jackson James John Chris Tom"

if "Chris" in word:
    print ("found him!")
else:
    print ("can't find Chris")
```

- When you construct an expression with the “in” operator the result will evaluate to a Boolean
Testing Strings with in and not in

You can also test to see if a string is not in another string by using the “not” keyword in your expression. Example:

```python
word = "Jackson James John Chris Tom"

if "Johnny" not in word:
    print ("No Johnny!")
else:
    print ("Johnny is here!")
```
Data Types and Objects
Data Types

- A type is just a kind or category of values.

- Python needs to know how to set aside memory in your computer based on what kind of information you want to store.

- Python also needs to know what kinds of operations or functions it can perform on your data. Some operations only work on certain types or do different things to different types. Many functions require specific types as arguments.
  - Is your statement or expression allowed?
  - What will it do?
Objects

- For our purposes, we can think of objects like types: they represent a specific kind of data, and have specific operators and functions that can be used with them.

- One kind of function we use with objects is called a "method". It operates directly on a value that is a particular kind of object (or type).
  
  ```
  obj.methodname()
  obj.methodname(arguments)
  ```

- A method means nothing by itself, it can only be called on an object.
String Methods
String Testing Methods

- String testing methods allow you to determine if certain patterns existing within a given string variable. For example:

```python
word = '1234'
if word.isdigit() == True:
    print ("All chars in word are digits!")
else:
    print ("Not all chars in word are digits!")
```

- In the above example we are calling the “isdigit” method on the string variable “word”. This method returns True if all characters contained in this string are numeric digits (0-9) and False if not.
## String Testing Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>isalnum</td>
<td>True if all characters are alphanumeric</td>
</tr>
<tr>
<td>isalpha</td>
<td>True if all characters are alphabetic</td>
</tr>
<tr>
<td>isdigit</td>
<td>True if all characters are digits</td>
</tr>
<tr>
<td>islower</td>
<td>True if all alpha characters are lower</td>
</tr>
<tr>
<td>isspace</td>
<td>True if all characters are “whitespace”</td>
</tr>
<tr>
<td>isupper</td>
<td>True if all alpha characters are upper</td>
</tr>
</tbody>
</table>
Programming Challenge: Character Analysis

- Write a program that counts the # of spaces, digits, vowels and consonants in a string that the user inputs. Example:

  >> Enter a phrase: Sally Sells 1000 sea shells.

  >> Spaces: 4
  >> Digits: 4
  >> Vowels: 5
  >> Consonants: 12
Modification Methods

- Remember that strings are immutable and cannot be directly changed.

- With that said, they do contain a number of “modification” methods that return new copies of the string that reflect a desired change. For example:

```python
word = "Craig"
newword = word.lower()
print (word)

>> craig
```
## String Modification Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lower()</code></td>
<td>Returns a lowercase version of the string</td>
</tr>
<tr>
<td><code>upper()</code></td>
<td>Returns an uppercase version of the string</td>
</tr>
<tr>
<td><code>rstrip()</code></td>
<td>Removes whitespace at end of string</td>
</tr>
<tr>
<td><code>lstrip()</code></td>
<td>Removes leading whitespace characters</td>
</tr>
<tr>
<td><code>capitalize()</code></td>
<td>Returns a copy of the string with the first character capitalized</td>
</tr>
<tr>
<td><code>title()</code></td>
<td>Returns a copy of the string with the first character of each word capitalized</td>
</tr>
<tr>
<td><code>swapcase()</code></td>
<td>Returns a copy of the string where case is swapped among all alpha characters</td>
</tr>
</tbody>
</table>
Programming Challenge

- Write a program that accepts a phrase from the user
- Strip out any leading or trailing “white space” from the string
- If the string has an even number of characters, make it all lowercase
- If the string has an odd number of characters, make it all uppercase
Searching and Replacing

- Programs often need to perform search and replace functions on data, much like the “find and replace” functionality that exists in your word processor.
Finding substrings

- You can find whether a string exists inside another string by using the find() method. Example:

```python
word = "Like finding a needle in a haystack!"
location = word.find("needle")
print (location)
```

- The find() method will return the index of the first occurrence of a substring within a string.

- If the find() method cannot find the desired substring it will return -1
Programming Challenge: He Who Shall Not Be Named

- Write a program that asks a user to type in a message.

- If they use the word “voldemort” anywhere in their message you should warn them to be more careful!
Replacementsubstrings

You can have Python replace all occurrences of a substring by using the replace() method. Example:

```python
word = "Voldemort had one goal in life - to kill Harry Potter."
newword = word.replace("Voldemort", "He who shall not be named")

print (word)
print (newword)
```

```console
>> Voldemort had one goal in life - to kill Harry Potter.
>> He who shall not be named had one goal in life - to kill Harry Potter.
```
Programming Challenge: Redacted!

- Write a program that redacts all instances of “Python” from the following string:

  “Python is a programming language that is fairly easy to learn. Python is great for beginning and advanced programmers.”

- To produce the following:

  “______ is a programming language that is fairly easy to learn. _____ is great for beginning and advanced programmers.”
Programming Challenge: Pronoun Switcher

- Write a program that replaces the pronoun “his” with the pronoun “her” in the following phrase:

  "Pat said that his score on his history test was 86."
Programming Challenge

- Write a function that counts the number of letters in a string. You should count both uppercase and lowercase letters.
  Example:

```python
x = count_letters("Python is fun")
print(x)
```

>> 11
Programming Challenge

- Write a program that prompts the user to enter in a valid Social Security Number, in the following format:

  NNN-NN-NNNN

  where ‘N’ is a digit. The program should display “Valid SSN” if the supplied string is valid and “Invalid SSN” if it is not.

- Extension: Turn your program into a function