

Objects and methods

- All the data values we've worked with are called *objects*.
- Some are simple things (like ints) and others are more complex (like strs and pictures).
- Each type of object is defined by some code that says what information needs to be stored to represent that type of object, and how.
- Many types of object also define functions that are specific to that type of object. We call these *methods* (not functions).

Calling methods

- Eg, str defines a method "lower" that converts any letters in the string to lower case.

- You call it like this:

```
s = "Hi, CSC108 students!"  
s.lower()
```

- You can't call it like this, because it's not a function:

```
s = "Hi, CSC108 students!"  
lower(s)
```

More about strings

- Strings are immutable.
- Strings are sequences of characters.
- The characters need not be letters. Eg "CSC108, winter 2009!" is a valid string.
- Each character has an *index*, which is its location in the string.
- The first index is 0.
- The string module defines many useful methods. Use `help(str)` and `dir(str.whatever)` to find out more.

Some string methods

- `S.replace(old, new)`: return a string, same as S but with all occurrences of old replaced by new. Does not change S.
- `S.count(substring)`: return the number of times substring occurs in S.
- `S.find(substring)`: return the index of the first occurrence of substring in S, starting from the left.
- `S.startswith(substring)`: return True iff S begins with the substring.
- And a very useful string function: `len(string)`

Why are some things methods and other things functions?

- The designer(s) of Python could have defined len (and find etc.) as a method or a function.
- Programmers can define their own new kinds of objects too (this is taught in csc148).
- The same choice has to be made for every operation you want to define for your new type of object: method or function. There is always a choice.

So how does one decide?

- One guideline: if the operation is only relevant to one type of object, make it a method defined for that type of object.
 - Eg: converting to lowercase.
- But if the operation is relevant to other types of object, make it a function, so it can be called with all of those types of object.
 - Eg: finding the length of something.
- You won't be asked to make these decisions in csc108. But the issue was probably bugging you.

String comparisons

- You can also use the comparison operators on strings.
- The comparison will be made alphabetically.
- For non-alphabetic characters, an ordering is defined. You only need to know that:
 - "a" < "b" < ... < "z"
 - "A" < "B" < ... < "Z"
 - "0" < "1" < ... < "9"
- You don't need to know whether:
 - "<" < "!"