ECS 10

9/28

Announcements

- Second program assigned, due Tuesday night.
- Based on Campus Community Book Project book.

ECS 10 Commenting Rules

- When you first introduce a variable, add a comment
  - What data type is it (integer, string, floating point)?
  - What is it supposed to contain? What does the data mean?

Variables

\[
\begin{align*}
&X = 5 \\
\end{align*}
\]

\[
\begin{align*}
&y = x \\
&2 \times 5 \\
\end{align*}
\]

\[
\begin{align*}
&x = 2 \\
&5 \times y \\
\end{align*}
\]
Getting user input

```python
inStr = input("Enter celsius temperature: ")
```

- This is an assignment statement.
- `inStr` is a variable.
- The value of the expression:
  ```python
  input("Enter celsius temperature: ")
  ```
  will be whatever string the user types as input.

Enter to exit

- Many programs end with:
  ```python
  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.
- The value produced by `input()` is not put into a variable; it is just thrown away.

Checking for no input

- Use the `if` statement:
  ```python
  if name == "":
      print ("No name received.")
  ```

  ```python
  if condition:
      do some statements
  ```

The empty string

- When the user just types the Enter key, `input()` produces 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

The `==` operator

- The condition (for example, `name == ''`) is testing for equal values.
- It is not an assignment statement!
- It uses the operator `==`, which is different from the `=` in an assignment statement.

Not equal

```python
if name != "":
    print ("You are the","name","meister!")
```

- The symbol `!=` means "is not equal"
- The symbol `==` means "is equal"
True and False

- These are called the Boolean data values.
- Boolean is one more data type.
- There is no “maybe” or “it depends”; True and False are the only possible Boolean values.
- This is an expression that produces a Boolean value.

Malformed if statements

- if name == “”;
  print (“Welcome!”)  Uses = instead of ==
- if name == “”
  print(“Welcome!”)  Missing :
- if name == “”;
  print (“Welcome!”)  No indentation

If ... else...

- if name == “”:
  print(“No name received.”)
else:
  print (“You are the”, name+”meister!”)

Either the block under if is done, or the block under else.
One or the other is executed, NEVER BOTH.

If ... else...

- if name == “”:
  print(“No name received.”)
else:
  print(“You are the”,name+”meister!”)

The command else is NOT indented.
In IDLE, use backspace key to get back to left side.

Block

- The indented statements under the if form a block.
  A block is always done together, or skipped together.

if name == “”:
  print(“No name received.”)
  print (“We will not proceed.”)
if...elif...else

- Only one of the blocks gets executed.
- Tries first condition; if that works, execute that block.
- If not, try second condition.
- If that fails too, the else gets executed.
- Can have as many elif's as you want.