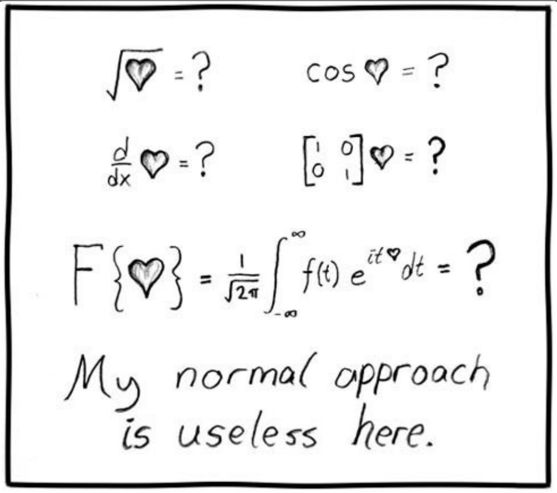

√♡ = ?
cos ♡ = ?
d/dx ♡ = ?
[1 0; 0 1] ♡ = ?
F{♡} = 1/√2π ∫ f(t) e^{it♡} dt = ?
My normal approach is useless here.

The Value of Everything & Procedure Practice

One-Slide Summary

- In Python, expressions **evaluate** to values. Five **evaluation rules** describe this process.
- **Lambda** means “make a function”. A lambda expression gives the formal parameter and function body. **Def** can also make functions.
- Evaluating a **function application** involves evaluating the function, finding its body, replacing the formal parameters with the evaluated actual arguments, and evaluating the result.

Lecture Outline

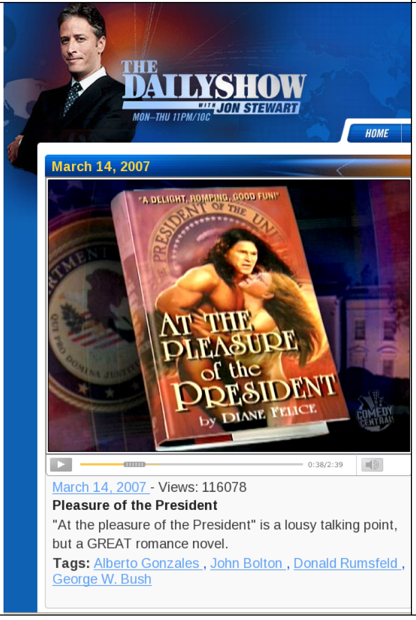
- Survey Responses
- Evaluation Rules
  - Lambda
- Problem Set 1
  - Decent Python

How To Use Lab Hours

- Read the problems on your own and try them out first.
  - You can **not** just go to a TA and say “I don't get it, what do I do.”
    - The TA is allowed to send you away.
  - You must demonstrate about five minutes worth of work: either on scratch paper, or with code you've tried and commented out.
  - For example: how would you do it in English?
- Talk to your friends.
- Do **not** expect to finish the Problem Sets in just the staffed lab time.
  - They take longer. You must do much work alone.

The Forum!

- Your questions for me are answered on the forum.
- Any questions right now?


THE DAILY SHOW WITH JON STEWART
MON–THU 11PM/10C
HOME
"A DELIGHT, ROMPING, GOOD FUN!"
AT THE PLEASURE of the PRESIDENT by DIANE FELICE
0:38/2:39

March 14, 2007

March 14, 2007 - Views: 116078

Pleasure of the President

"At the pleasure of the President" is a lousy talking point, but a GREAT romance novel.

Tags: Alberto Gonzales , John Bolton , Donald Rumsfeld , George W. Bush

Problem Set 1

- Python's **Evaluation Rules** tell you how to find the value of any expression.
- Questions 1 and 2 ask you to evaluate Scheme expressions in your mind
  - This is a popular exam question.
- Without Evaluation Rules: guesswork
- Once you know the Evaluation Rules, you can answer without any guessing!

Evaluation Rules

- **Primitives** (55 * 66)
  - Evaluate to their pre-defined values
- **Names** (x + 2)
  - Evaluate to the value associated with that name
- **Application** square-root(144)
  - Eval all sub-expressions. Apply the value of the first (a function) to the values of the others.
- **Lambda** (lambda (x) : (x * x * x))
  - Evaluates to a function with parameters and body
- **If** if (3 < 5): 99
  - Eval predicate. If False/0/[]/””, eval second option. Otherwise, eval the first option.

Primitive Examples

What do these evaluate to?

5

-88

True

False

+

Primitive Examples

```
5       -->   5
-88     -->   -88
True    -->   True
False   -->   False
  +       -->   Syntax Error
```

Name Examples

x = 55

y = 66

What do these evaluate to?

x

y

z

Name Examples

x = 55

y = 66

```
x   --> 55
y   --> 66
 z    --> NameError: name 'z' is not defined
```

Application E

What do these evaluate to?

sqrt(16)

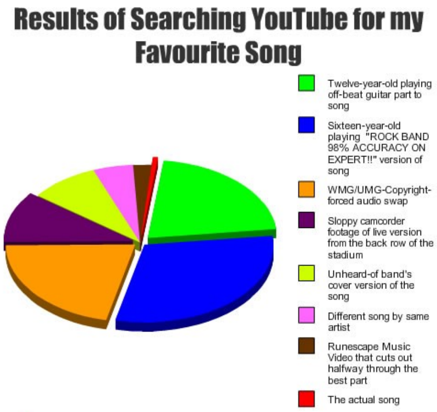
abs(-5)

len(“Hi”)

(1 + 2)

(1 + 2 + 3)

(+ 1)



Application Examples

sqrt(16)	--> 4
abs(-5)	--> 5
len(“Hi”)	--> 2
(1 + 2)	--> 3
(1 + 2 + 3)	--> 6
(+ 1)	--> 1

Liberal Arts Trivia: Anthropology

- This American cultural anthropologist is famous for her studies of Samoa and her reports about the purportedly healthy attitude towards sex in South Pacific and Southeast Asian traditional cultures, which influenced the women's liberation movement (e.g., by claiming that females dominated in Chambri and Papau New Guinea without problems). Five years after she died, her work was challenged by Derek Freeman.

Liberal Arts: Slavic Folklore

- This witch-like character in Slavic folklore lives in a walking house with chicken feet (but no windows and no doors), flies around on a giant mortar, and kidnaps (presumably to eat) small children. Modest Mussorgsky's *Pictures at an Exhibition*, a piano suite composed in 1874, features "The Hut on Bird's Legs" as its penultimate movement.

Lambda

- **Lambda** means “make a function”.
- Consider: cube(x) = x * x * x
- Python: def cube(x): return (x * x * x)
- Lambda: cube = (lambda (x) : (x * x * x))
- Notes:
  - With lambda, the word “return” is implicit!

Anatomy Of A Function

- **cube = (lambda $\underbrace{\textbf{(x)}}_{\text{formal parameters}}$: ($\underbrace{\textbf{x * x * x}}_{\text{function body}}$))**
- $\underbrace{\textbf{cube(}\underbrace{\textbf{5}}_{\text{actual arguments}}\textbf{)}}_{\text{function application}}$
- To **evaluate a function application**, replace it with the function body, and then replace every formal parameter with its corresponding actual argument.
- cube(5) -> (x * x * x) -> (5 * 5 * 5) -> 125

Lambda Examples

cube = (lambda (x) : (x * x * x))
foo = (lambda (p, q) : (p + q))
bar = (lambda (a, b, c) : (a * c))

cube(3)

foo(5, 6)

bar(4, 5, 6)

foo(cube(3), 1)

What do these evaluate to?

Lambda Examples

- cube = (lambda (x) : (x * x * x))
- foo = (lambda (p, q) : (p + q))
- bar = (lambda (a, b, c) : (a * c))
- cube(3)   -> 3*3*3 -> 27
- foo(5, 6) -> 5+6 -> 11
- Bar(4, 5, 6)   -> 4*6 -> 24
- foo(cube(3), 1) ->...-> 27+1 -> 28

Lambda Lambda Lambda

- Consider these two functions:
  - cube = (lambda (x) : (x * x * x))
  - cube = (lambda (y) : (y * y * y))
- Are they different?
- Consider:
  - nail = (lambda (x, y) : (x + y))
  - polish = (lambda (y, x) : (y / x))
- What is:
  - polish((nail(6, 4), 2)

Do this now on paper!

Sally Hansen does Lambda

nail = (lambda (x y) (x + y))

polish = (lambda (y x) (y / x))

polish(nail(6, 4), 2)

- This is a call to **polish** with tricky arguments.
- Recall the rule: evaluate the arguments first.
  - Argument 1:   nail(6, 4) -> (x + y) -> (6 + 4) -> 10
  - Argument 2:   2 -> 2
- Now take **polish**'s body, and replace the formal parameters with the actual arguments:
  - (y / x) -> (10 / 2) -> 5

Why not (2 / 10) ?

If Examples 1

```
if True:
  Return “yes”
else:
  Return “No”


if False:
  Return “yes”
else:
  Return “no”
```

What do these evaluate to?

If Examples 1

```
if True:
  Return “yes”
else:
  Return “No”


if False:
  Return “yes”
else:
  Return “no”
```

“yes”

'no'

If Examples 2

```
If (3 < 5):
  Return “ant”
else:
  Return “bat”


if “x”:
  Return “y”
else:
  Return “z”
```

What do these evaluate to?

If Examples 2

```
If (3 < 5):
  Return “ant”
else:
  Return “bat”
```

'ant'

```
if “x”:
  Return “y”
else:
  Return “z”
```

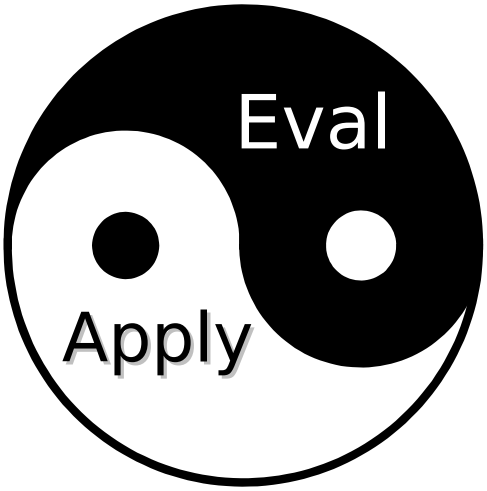
“y”

Python Trickery

- +(100, 100)
  - Error: You can't use primitive + like a function name. Write (100 + 100) instead.
- if (not “batterie”): “fouetté” else: “plié”
  - “plié”. **(not “batterie”)** returns **False**, because **“batterie”** is not **False/[]/””/0**.
    - def not(v): if v: return False else: return True
- Does (if *X*: True else: False) always equal *X* ?
  - Yes for True, False, (3 < 5), (5 > 6).
  - No for 3, 17, “hello”.

Evaluating expressions and **Applying functions** are defined in terms of each other.

Without Eval, there would be no Apply, Without Apply there would be no Eval!



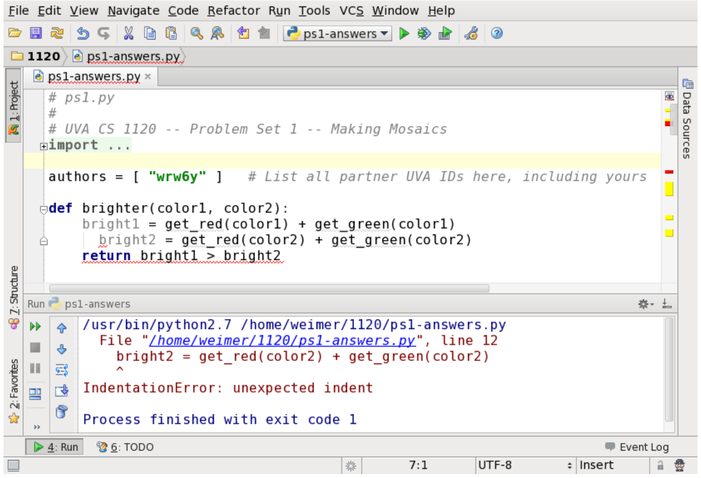
Now You Know All* of Python!

- Once you understand **Eval** and **Apply**, you can understand all Python programs!
- *Except:
  - There are many primitives, and you need to know their predefined meaning.
  - There are a few more special forms (like **if**).
  - We have not define the evaluation rules precisely enough to unambiguously understand all programs (e.g., what does “value associated with a name” mean?).

Now On To Problem Set 1

- Smooth transition ...

Python Enforces Tabbing



```
# ps1.py
#
# UVA CS 1120 -- Problem Set 1 -- Making Mosaics
import ...

authors = [ "wrw6y" ]   # List all partner UVA IDs here, including yours

def brighter(color1, color2):
    bright1 = get_red(color1) + get_green(color1)
      bright2 = get_red(color2) + get_green(color2)
    return bright1 > bright2
```

```
/usr/bin/python2.7 /home/weimer/1120/ps1-answers.py
  File "/home/weimer/1120/ps1-answers.py", line 12
    bright2 = get_red(color2) + get_green(color2)
    ^
IndentationError: unexpected indent

Process finished with exit code 1
```

Brighter **brighter**?

```
def brighter(c1, c2):
     X1 = get_red(c1) + get_green(c1)
     X2 = get_red(c2) + get_green(c2)
     return X1 > X2
```

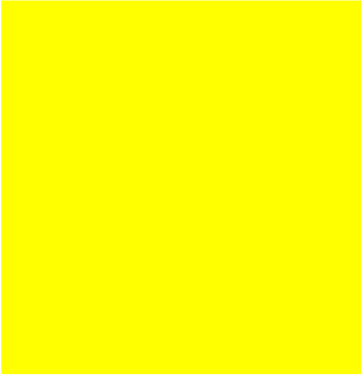
What can we do about this duplicated code?

Brighter **brighter**?

```
def brightness(color):
  return get_red(color)+get_green(color)

def brighter(c1, c2):
  return brightness(c1) > brightness(c2)
```

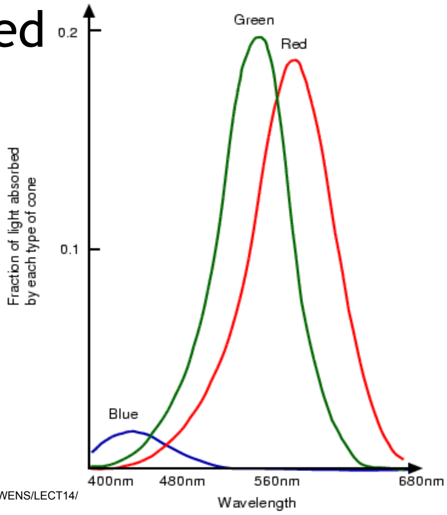
Believable **brighter**?



make-color(255, 255, 0)

make-color(255, 1, 255)

Color Absorbed



http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT14/

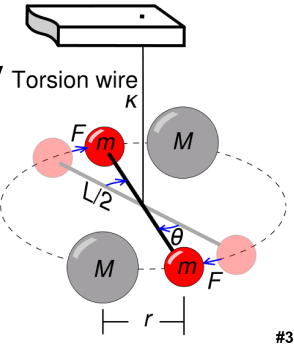
Cognitive Scientist's Answer

```
def brightness(color):
     return (0.299 * get_red(color)) +  \
            (0.587 * get_green(color)) +\
            (0.114 * get_blue(color))

def brighter(color1, color2):
     return brightness(color1) > \
            brightness(color2)
```

Liberal Arts Trivia: Physics

- This 1797 torsion balance experiment, sometimes called "weighing the earth", was the first to measure the force of gravity between masses in the laboratory, and the first to yield accurate values of the gravitational constant and thus the mass of the Earth.



Liberal Arts Trivia: Grab Bag

- Q. This series of music video games was produced by Konami in 1998. The series pioneered the rhythm and dance genre in video games. Players stand on a “dance platform” or stage and hit colored arrows laid out in a cross with their feet in time with musical and visual cues.

Liberal Arts Trivia: Drama

- This classical Athenian tragedy by Sophocles, first performed in BC 429, is widely considered a supreme masterpiece of the art of Drama. The Oracle at Delphi tells the protagonist that he is doomed to marry his mother and kill his father. He goes on to do so, but not before solving the riddle of the sphinx: What is the creature that walks on four legs in the morning, two legs at noon, and three in the evening? Name the play *and* answer the riddle.

What should you do if you can’t get your code to work?

- Keep trying: think of alternate approaches
- Get help from the TAs and your classmates
- But, if it's too late for that ...
  - **In your submission, explain what doesn’t work and as much as you can what you think is right and wrong**
- If you get less than 50% on the automatic adjudication part, the TAs will look over your source and give partial credit.

Homework

- (In theory) You now know everything you need for PS1, PS2, PS3 and PS4 ...
- Honor Pledge due today (now!)
- Problem Set 1 due September 6
  - ... at 3:30pm