ECS 15

Strings, input
Outline

- Strings, string operation
- Converting numbers to strings and strings to numbers
- Getting input
- Running programs by clicking on them
Dick said, “See what Mike and I have.
This is Puff with me.
That is Spot with Mike.”

Mike said, “Come here, girls.
See Spot and Puff in a little play.”
Dictionary:

**string**

(strĭng)

*n.*
1. A cord usually made of fiber, used for fastening, tying, or lacing.
2. Something configured as a long, thin line: *limp strings of hair.*
3. A plant fiber.
4. A set of objects threaded together: *a string of beads.*
5. A series of similar or related acts, events, or items arranged or falling in or as if in a line. See synonyms at *series.*
Writing a form letter

- "Parts of speech" in programming
- Using variables and strings
Color: parts of speech

- Dark red – comments! Good idea to start off with a comment describing purpose of program and its overall strategy.
- Orange – Python commands. If you misspell it, it’s not orange.
- Green – strings
- Black – numbers, variables
- Purple – functions
Expressions and values

- 34+57 is an **expression**.
- Its value is 91

- “Coca-cola” is also an expression
- Its value is “Coca-cola”

- Either kind of expression can be assigned to variables.
IDLE vs script window

- Type an expression in IDLE window – it types back the value.
- Type an expression (alone) in script window and the value does NOT appear when you run the program.
- Has no effect unless it’s assigned to a variable, explicitly printed, etc.
Strings and quotes

- “Coca-cola” and ‘Coca-cola’ are two ways of writing the same string.
- “Macy’s” and ‘Macy’s’ are two ways of writing a string containing an apostrophe.
- Can use ‘ or “ to make strings with apostrophes or quotes.
Concatenation

- An **operator** on strings
- “My “+”cat” has the value “My cat”
A few fun string operators

- `a+b`
- `a*i`
- `a.upper()`
- `a.lower()`
- `a.capitalize()`
- `a.count('sub')`
- `a.replace('sub1','sub2',n)`
Different ways to skin a…

- Here are three ways to do the same thing
  - `print ("My cat", catName)`
  - `print ("My cat " + catName)`
  - `print ("My cat", end=' ')`
    - `print(catName)` (this one works only in programs, not in the IDLE window).
Newline

- “\n” produces a carriage return
- Two ways to do the same thing:
  - print ("A rose\nis a rose\nis a rose.")
  - print ("A rose")
    print ("is a rose")
    print (‘is a rose.’)
Quick review

- 15+22 is a number (or numeric) expression.
- It uses **addition**
- Its value is 37

- “drive”+“way” is a string expression
- It uses **concatenation**
- It’s value is “driveway”
Integers vs strings

5 is an integer.

“5” is a string.

Rover is a dog.

“Rover” is a string.
Convert integer to string

- `int("5")` is the integer 5
- `str(5)` is the string "5"
### Functions

- int() and str() are **functions**
- Like in algebra, \( f(x) \).
- Python computes the value of a function, just like it computes the value of an expression.

- The value of int(“5”) is the integer 5.
- The value of str(6) is the string “6”.
Converting numbers

- 5.0 and 6.33 are floating-point numbers
- 5 and 2 are integers
- The value of float(2) is the floating-point number 2.0
- The value of int(6.0) is the integer 6
- The value of int("3.8") is...
Getting input

- Use the function `input()` (used to be `raw_input` in Python 2)
- Example: `color = input("Favorite color: ")`
  - Python evaluates the function `input`...
  - by printing the prompt “Favorite color: “
  - the user types in “red”
  - the string “red” becomes the value of the input function
  - the variable `color` gets the value “red”
The ID program

- First line:
  \[ x = \text{input}("\text{Four-digit number: \"}") \]

- \( x \) gets the string “7554”, not the integer 7554.

- Python gets confused if we try to do arithmetic with a string.

- We need to convert “7554” to an integer
Running program by clicking

- Why does it close up?
Keep window open

- Add final line to program:
  ```python
  input("Type enter to exit: ")
  ```
- We do not do anything with the value of this last input function; we are happy to throw it away!
- Mac and Windows versions of Python are slightly different. The **BEST** way is to try it out!
Handling bad input

- Crazy input causes our program to crash!
- This is very bad.
- We do not have a good way to fix this yet...
Let’s write a program together

- Similar to first Python lab
- Print the first 4 Fibonacci numbers

- Start by create a program file
- Comments
- Define variables
- Statement of calculations
- print