

ECS 15



Strings, input

Outline

- ❑ Strings, string operation
- ❑ Converting numbers to strings and strings to numbers
- ❑ Getting input
- ❑ Running programs by clicking on them
- ❑ Menu program

Announcements

- Lab5 is due next Wednesday, 5pm
- Do it early in case there is a hang-up.

Topic

- Writing a form letter
 - “Parts of speech” in programming
 - Using variables and strings



“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

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"



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",`
`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.



Rover is a dog.

“5” is a string.

“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 `raw_input()`
- Example: `color = raw_input(" Favorite color: ")`
 - Python evaluates the function `raw_input...`
 - by printing the prompt "Favorite color: "
 - the user types in "red"
 - the string "red" becomes the value of the `raw_input` function
 - the variable `color` gets the value "red"



The ID program

□ First line:

```
x = raw_input("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:
`raw_input("Type enter to exit: ")`
- We do not do anything with the value of this last `raw_input` function; we are happy to throw it away!



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...

Variable Names

- Legitimate names:
 - Letter (upper & lower case), number, underscore
 - Do not start with a number
 - Ex: x, y, m1, m2, xysgfh, my_ID, my_age, myAge, etc.
- Good names
 - Easy to remember and to understand the meaning
 - Not too long, not too short
 - Ex: my_age, myAge, etc.