

## Objectives

- String Methods
- String formatting

**Payment details**  
 The total amount that will be debited to your credit card is \$100.36 for 52 weeks, plus \$10.00 for the tip, for a total amount of \$110.36

“The total amount that will be debited to your credit card is \$100.36 for 52 weeks, plus \$10.00 for the tip, for a total amount of \$110.36”

Feb 13, 2017

Sprenkle - CSC1111

1

## Review: String Operations

- How do we represent text?
- How can we represent really long text?
- How can we combine strings?
- How can we combine strings multiple times?
- How can we find out how long a string is?
- How can we find out the character at a certain position?
- How can we iterate through a string?

Feb 13, 2017

Sprenkle - CSC1111

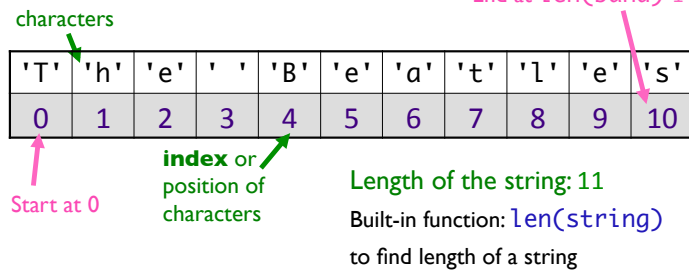
2

## Strings

- A *sequence* of one-character strings

➤ Example:

`band = "The Beatles"`



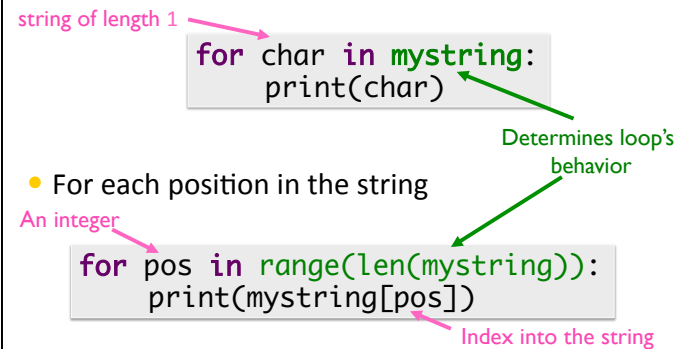
Feb 13, 2017

Sprenkle - CSC1111

3

## Summary: Iterating Through a String

- For each character in the string



Feb 13, 2017

Sprenkle - CSC1111

4

## Revised Pick4 Game

- To play: pick 4 numbers between 0 and 9
- To win: select the numbers that are selected by the magic ping-pong ball machine
- Done previously: Simulate the magic ping-pong ball machines
- Additional Functionality:
  - Determine if the user picks the winning number

Feb 13, 2017

Sprenkle - CSCI

pick4winner.py

5

## USING THE STR API

Feb 13, 2017

Sprenkle - CSCI111

6

## Review

- What is an API?
- How do we call methods on an object?

Feb 13, 2017

Sprenkle - CSCI111

7

## str Methods

- **str** is a *class* or a *type*
- **Methods**: available operations to perform on **str** objects
  - Provide common functionality
- To see all methods available for **str** class
  - `help(str)`


Feb 13, 2017

Sprenkle - CSCI111

8

## 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:
  - `<str_obj>.methodname([arguments])`
  - Example: `filename.find(".py")`


  
Executed on this string

Feb 13, 2017 Sprengle - CSC1111 9

## Common str Methods

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

Feb 13, 2017 Sprengle - CSC1111 `string_methods.py`

## Common str Methods

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

Feb 13, 2017 Sprengle - CSC1111 11

## String Methods vs. Functions

### Functions

- All input comes from arguments/parameters
- Example: `len` is a built-in function
  - Called as `len(strobj)`

### Methods

- Input comes from arguments *and* the string the method was called on
- Example:
  - `strobj.upper()`

Feb 13, 2017 Sprengle - CSC1111 12

## Using the APIs

- Given a problem, break down the problem
  - Can any of the parts of the problem be solved using a method in the API?

Feb 13, 2017

Sprenkle - CSCI111

13

## Are You Smarter Than a 5th Grader?

- Problem in spelling from the show: How many a's are in abracadabra?
  - Solve using **str** methods
- Silly problem but can generalize to other problems
  - How many a's are in a given word?
  - How many of a certain letter are in a given word?

Feb 13, 2017

Sprenkle - CSCI111

14

## Verifying User Input

- How can we verify that the user entered the lottery number in the correct format?

`pick4winner_better_error_handling.py`

Feb 13, 2017

Sprenkle - CSCI111

15

## Usability

- Want users to *like* to use your software
  - More revenue
  - Develop even better software
- How Apple makes money:
  - best user interfaces → user buys products

Feb 13, 2017

Sprenkle - CSCI111

16

## Escape Sequences

- Escape character: `\`
- Escape sequences
  - newline character (carriage return) → `\n`
  - tab → `\t`
  - quote → `\"` or `\'`
  - backslash → `\\`
- Example:
  - `print("To print a \\, you must use \\\"\\\\\\")`
    - What does this display?

Interactive demonstration

Feb 13, 2017

Sprengle - CSC1111

demo\_str.py

17

## Practice

- Display To print a tab, you must use `'\t'`.
- Display I said, "How are you?"

escape\_sequence.py

Feb 13, 2017

Sprengle - CSC1111

18

## Problem with print

- By default, `print` puts spaces in the output whenever there is a comma
- Example:

```
x = 13.54
print("You owe $", x, ". ")
```

Displays:

You owe \$ 13.54 .

Extra spaces

Possible solutions?

Feb 13, 2017

Sprengle - CSC1111

19

## Solution: using print

- Use `sep` parameter

```
x = 13.54
print("You owe $", x, ".", sep="")
```

Any issues with this solution?

Only works if want that separator for *all* separators

Feb 13, 2017

Sprengle - CSC1111

20

## Solution: using `str()`

- Recall: `str()` is constructor/converter function to convert other data types to strings
  - Example: `str(33) → '33'`
- Use constructor with the `+` (i.e., *concatenation*) operator when printing output
  - Gets rid of extra spaces you don't want.

```
print("You owe $" + str(x) + ".")
```

Feb 13, 2017

Sprenkle - CSCI111

21

## Another problem with print

```
SALES_TAX=.053 # the sales tax in VA
value = eval(input("How much does your item cost? "))
with_tax = value * (1+SALES_TAX)
print("Your item that cost $", value, end=' ')
print("costs $", with_tax, "with tax.")
```

Feb 13, 2017

Sprenkle - CSCI111

sales\_tax.py 22

## FORMATTING STRINGS

Feb 13, 2017

Sprenkle - CSCI111

23

## Solution: format Method

- How to use:
  - `"templatestring".format(<whattoformat>)`
- `templatestring` allow us to control how output is displayed to user
  - Right, left justification
  - Number of decimals to display

Feb 13, 2017

Sprenkle - CSCI111

24



### Example using Format Operator

```

    print("Your item that cost ${:.2f}".format(value))
    print("costs ${:.2f} with tax".format(tax))
    
```

Alternative:

```

    print("Your item that cost ${:.2f} costs ${:.2f} with tax".format(value, tax))
    
```

Feb 13, 2017

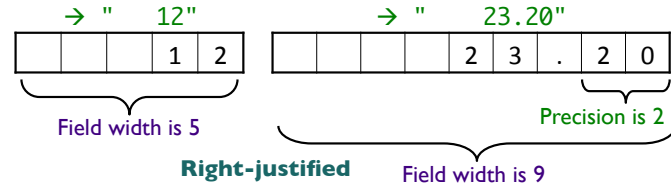
Sprenkle - CSC1111

29

### Example Format Specifiers

```

    "{:5d}".format(12)    "{:9.2f}".format(23.1999)
    
```



- What if precision is bigger than the decimal places?
- What if field width is smaller than the length of the value?

Any guesses? Try out in interpreter.

Feb 13, 2017

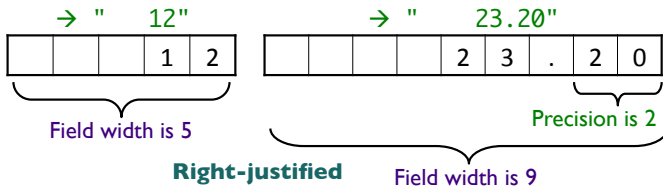
Sprenkle - CSC1111

30

### Example Format Specifiers

```

    "{:5d}".format(12)    "{:9.2f}".format(23.1999)
    
```



- What if precision is bigger than the decimal places?
  - Fills decimal with 0s
- What if field width is smaller than the length of the value?
  - String contains entire value

Feb 13, 2017

Sprenkle - CSC1111

31

### Formatting Practice

- x = 10
- y = 3.5
- z = "apple"
- "{:6d}".format(x)
- "{:6.2f}".format(x)
- "{:06.2f}".format(y)
- "{:6.2f}".format(y)
- "{:^10s}".format(z)
- "{:5d} {:<7.3f}".format(x,y)

Feb 13, 2017

Sprenkle - CSC1111

32



## Example: Printing Out Tables

- A table of temperature conversions

Temp F	Temp C	Temp K
-459.7	-273.1	0.0
0.0	-17.8	255.2
32.0	0.0	273.1

- If we want to print data in rows, what is the template for what a row looks like?

➤ How do we make the column labels line up?

Feb 13, 2017

Sprenkle - CSCI1111

temp\_table.py 33

## Looking Ahead

- Lab 5 tomorrow

Feb 13, 2017

Sprenkle - CSCI1111

34