Second program assigned, due Thursday.
Food Quiz!

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

\[ x = 5 \]

\[ y = x \]

\[ x = 2 \]

\[ y = 5 \]
Getting user input

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

- This is an assignment statement.
- `inStr` is a variable.
- `input` is a function
- The value of the expression:
  
  ```python
  input("Enter celsius temperature: ")
  ...
  ```

  will be whatever string the user types as input.

Checking for no input

- Use the `if` statement:

  ```python
  if name == "":
      print ("No name received.")
  ```

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

The empty string

- Characters are letters, numbers, punctuation...what you get when you hit a keyboard key.
- 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 (name+"-o-rino")
```

- The operator `!` means "is not equal"
- The operator `==` 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.

  ```python
  name == ""
  ```

- This is an expression that produces a Boolean value.
Block

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

```python
if name == "":
    print( "No name received.")
    print( "We will not proceed.")
```

Malformed if statements

- if name = "": Uses = instead of ==
  ```python
  print ("Welcome!")
  ```
- if name == "": Missing :
  ```python
  print("Welcome!")
  ```
- if name = "": No indentation

If ... else...

```python
if name == "":
    print ("No name received.")
else:
    print (name+"-o-rino")
```

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

If ... else...

```python
if name == "":
    print( "No name received.")
else:
    print (name+"-o-rino")
```

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

if ... elif...else

```python
if name == ‘Fido’:
    species = ‘dog’
elif name == ‘Fluffy’:
    species = ‘cat’
else:
    species = ‘human’
```
if...elif...else

- Only one of the blocks gets executed.
- Trys 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.