

# EECS 12: Lecture 1

## Variables, Expressions, Statements, Programs

Mark E. Phair

[mphair@gmail.com](mailto:mphair@gmail.com)

UC Irvine EECS

June 28th, 2006

# Agenda

- Types and values
- Variables
- Expressions
- Statements
- Programs
- Comments

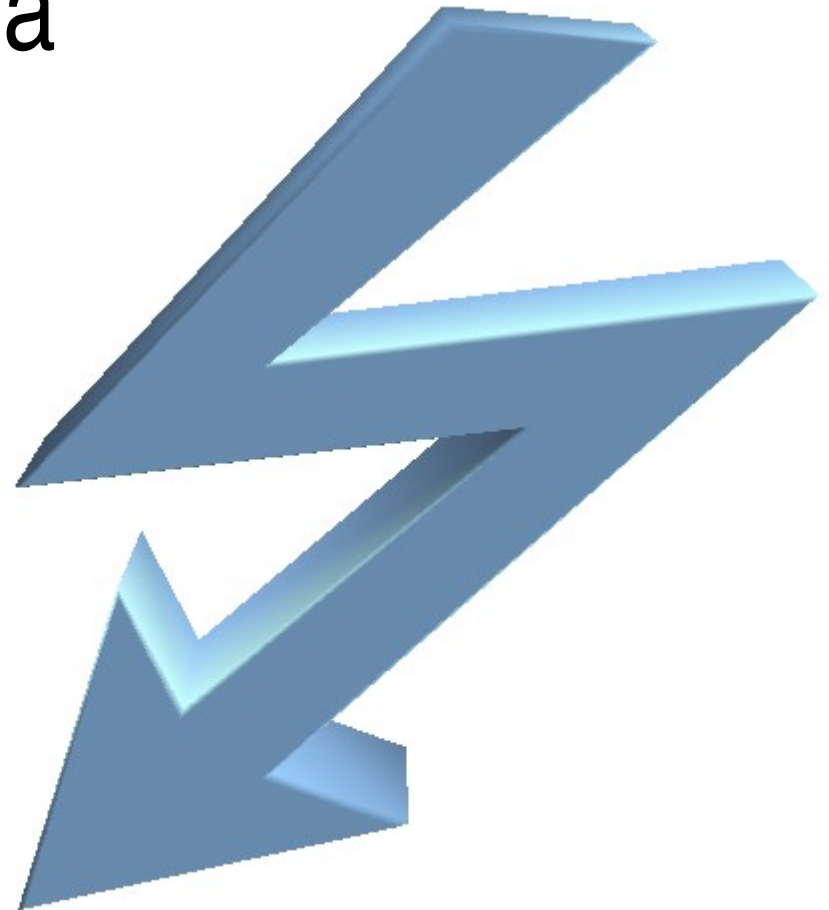


Figure 1: Very important and meaningful graphic

# Types

- Already familiar with types from math (and from last lecture!)
- python `type ( )`
- Let's investigate what types things are: find things that are all of the following types:
  - easy: int, str, float
  - more difficult: builtin\_function\_or\_method, list, type

# Converting between Types

- `int()`, `str()`, `float()`
- `ord()`, `chr()` (ASCII/Unicode)
- Let's try a few...

# Values

- Values are things that have types (this is subtly different from other languages)
- 234
- 11235
- 3.14159
- “hello, world”
- “yes, there are other strings besides 'hello, world'”

# Variables

- Variables have assigned values...
- Variables have names...
- Variables can be thought of as *named values*
- Naming **rules**: letter followed by digits, letters, and underscores (`_`)
- Naming **conventions**: typically, variables do not start with a capital letter

# Assignment (no, not as in homework)

- $x = 1$
- `aString = "hello, turtle"`
- `this_is_a_rather_long_variable_name = 0`
- **$x = x + 1$**
- Let's explore... assign some variables some values...  
perform some computations

# Expressions

- *syntactically correct* combinations of values, variables, and operators
- when an expression is evaluated, it has a value
- expressions by themselves do not have values ( $x+1$  cannot have a value until we know what the value of  $x$  is)



# print

- `>>> x = 5`
- `>>> print x`
- `5`
- `>>> y = 3`
- `>>> print x, "is awesome, but", y,`  
`"is cool"`
- `5 is awesome, but 3 is cool`

# Statements

- so far:
  - assignment
  - `print`
  - expressions, in the interpreter (not usually useful elsewhere)
- one per line of a program

# Programs

- Collections of statements
- Executed (in python) in sequential order
- Stored in a file (.py file extension)
- Let's explore: Write a program that
  - assigns some values to some variables
  - prints out the values of the variables

# Comments

- VERY IMPORTANT!!!!!!
- # comment, because of the #
- not a comment #this is comment
- Comments tell other people reading your code what it does, what you were thinking, unexpected side effects
- Sometimes the “other person” reading it is **YOU!**