Announcements

- Next assignment due Tues 10/7
- “I turned in the wrong file.”
- New TA - Kartik Pandit

Last Time

- Data types
  - Integers, eg. 87
  - Floating point numbers, eg. 6.78
  - Strings, eg. “Call me a taxi!”
- Variables
- Programs

Integers vs strings

5 is an integer. Fiona is a dog.

‘5’ is a string. ‘Fiona’ is a string.

Variables

- $x = 5$
- $y = x$
- $x = 2$

Assigns the integer 5 to the variable x
The variable is on the left
The value going into the variable is on the right.
Variable is like a box, holding a value
Variables

\[
\begin{align*}
5 & \quad x \\
5 & \quad y \\
y = x
\end{align*}
\]

Why variables?

- One reason – You can assign a value in one place and use it lots of places. So it’s easy to change the value later.

Newline

- "\n" produces a carriage return
- Two ways to do the same thing:
  - \texttt{print "A rose\nis a rose\nis a rose."}
  - \texttt{print "A rose"}
    - \texttt{print "is a rose"}
    - \texttt{print 'is a rose'.}

Concatenation

- An \texttt{operator} on strings
- "hyper"+"reality" is a string expression that has the value "hyperreality"
- "Gillian"+"." has the value "Gillian."
- Addition is an operator on numbers that happens to be written with the same symbol.
- \texttt{2+5} is an arithmetic expression that has the value 7.

Different ways to skin a...

- Here are three ways to do the same thing
  - \texttt{print "My cat","catName}
  - \texttt{print "My cat +catName}
  - \texttt{print "My cat",}
    - \texttt{print catName}
  - (this last one works only in programs, not in the IDLE window. The comma prevents going to a new line).
The user

The person running the program.

Getting data from the user

\[ x = \text{raw_input('Enter a name: ')} \]

- The user enters a string
- The string is assigned to the variable \( x \)

Getting input

- Use the function \text{raw_input()}\n- Example: \text{color = raw_input(" Favorite color: ")}
  - Python evaluates the function \text{raw_input}…
  - by printing the prompt “Favorite color: “
  - the user types in “red”
  - the string “red” becomes the value of the \text{raw_input} function
  - the variable \text{color} gets the value “red”

raw_input() makes a string

\[
>>> x = \text{raw_input("Enter a number: ")}
enter a number: 5
>>> x
'5'
\]

- \( x \) is assigned the string ‘5’
- The only kind of data you can get is a string!

Why variables?

- Reason 1 – You can assign a value in one place and use it lots of places. So its easy to change the value later.
- Reason 2 – you can get a data from the user and refer to it in your program.

Enter to exit

- Many examples in the book end with:
  \[ \text{raw_input("Press enter to exit")} \]
  - Python waits for the user to enter something. As soon as the user hits the enter key, the program is done and it exits.
  - Notice the value produced by \text{raw_input()} is not put into a variable; it is just thrown away.
The empty string

- The variable name contains a string, but it is a string containing no characters.
- It is called the empty string.
- The length of string "Daniel" is 6
- The length of the empty string "" is 0

Checking for the empty string

- Using the if statement:

```python
if name == "":
    name = "Resident"
```

- if condition:
  do some statements

More on this next time!