Smith College
Computer Science

CSC 111
Introduction to Computer Science

Spring 2018 — Week 1

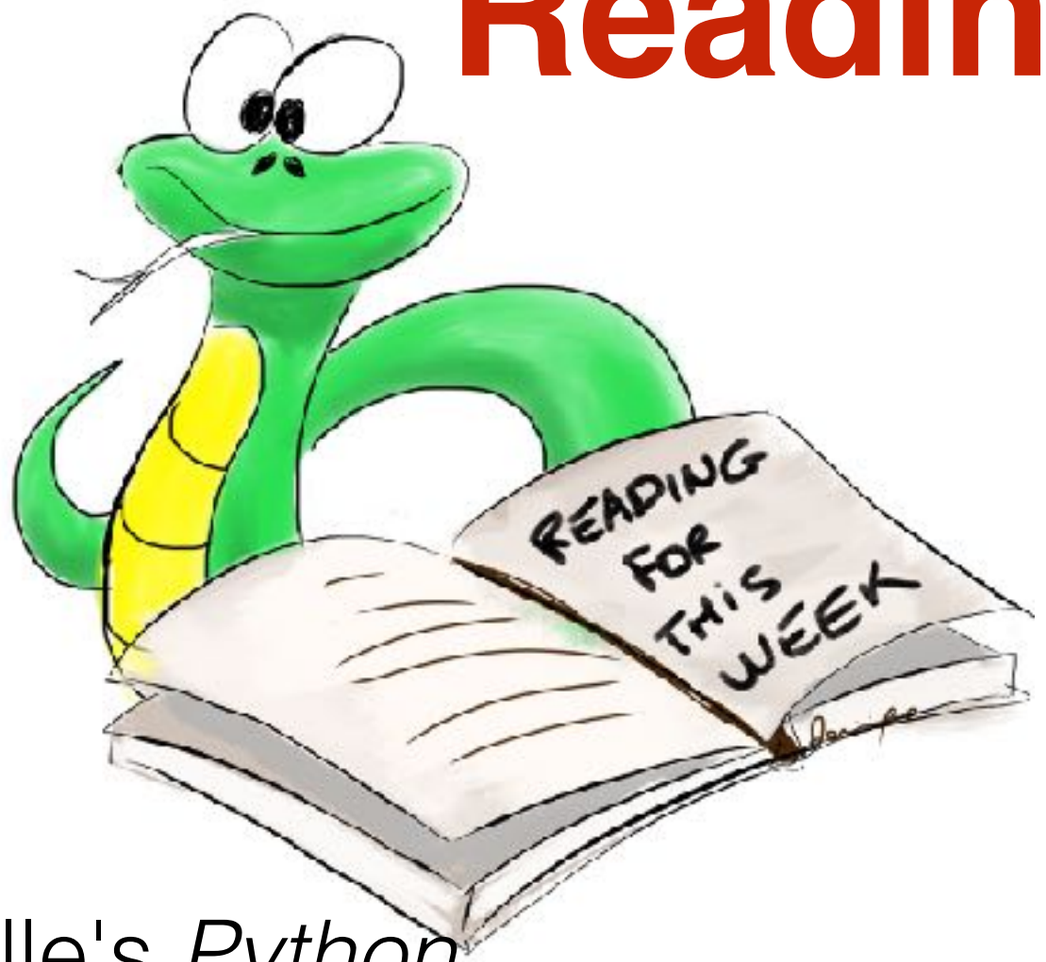
Dominique Thiébaut
dthiebaut@smith.edu

Quick Review

Goals for This Week

- Learn the Rules for **Pair Programming**
- Learn how to use **Idle**
- Write simple programs that use **variables, for loops,** and **output** informationf
- **Install** Python and Idle on laptop (optional)
- Learn how to **submit** Python programs to **Moodle** (lab+homework)

Reading



- Read **Chapter 1** in John Zelle's *Python Programming*

What is a Programming language?

Important Concepts…

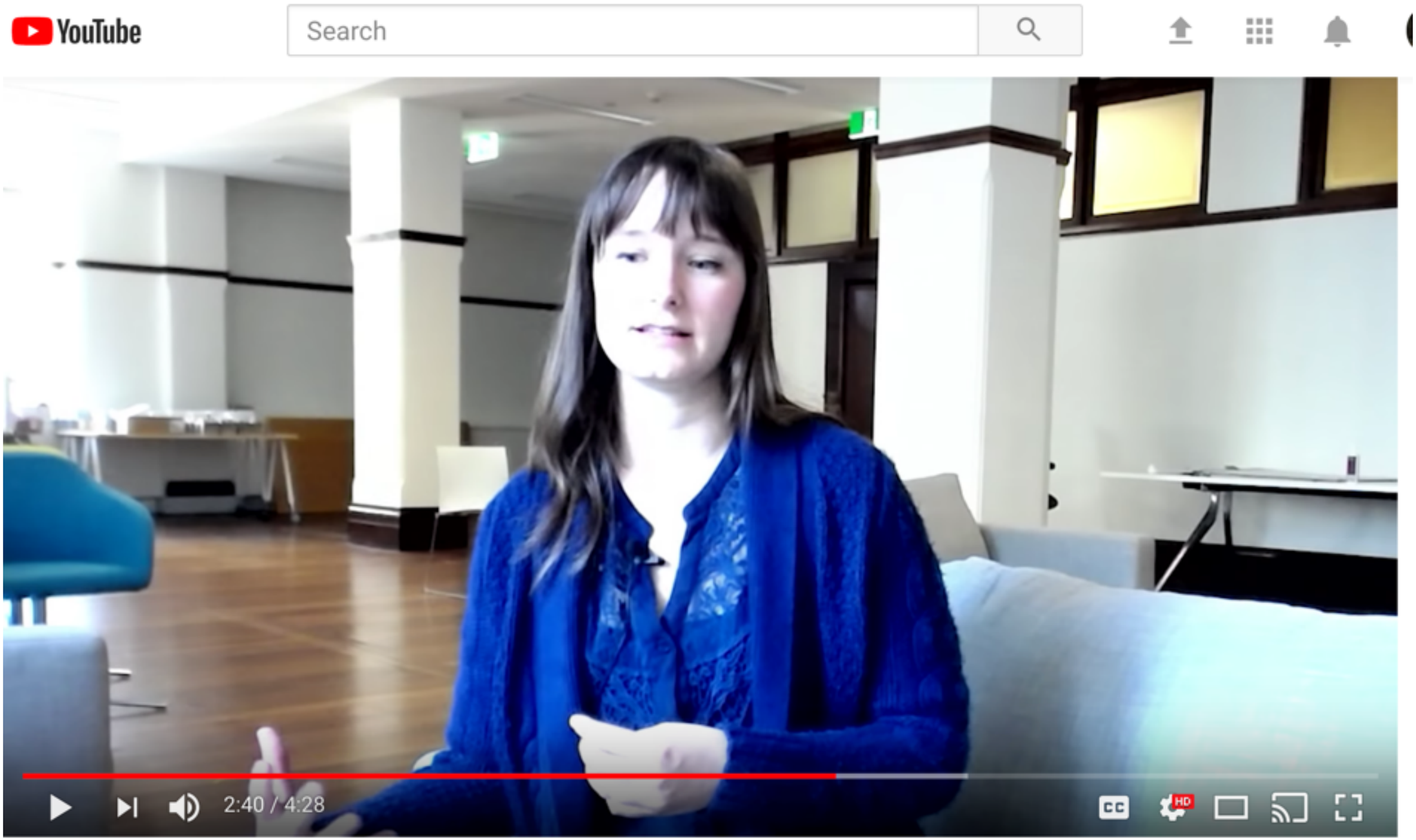
- **Syntax and keywords**

  and del from not while as elif global or with assert else if pass yield break except import **print** class exec in raise continue finally is return **def for** lambda try

- **Algorithm**

Rules for Pair Programming

https://youtu.be/fQ-x-T34z9w



An Example Program

example1.py - /Users/thiebaut/Desktop/Dropbox/111/example1.py

```
# A simple program taken from Zelle, Chapter 1
# D. Thiebaut

def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
        print( x )

main()
```

Ln: 12 Col: 0

INDENTATION IS IMPORTANT

COMMENT

DIFFERENT COLORS: SYNTAX HIGHLIGHTING

example1.py - /Users/thiebaut/Desktop/Dro...py

```
# A simple program taken from Zelle, Chapter 1
# D. Thiebaut

def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
        print( x )

main()
```

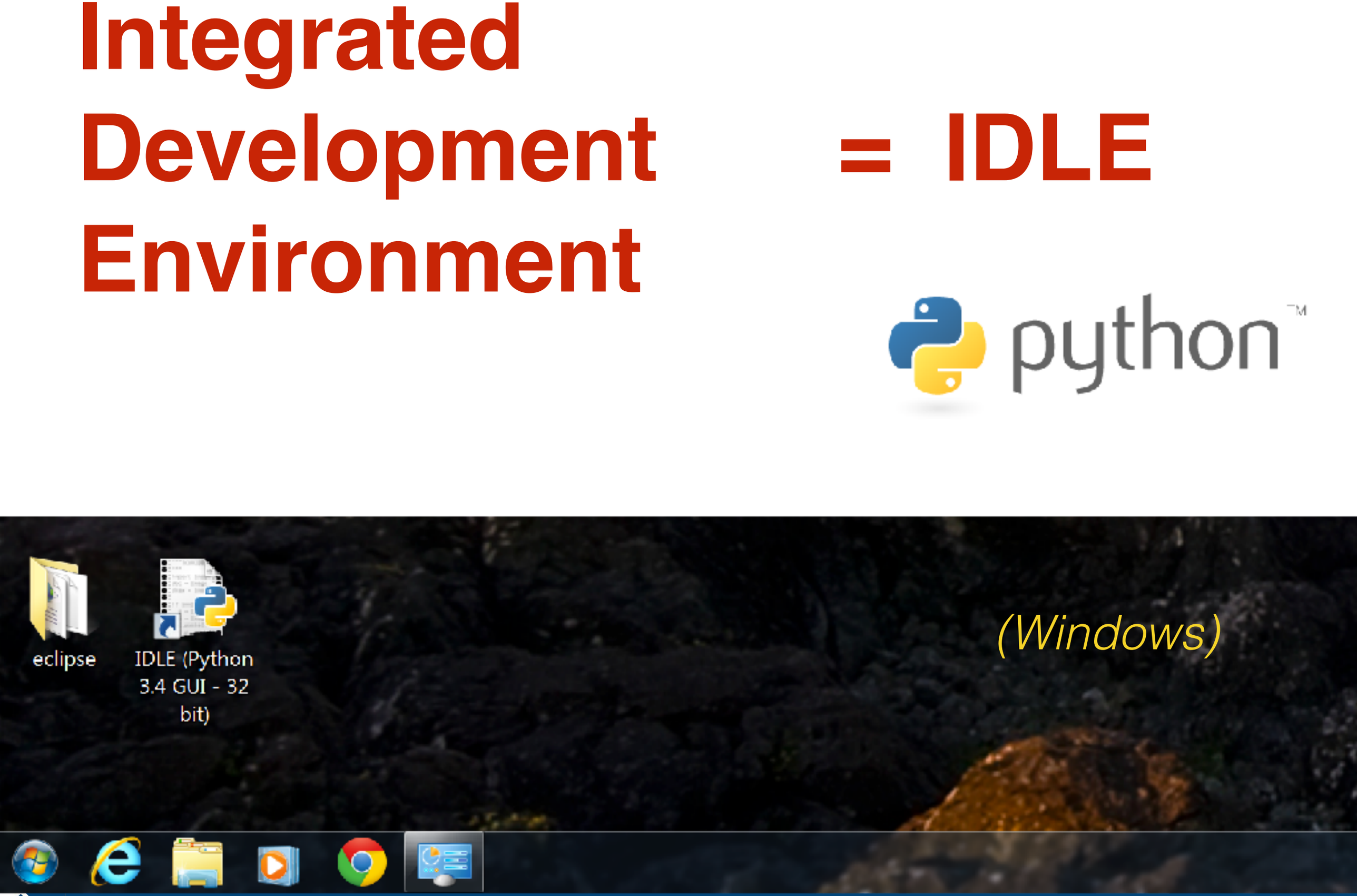
SPECIAL TOOL: EDITOR I D E

Integrated Development Environment = IDLE

Integrated Development Environment = IDLE

Integrated Development Environment = IDLE

Integrated Development Environment = IDLE



DEMO TIME!

example1.py - /Users/thiebaut/Desktop/Dropbox/111/example1.py

```
# A simple program taken from Zelle, Chapter 1
# D. Thiebaut

def main():
    print( "This program illustrates a chaotic function" )
    x = eval( input( "Enter a number between 0 and 1: " ) )
    for i in [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]:
        x = 3.9 * x * ( 1 - x )
        print( x )

main()
```

Ln: 6 Col: 28


Beginning
of the
Semester…

Concepts to Cover in Demo

- **Console** vs. **Edit window**
- ***Variables***
  - numbers: **integers** and **floats**
  - text: **strings** of characters
- **print** function

Demo Programs To Play With…

```
age = 20
year = 2015
yearBorn = year - age

print( "you are", age )
print( "you were born in", yearBorn )
```

```
name = "Alex"
college = "Smith College"
print( name, "goes to", college )
```

```
for name in  [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
     print( name )
     print( "————————" )
```

Demo Programs To Play With… (cont'd)

```
for name in  [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
     print( name, len( name ) )
```

```
print( "hello" * 4 )
print( "-" * 10 )
greetings = “hello”
dash = "-"
print( greetings * 4 )
print( dash * 10 )
```

```
greetings = "hello"
longGreetings = greeting * 4
print( greetings )
print( longGreetings )
```

Demo Programs To Play With… (cont'd)

```
for name in  [ "Lea Jones", "Julie Fleur", "Anu Vias" ]:
     bar = len( name ) * "-"
     print( name )
     print( bar )
```

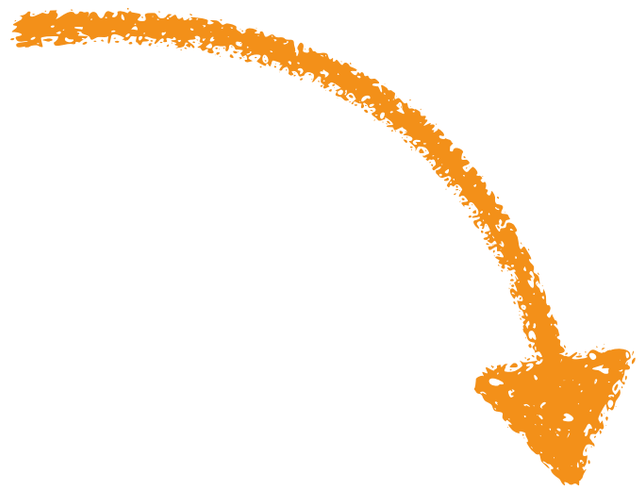
```
print( "hello" * 4 )
print( "-" * 10 )

greetings = “hello”
dash = "-"
print( greetings * 4 )
print( dash * 10 )
```

```
greetings = "hello"
longGreetings = greeting * 4
print( greetings )
print( longGreetings )
```

Exercise 1

Lea
Mary
Alice
Lujun
Anu
Shweta



```
======= RESTART: /Users/thiebaut/Desktop/Drop
Lea
Mary
Alice
Lujun
Anu
Shweta
>>>
```

Exercise 2

Lea
Mary
Alice
Lujun
Anu
Shweta

```
====== RESTART: /Users/thiebaut/Desktop/Dropbox
Lea
Box:                Id:

Mary
Box:                Id:

Alice
Box:                Id:

Lujun
Box:                Id:

Anu
```

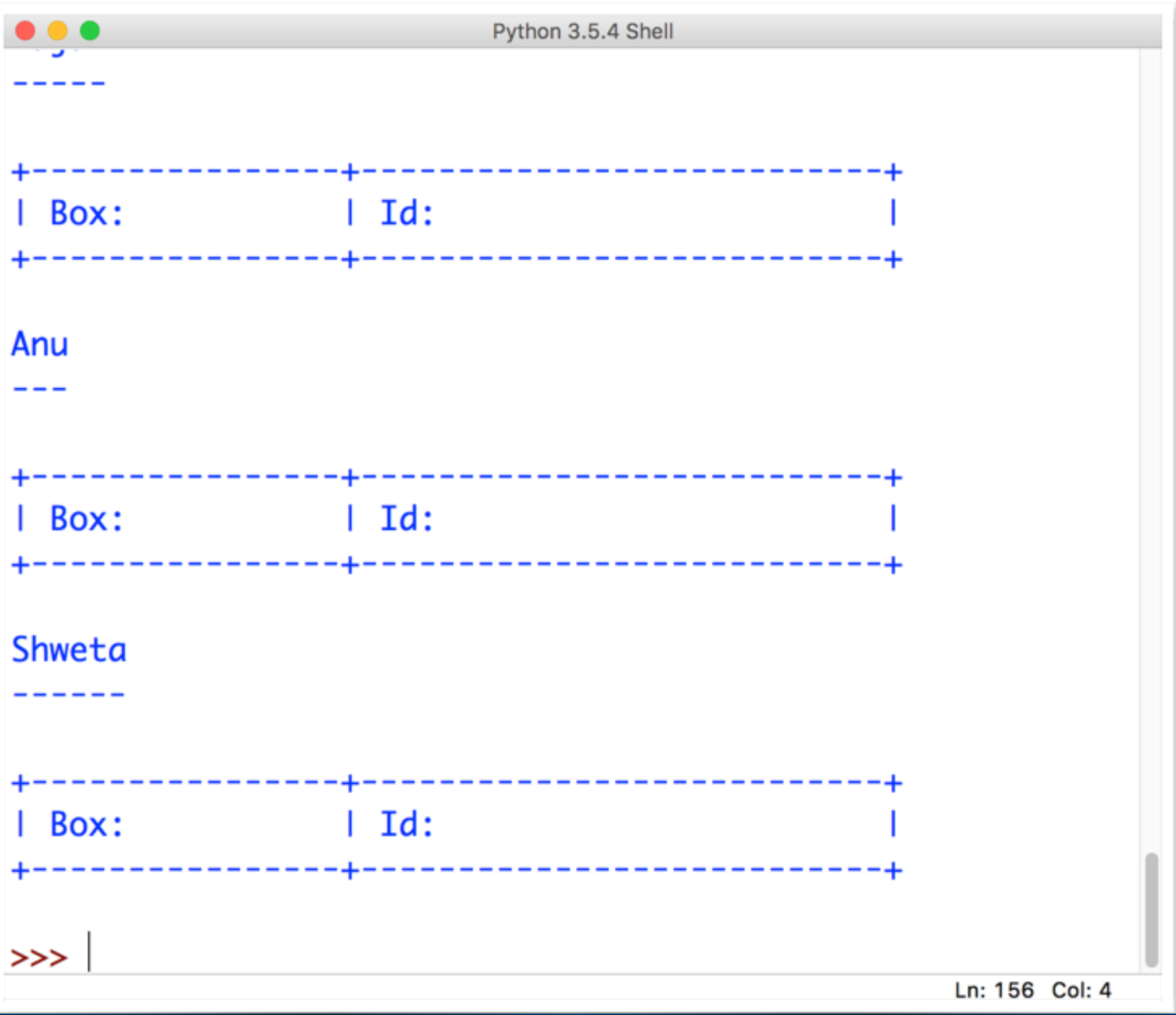
Exercise 3

Lea
Mary
Alice
Lujun
Anu
Shweta

```
====== RESTART: /Users/thiebaut/Desktop/Dropbox/1
Lea
+----------------+-----------------------------+
| Box:           | Id:                         |
+----------------+-----------------------------+
Mary
+----------------+-----------------------------+
| Box:           | Id:                         |
+----------------+-----------------------------+
Alice
+----------------+-----------------------------+
| Box:           | Id:                         |
+----------------+-----------------------------+
```

Exercise 4

Lea
Mary
Alice
Lujun
Anu
Shweta

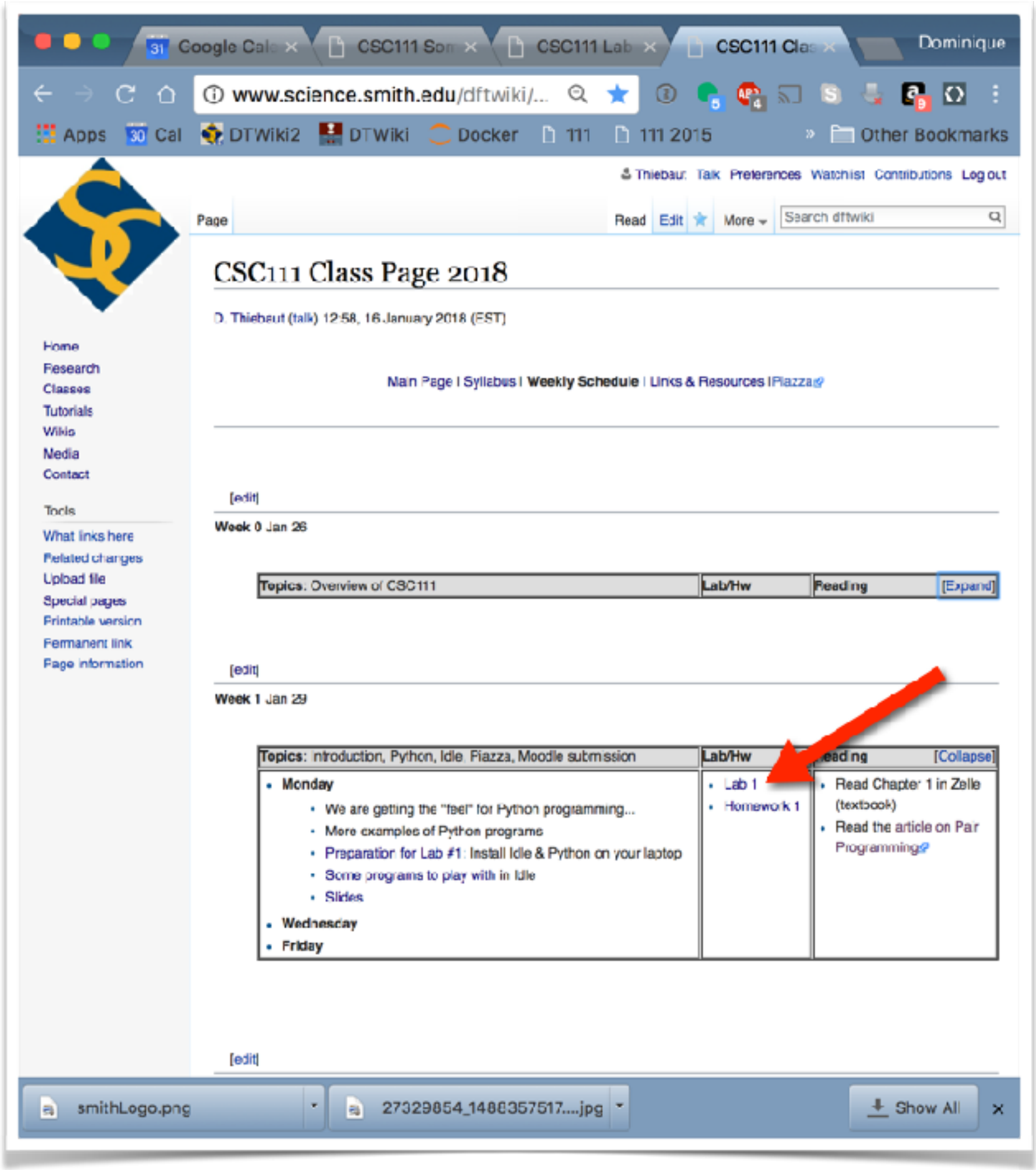


We stopped here last time…

Outline

- **Introduction to Lab 1**
- **Assignment**
- **Introduction to Variables**
- **Exercise**

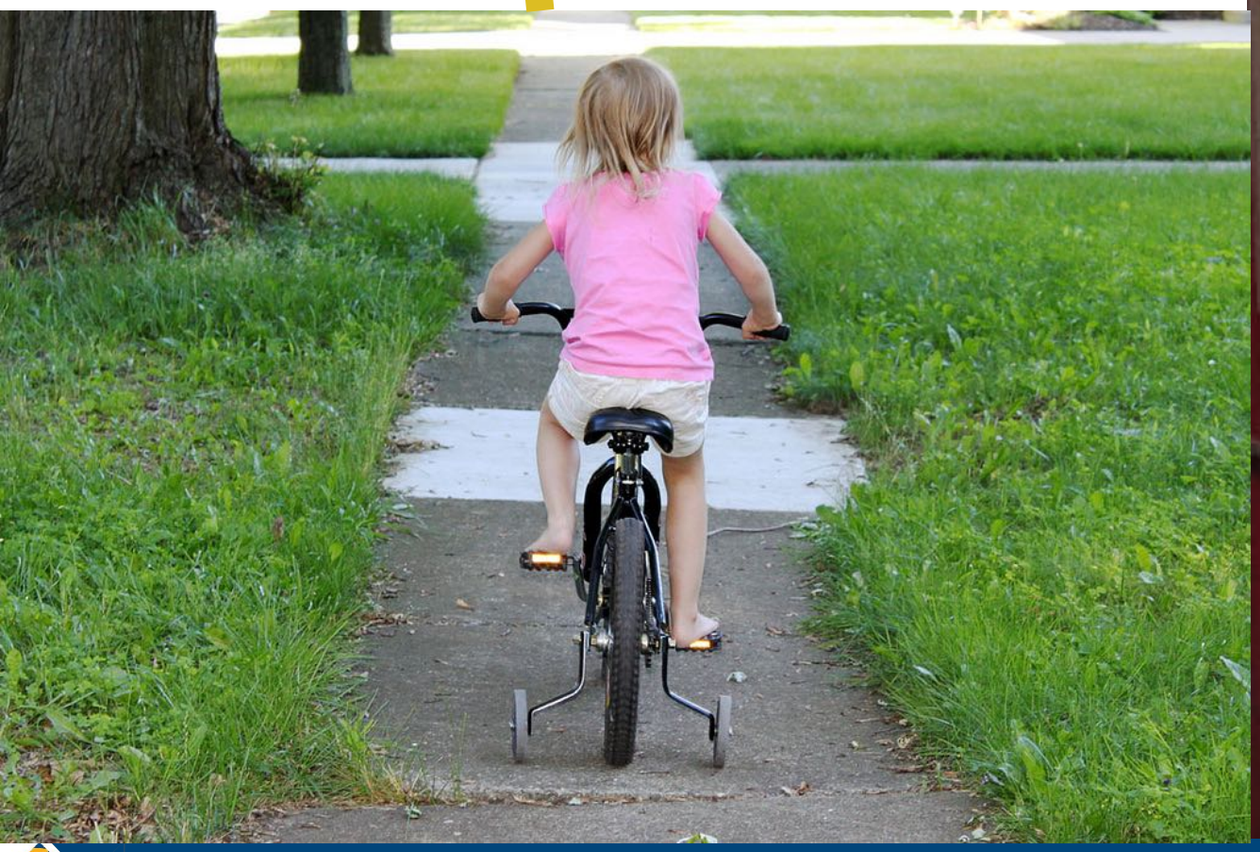
Lab 1


Google Cale
CSC111 Son
CSC111 Lab
CSC111 Clas
Dominique
www.science.smith.edu/dftwiki/...
Apps
Cal
DTWiki2
DTWiki
Docker
111
111 2015
Other Bookmarks
Thiebaut Talk Preferences Watchlist Contributions Log out
Page
Read
Edit
More
Search dftwiki
CSC111 Class Page 2018
D. Thiebaut (talk) 12:58, 16 January 2018 (EST)
Main Page | Syllabus | Weekly Schedule | Links & Resources | Piazza
Home
Research
Classes
Tutorials
Wikis
Media
Contact
Tools
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
Page information
[edit]
Week 0 Jan 26
Topics: Overview of CSC111
Lab/Hw
Reading
[Expand]
[edit]
Week 1 Jan 29
Topics: ntroduction, Python, Idle Piazza, Moodle submission
Lab/Hw
ead ng
[Collapse]
Monday
We are getting the "feel" for Python programming...
More examples of Python programs
Preparation for Lab #1: Install Idle & Python on your laptop
Some programs to play with in Idle
Slides
Wednesday
Friday
Lab 1
Homework 1
Read Chapter 1 in Zelle (textbook)
Read the article on Par Programming
[edit]
smithLogo.png
27329054_1486357517....jpg
Show All

Practice Python!

Beginning of the Semester

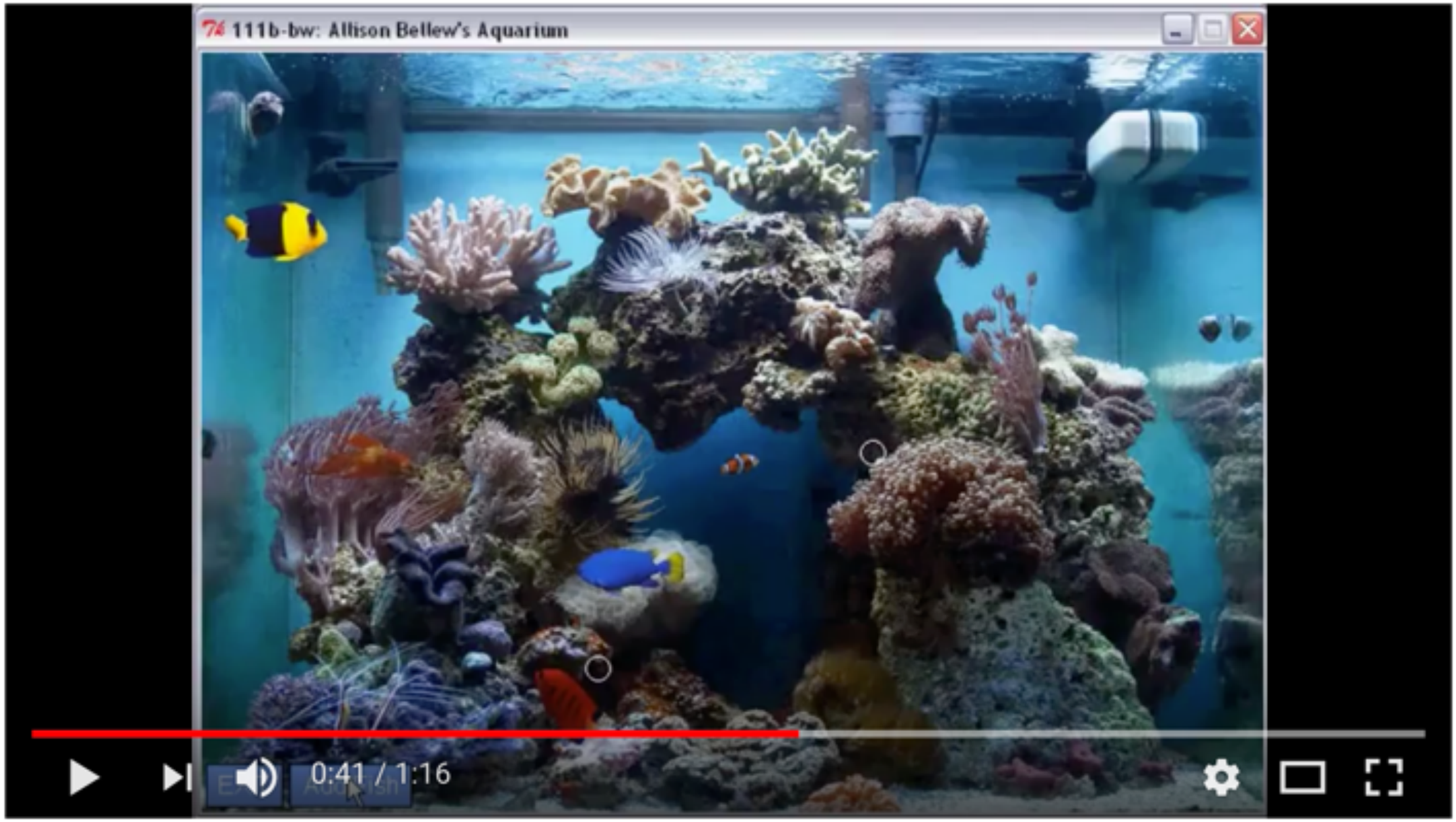
AFTER
ONE SEMESTER



Computer Science Major

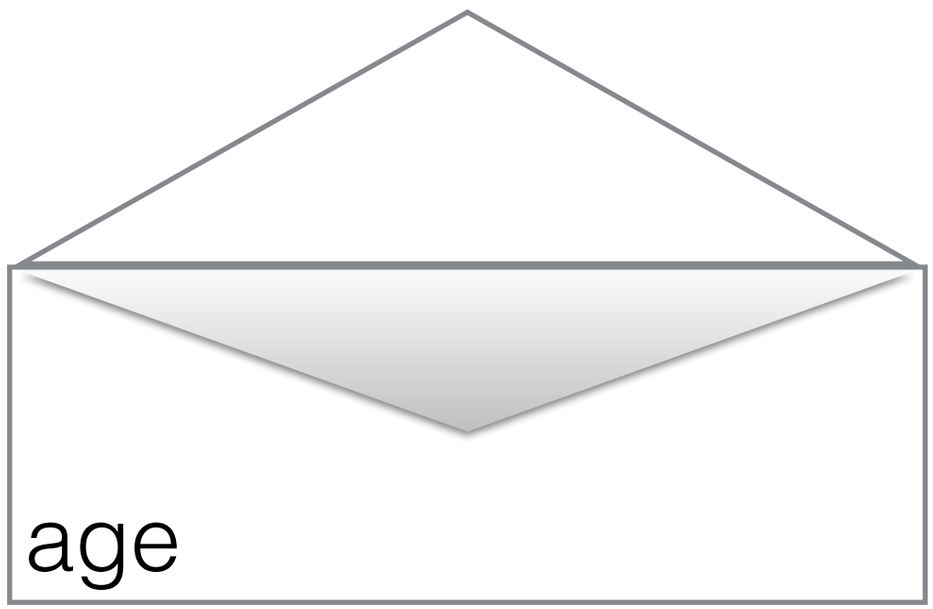
Final Project From the Past

https://www.youtube.com/watch?v=g_82xHimSNE

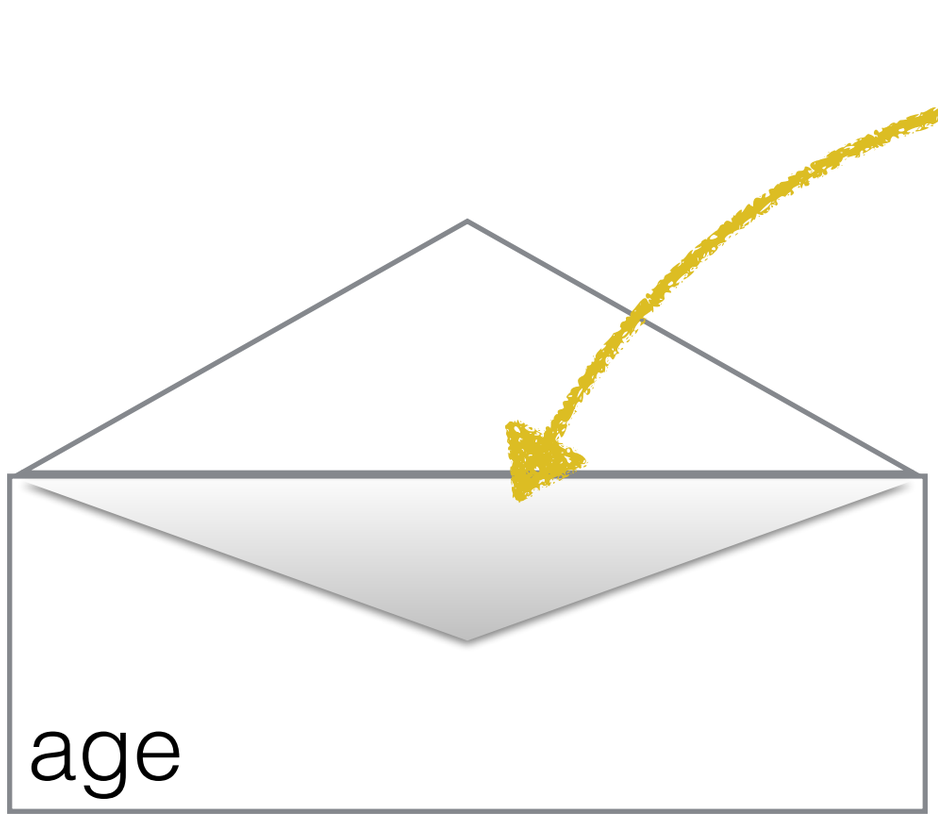


Memory

Variables

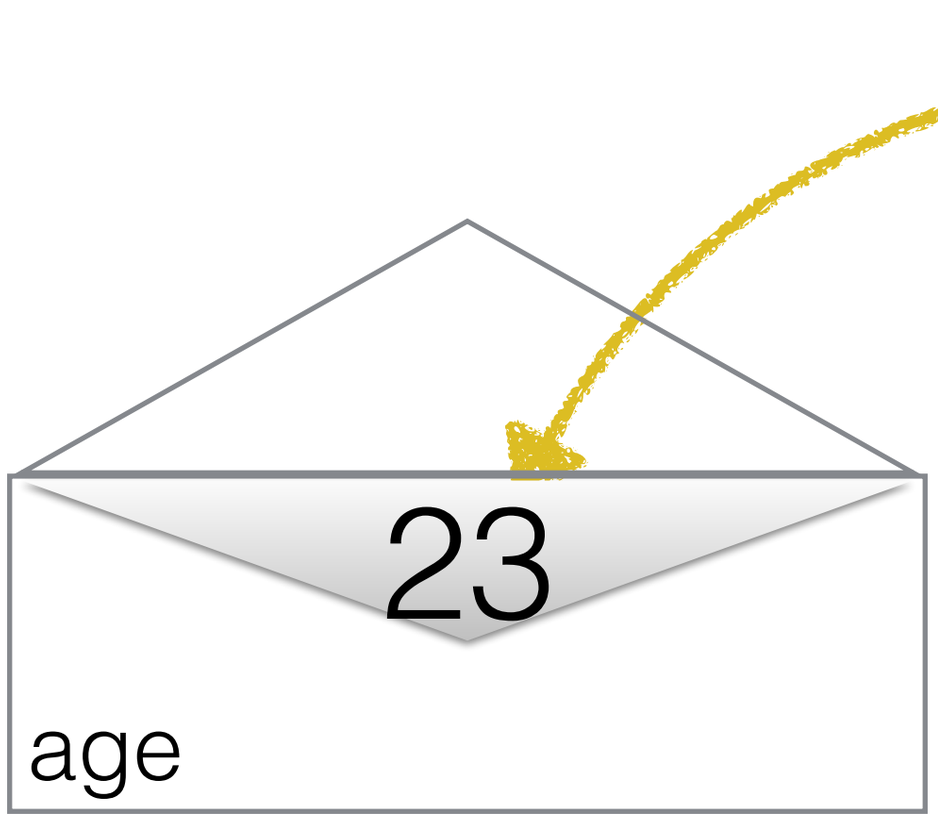


Variables



age = 23

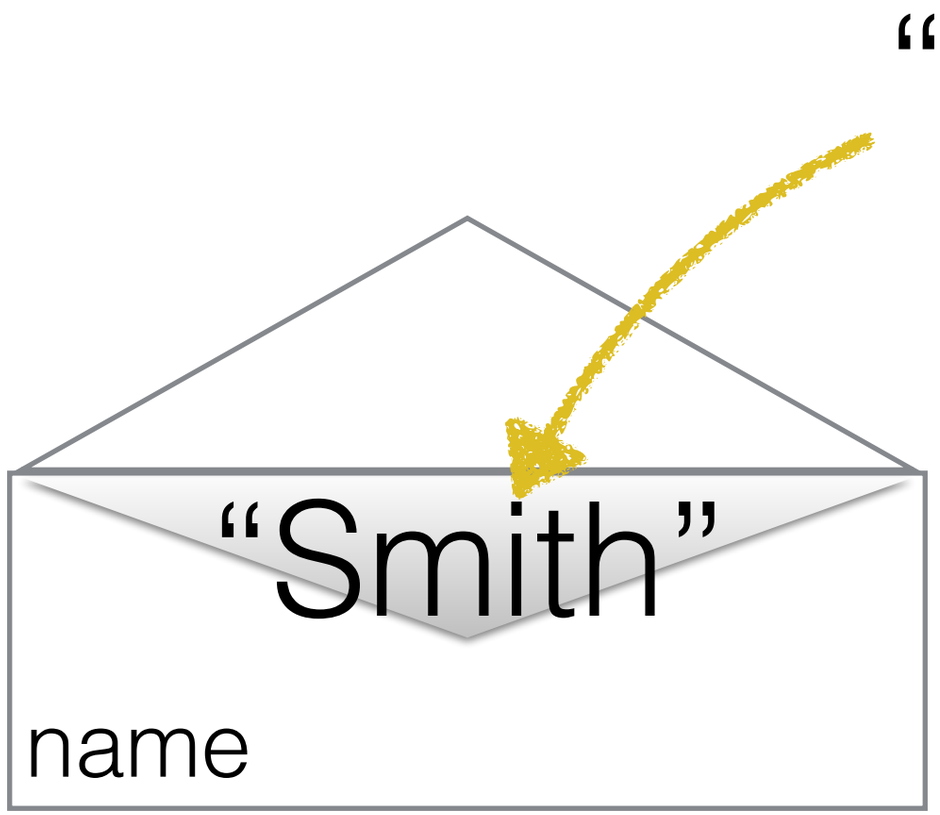
Variables



age = 23

assignment

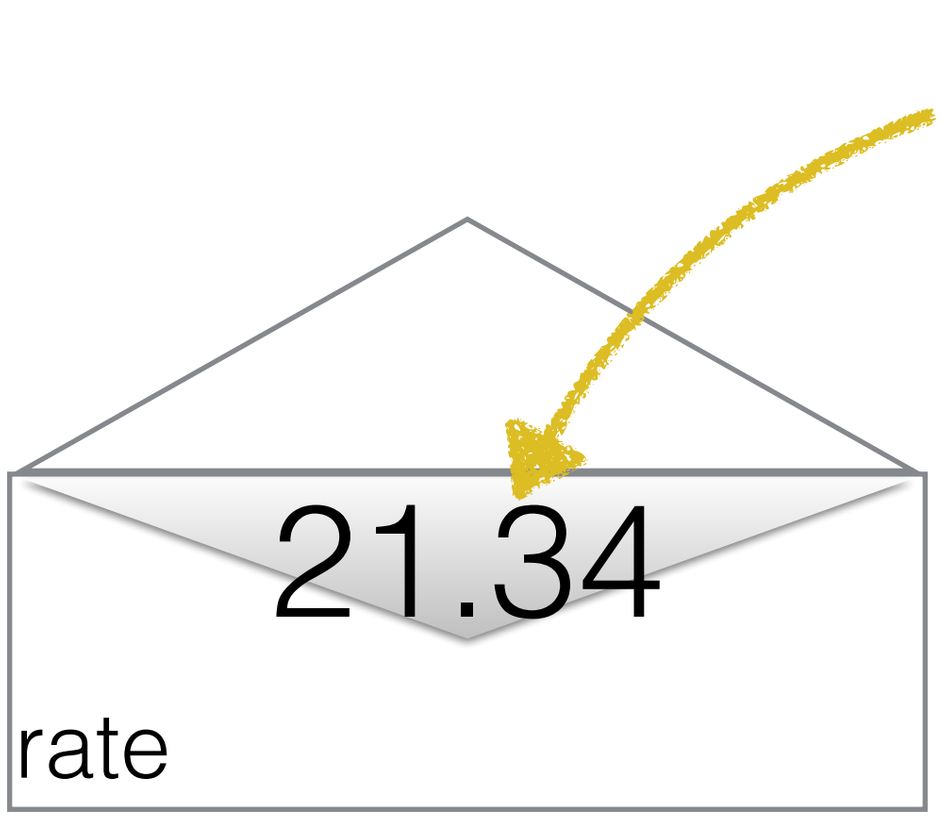
Variables



name = "Smith"

assignment

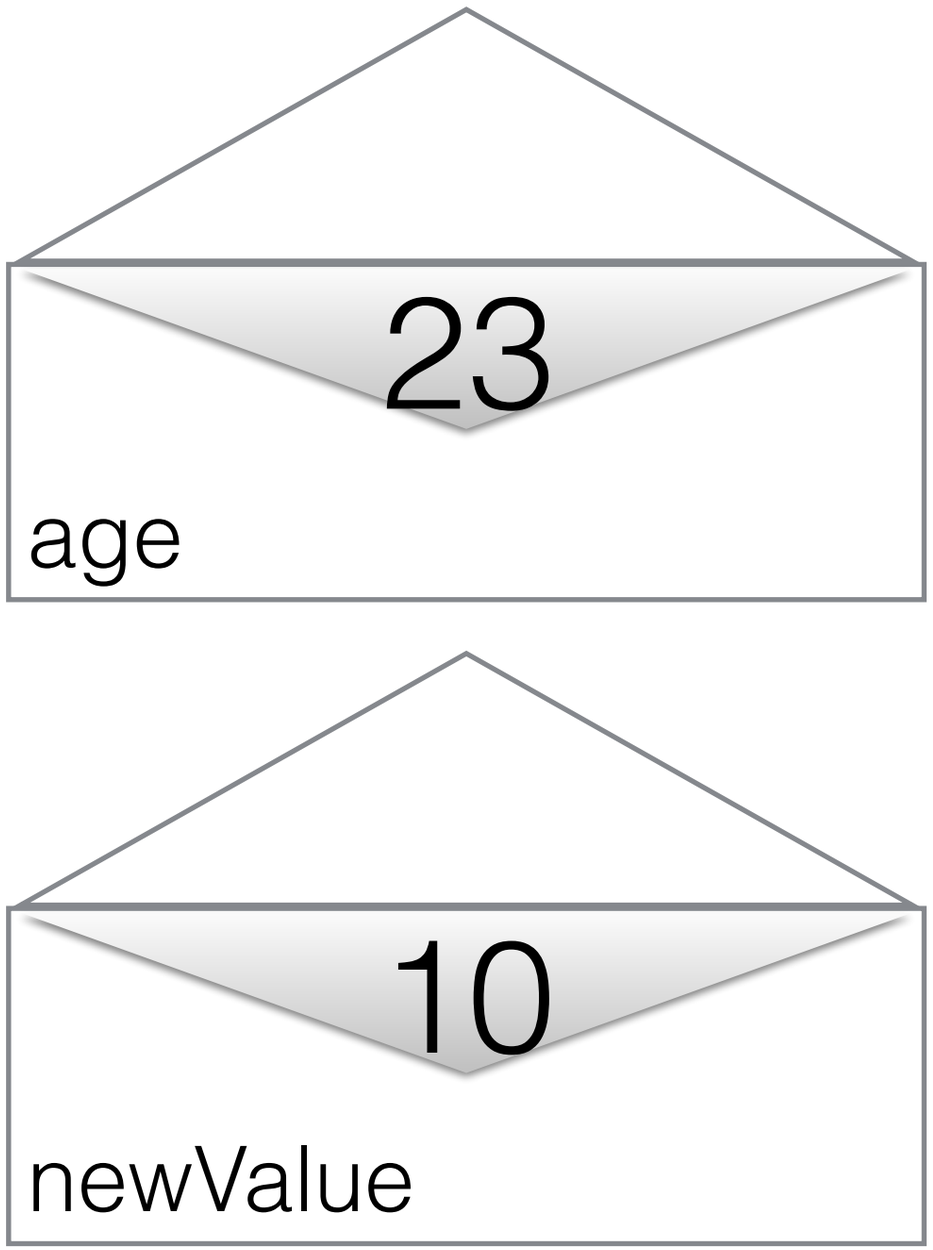
Variables



rate = 21.34

assignment

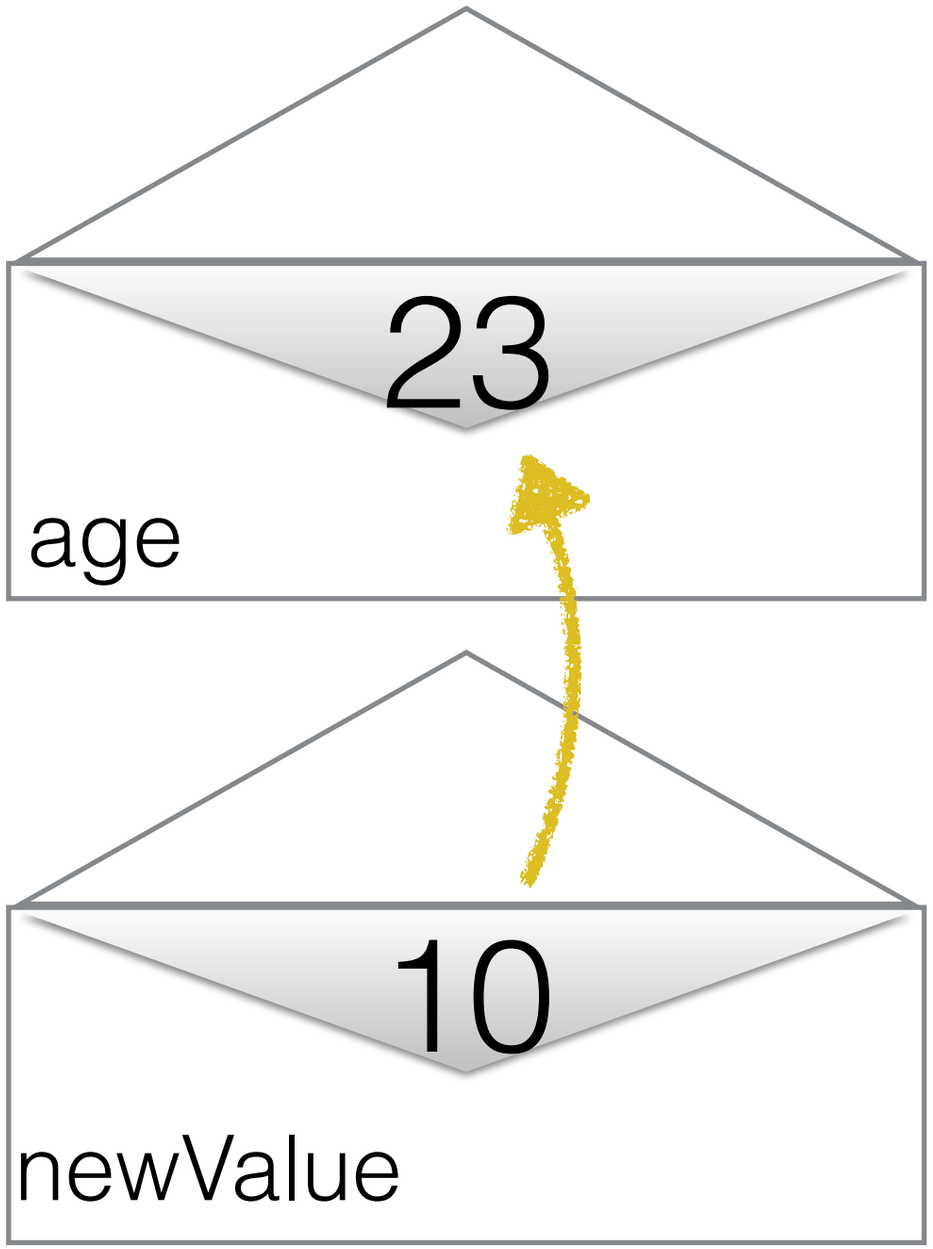
Variables & Expressions



age = 23

newValue = 10

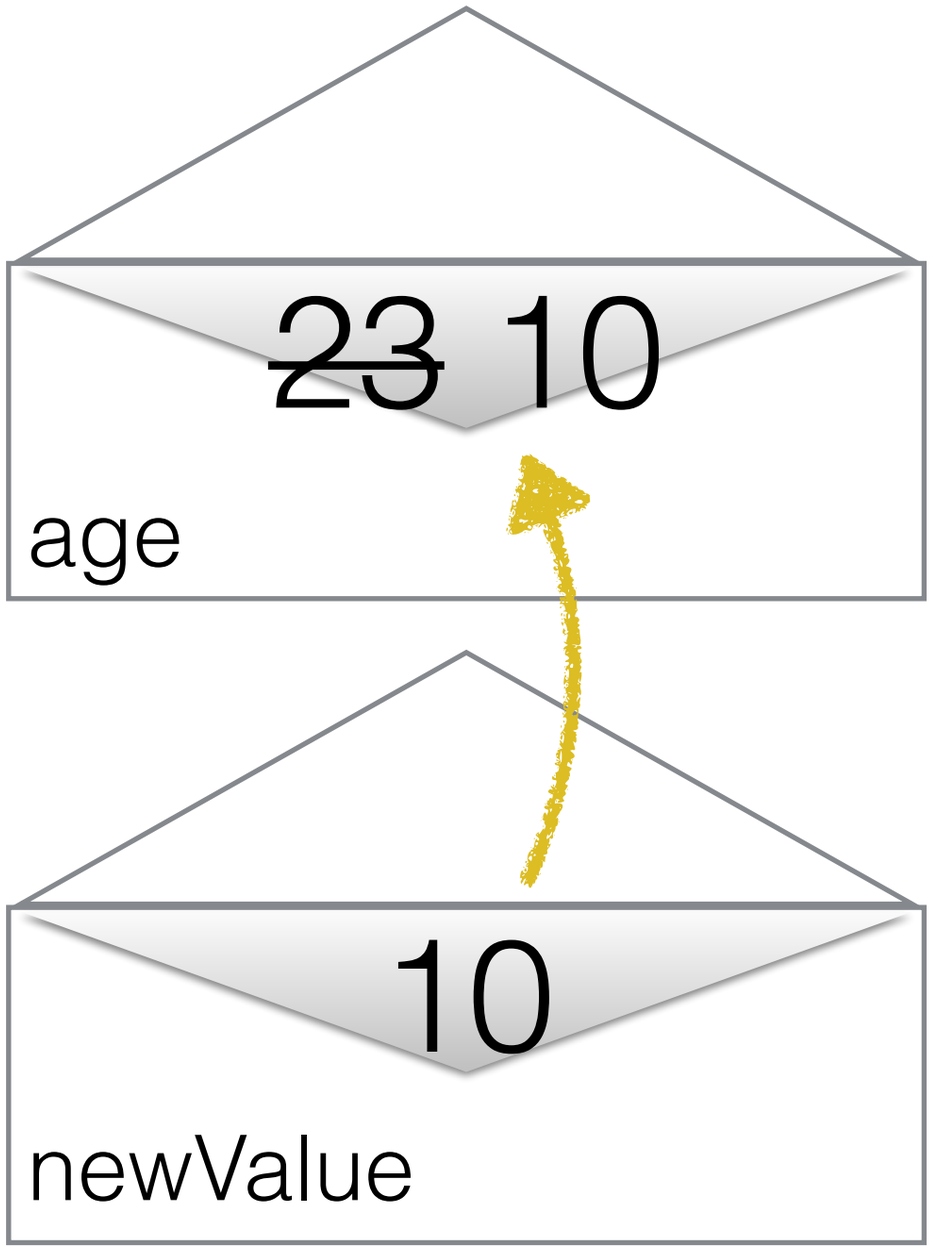
Variables & Expressions



age = 23
newValue = 10

age = newValue

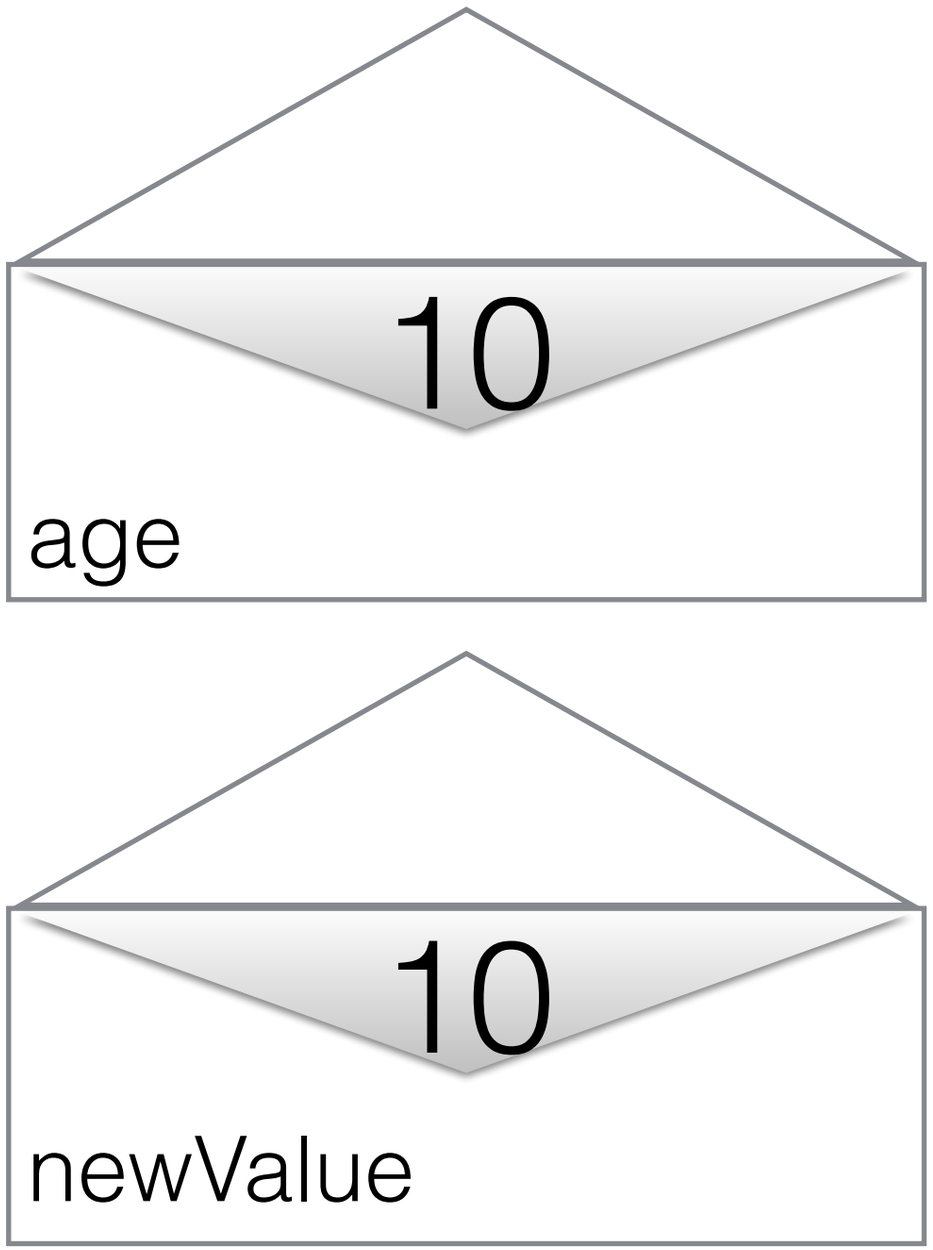
Variables & Expressions



age = 23
newValue = 10

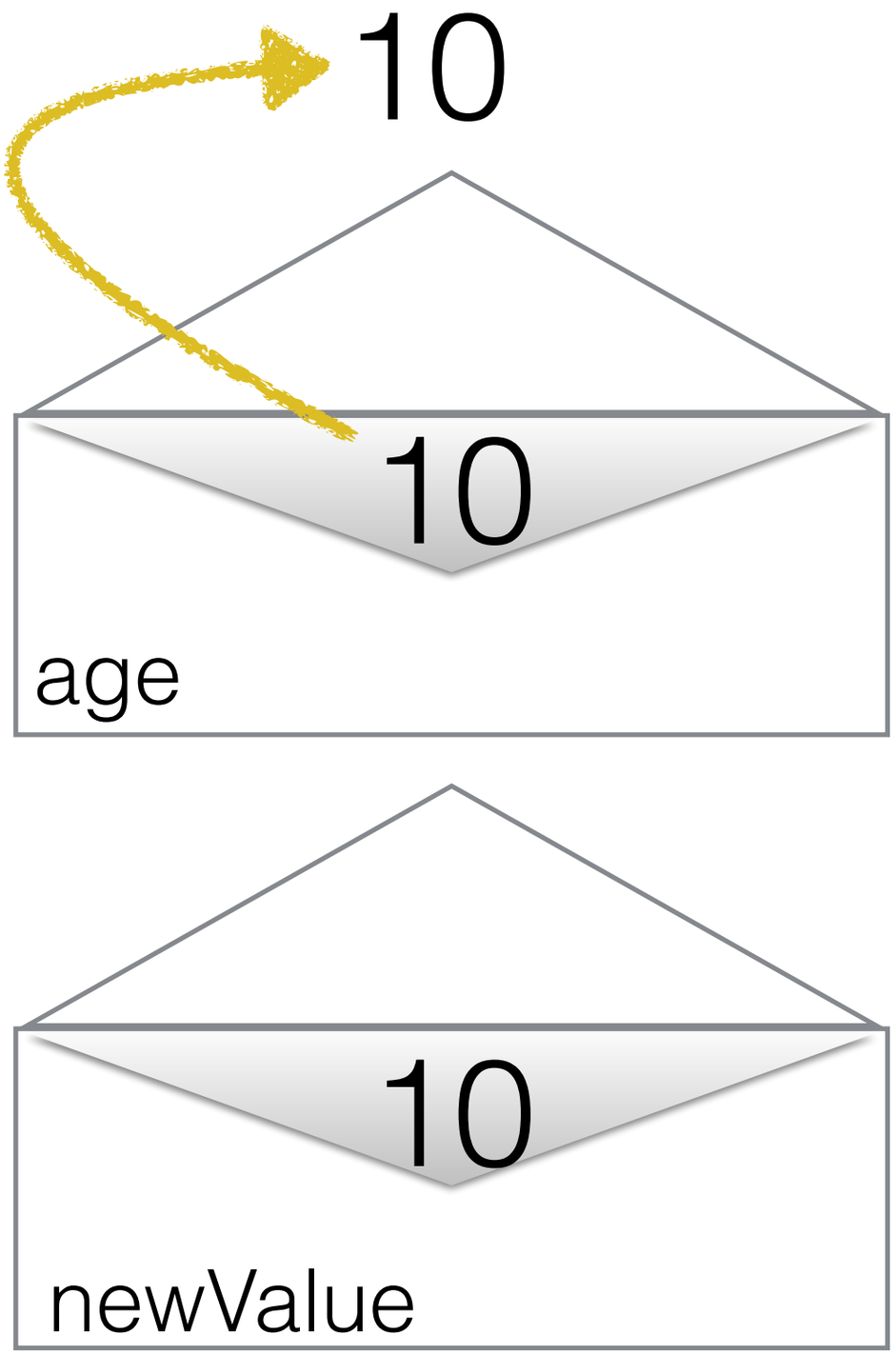
age = newValue

Variables & Expressions



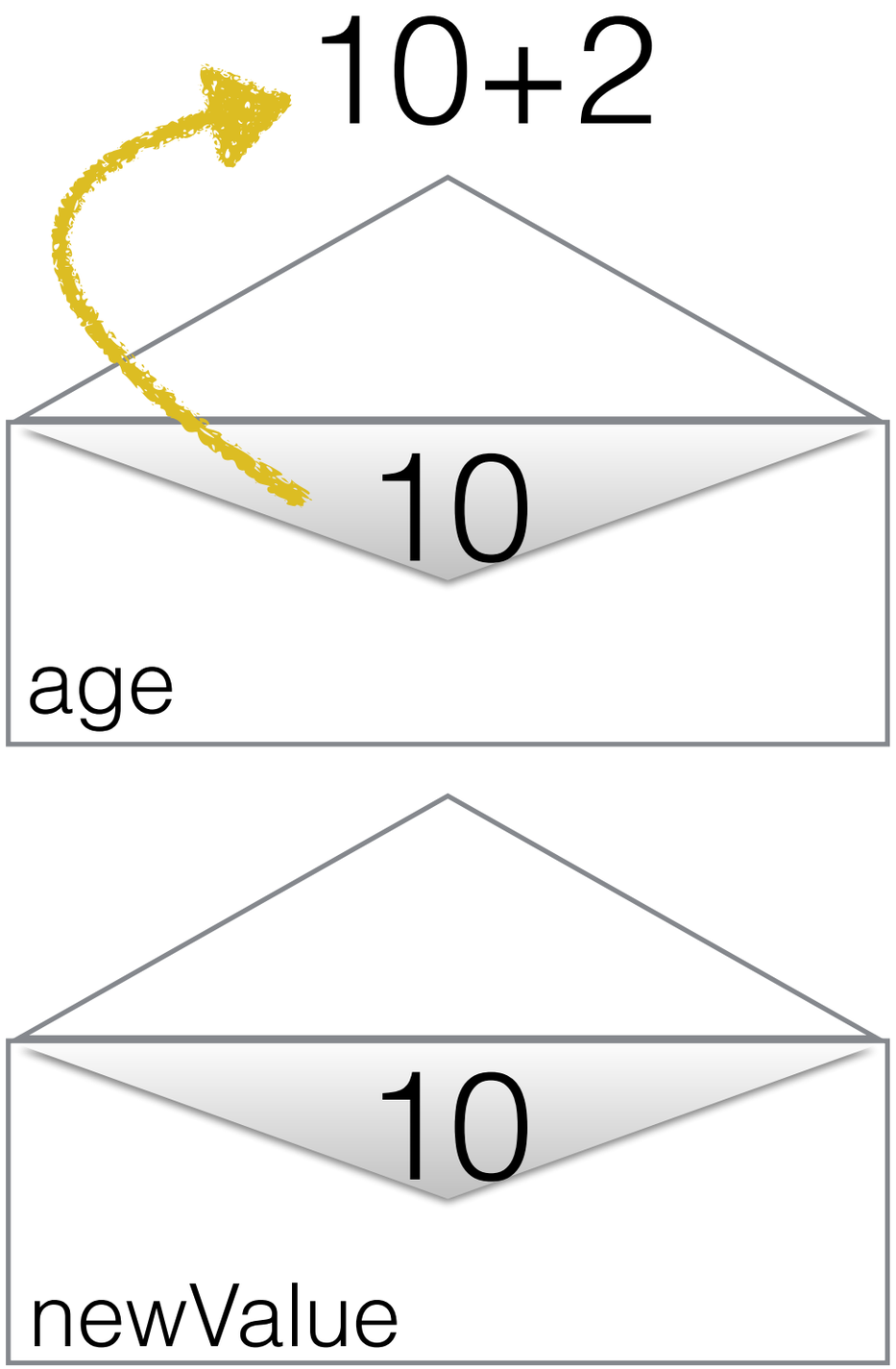
age = 23
newValue = 10
age = newValue
age = age + 2

Variables & Expressions



age = 23
newValue = 10
age = newValue
age = age + 2

Variables & Expressions

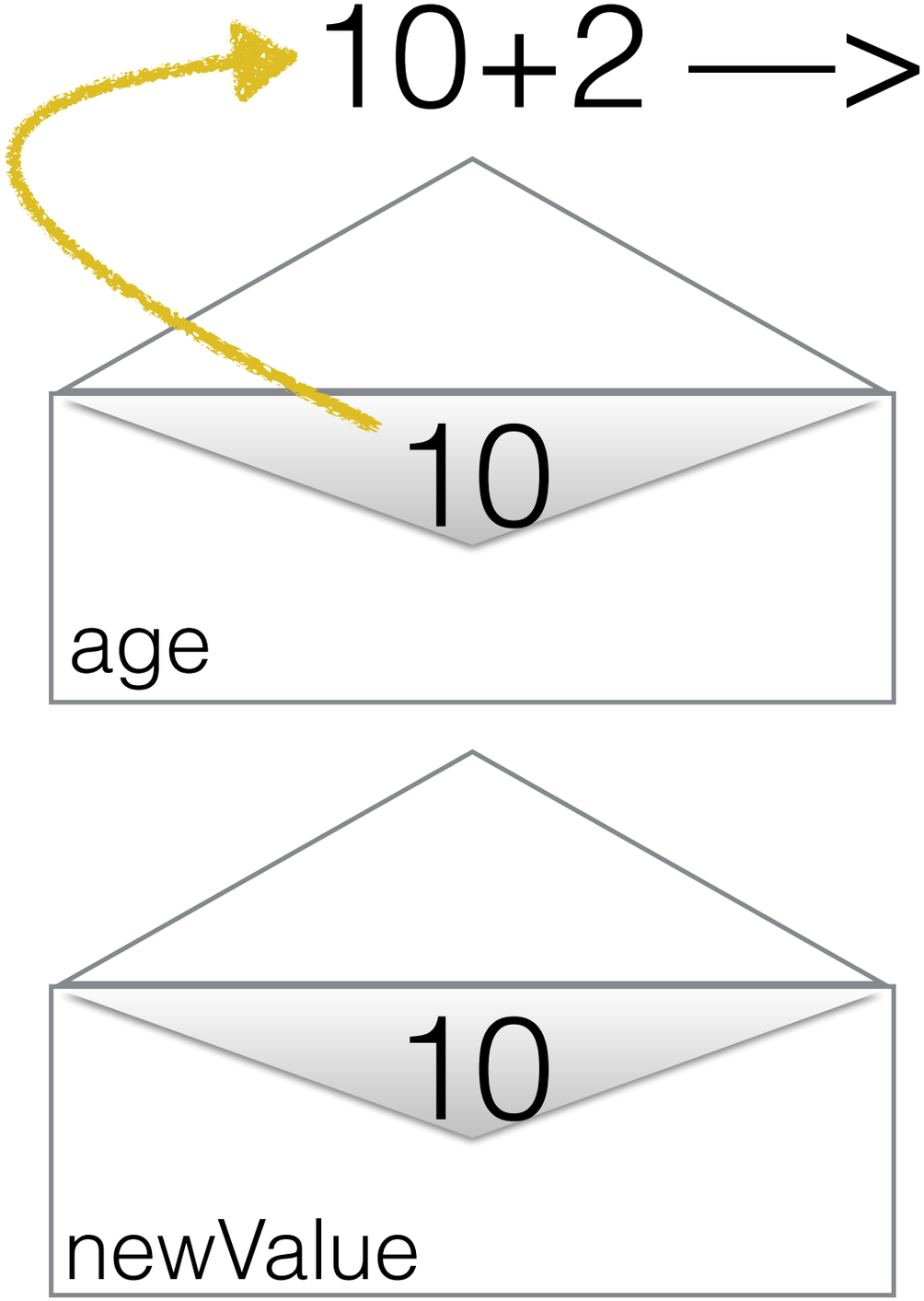


age = 23
newValue = 10

age = newValue

age = age + 2

Variables & Expressions



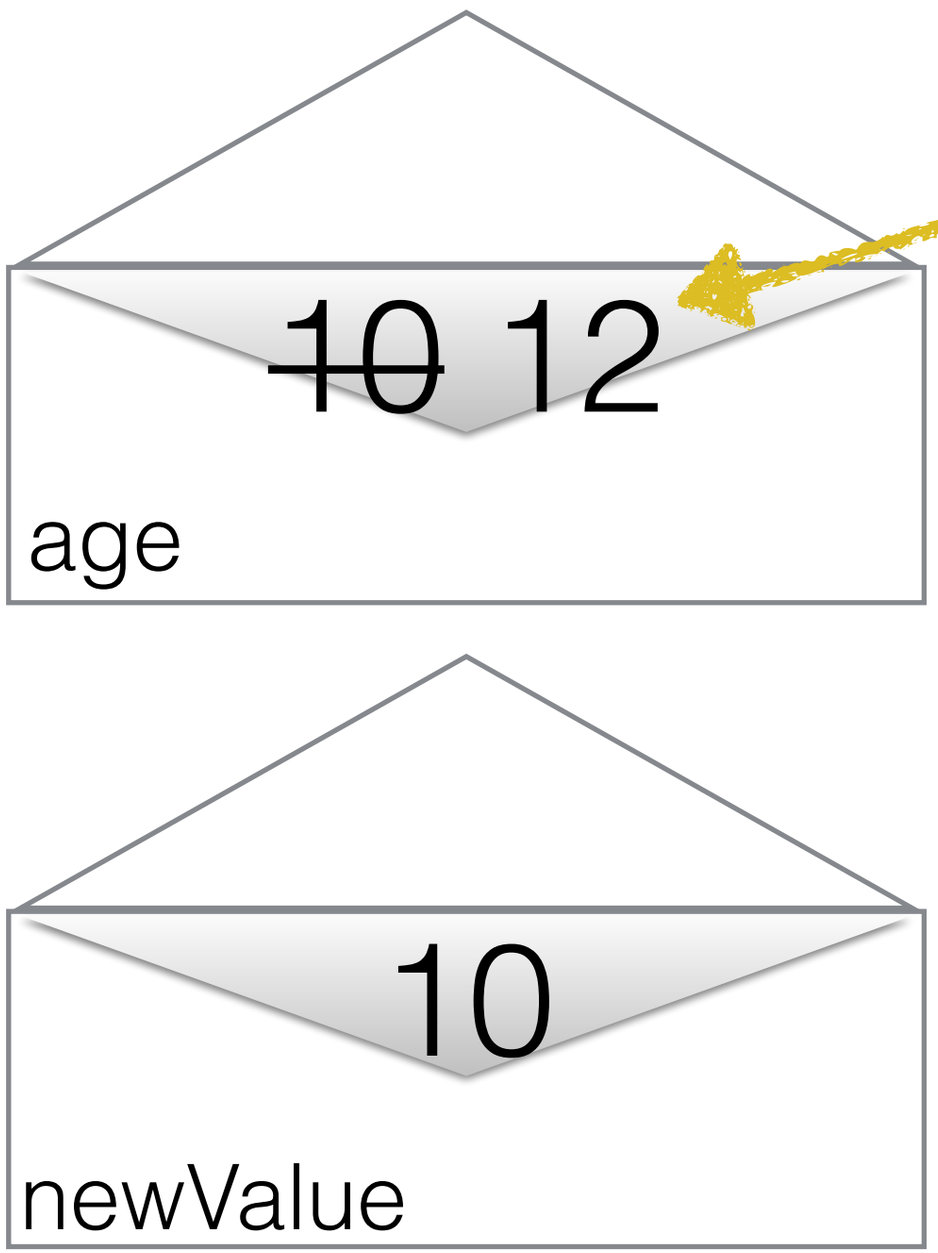
age = 23
newValue = 10

age = newValue

age = age + 2

Variables & Expressions

10+2 —> 12



age = 23
newValue = 10

age = newValue

age = age + 2

Exercise

```
a = 10
b = 20
c = 30
a = b                    # a = ?
```

Exercise

```
a = 10
b = 20
c = 30
a = b                    # a = 20
b = a                    # a = ?     b = ?
```

Exercise

```
a = 10
b = 20
c = 30
a = b                    # a = 20
b = a                    # a = 20    b = 20
c = c * 2                # c = ?
```

Exercise

```
a = 10
b = 20
c = 30
a = b                    # a = 20
b = a                    # a = 20    b = 20
c = c * 2                # c = 60
d = d - 10               # d = ?
```

Exercise

```
a = 10
b = 20
c = 30
a = b                    # a = 20
b = a                    # a = 20    b = 20
c = c * 2                # c = 60
d = d - 10               # NameError:
                         # name 'd' is not defined
```

Naming Variables

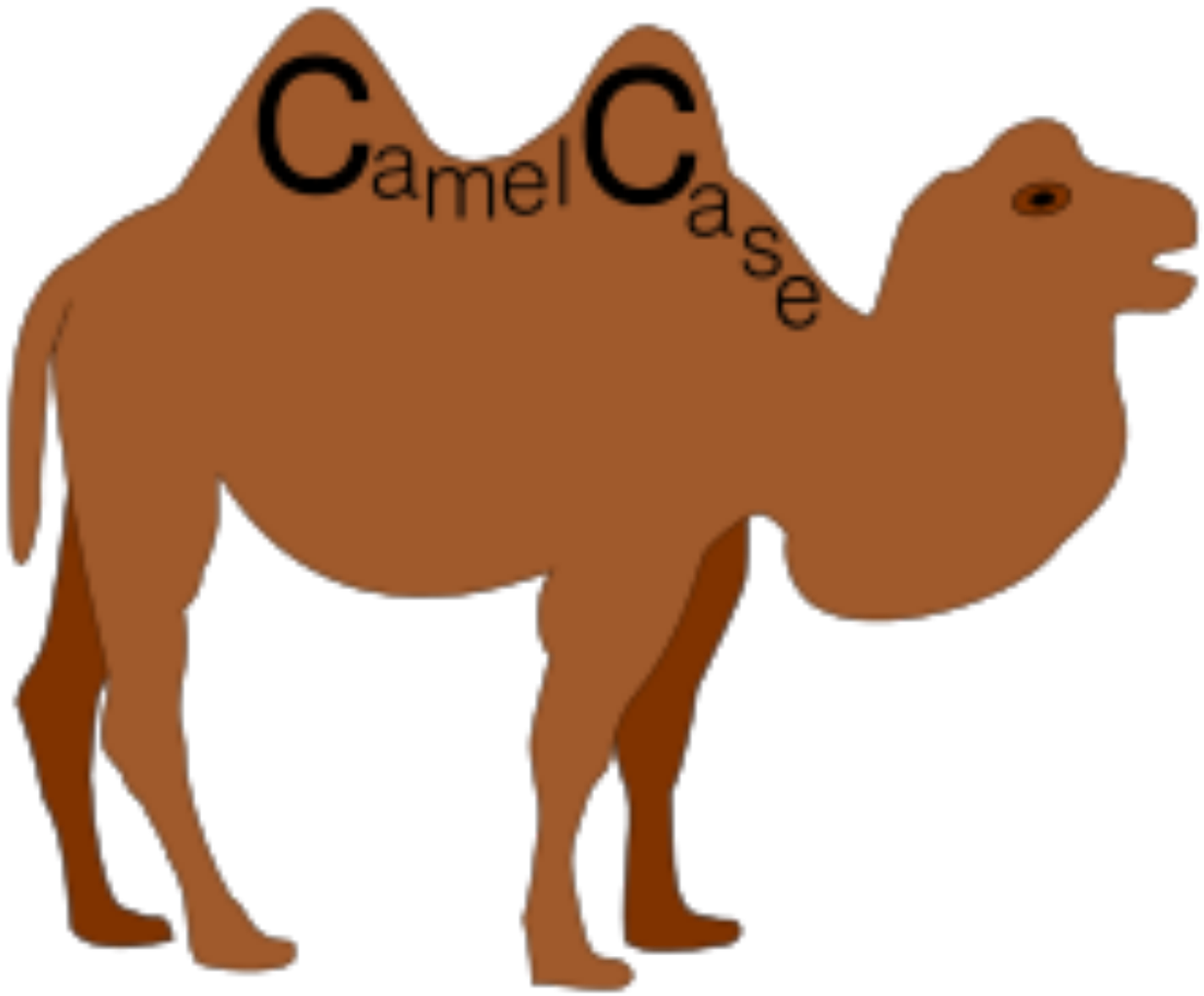
- Variable name cannot be a **keyword**

  and del from not while as elif global or with assert else if pass yield break except import print class exec in raise continue finally is return def for lambda try

- First letter must be **alphabetic** (upper- or lower-case, or underscore)

- Can be followed by 0, 1, or more **letters**, **digits**, or **underscore**

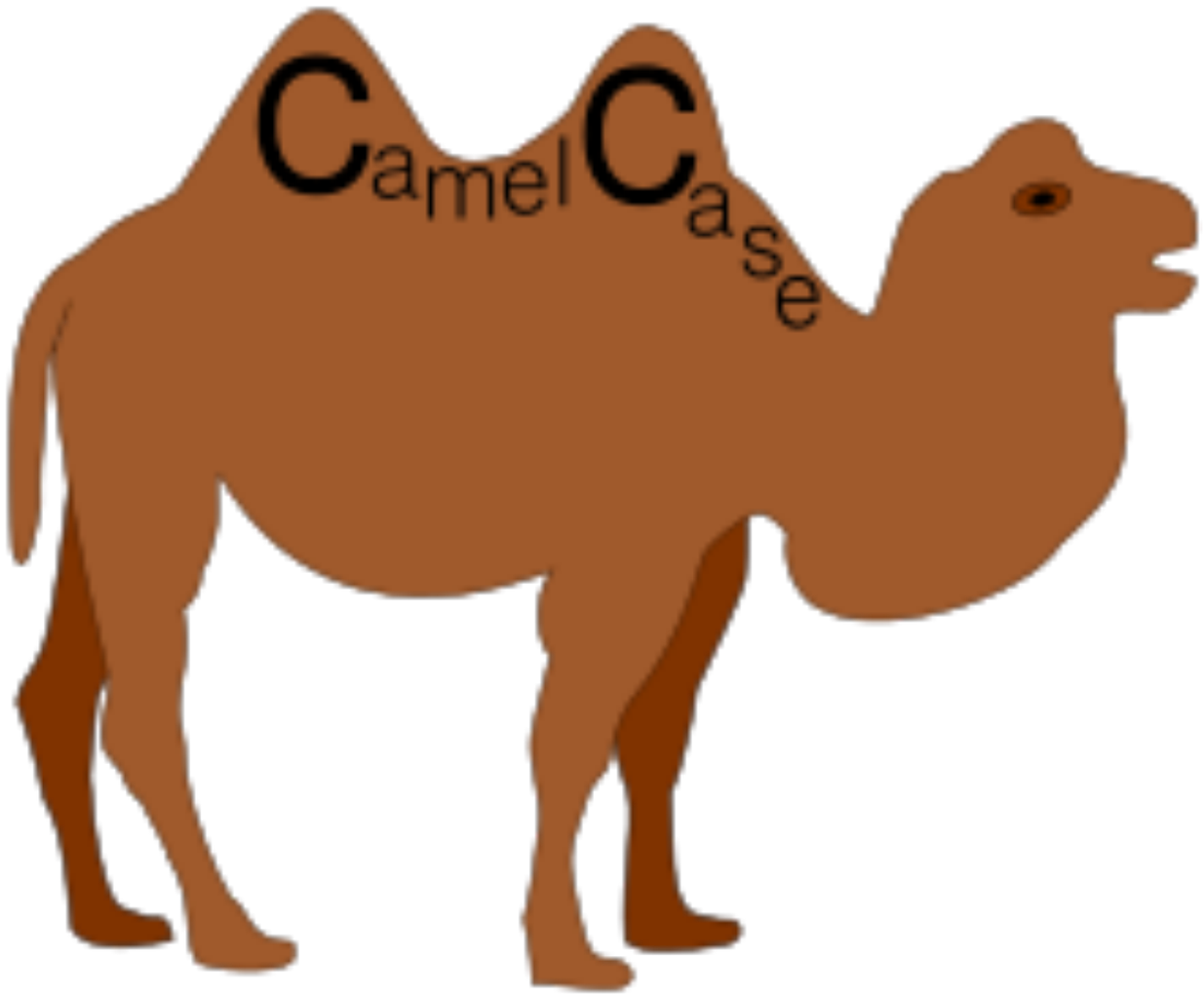
Naming Variables



a
age
delta
name1
name2
R2D2
aVeryLongName

1tooMany

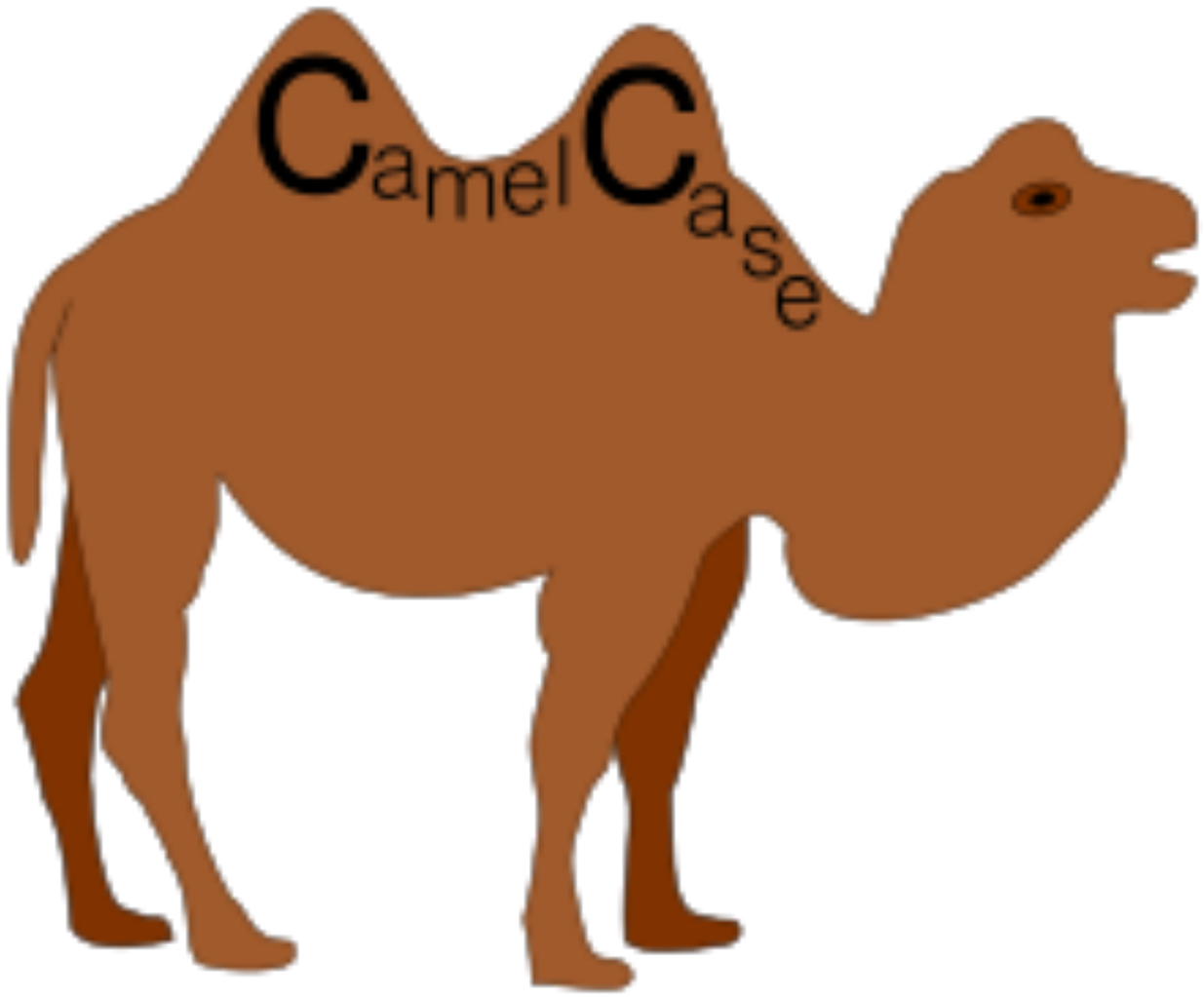
Naming Variables



a
age
delta
name1
name2
R2D2
aVeryLongName

~~1tooMany~~

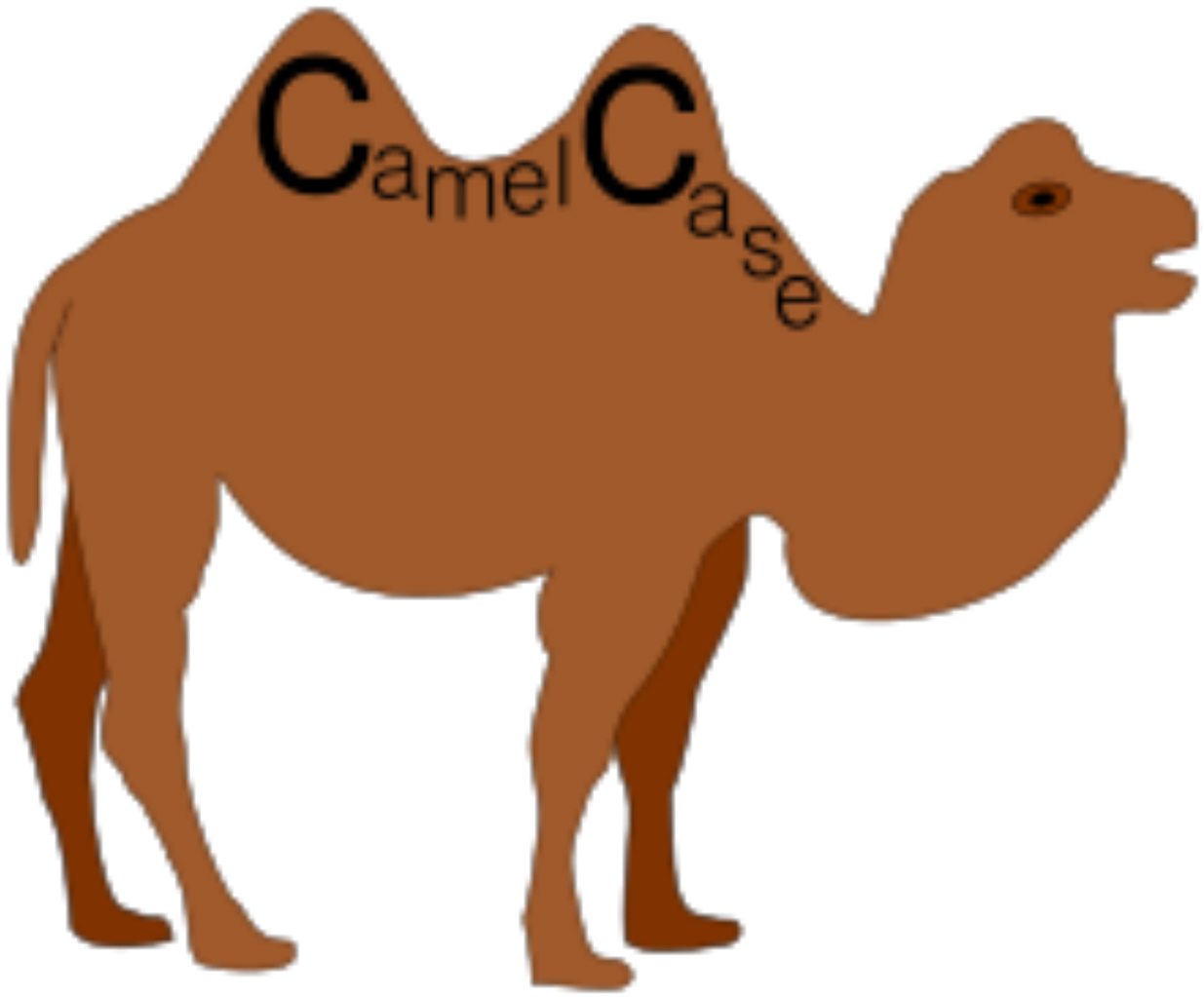
Naming Variables



this_is_good_too
but
wePrefer
thisIsGoodToo

lambda
for
def

Naming Variables



this_is_good_too
but
wePrefer
thisIsGoodToo

~~lambda~~
~~for~~
~~def~~

Exercise 1

```
***
Mae
*****
Alice
*******
Felicia
```

Exercise 2 (Tricky and Unfair)

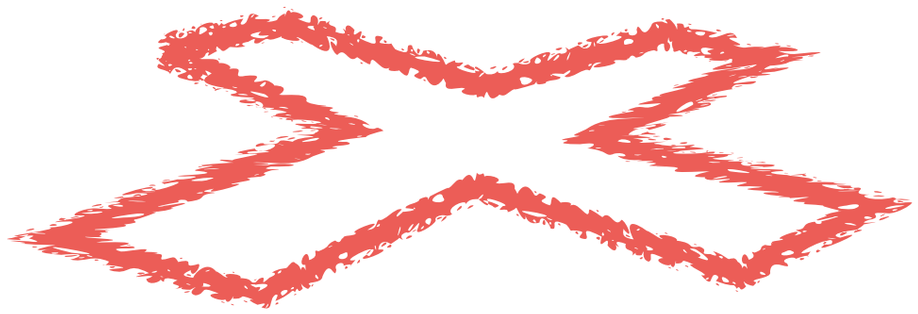
```
*
Mae
*********
Alice
****
Felicia
**
```

We stopped here last time…

Outline

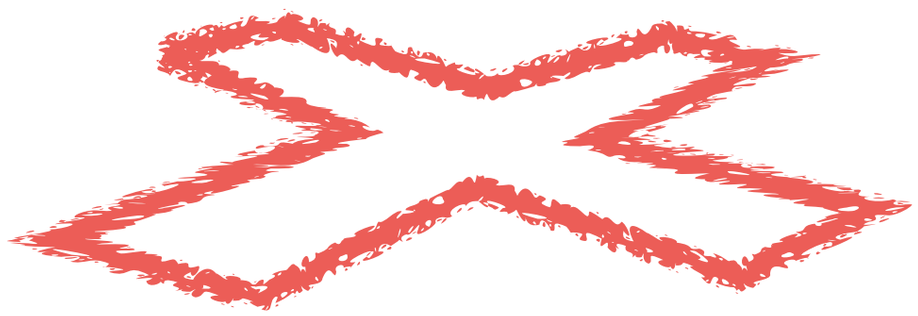
- The Programming Process
- Memory: RAM
- Variables revisited
  - Literals: numbers, strings, lists
  - Types: type( )
    - Multiple assignments
    - Operators. Overloaded operators.
- Loops
  - range( ); list( )
- Programming exercises

The Programming Process



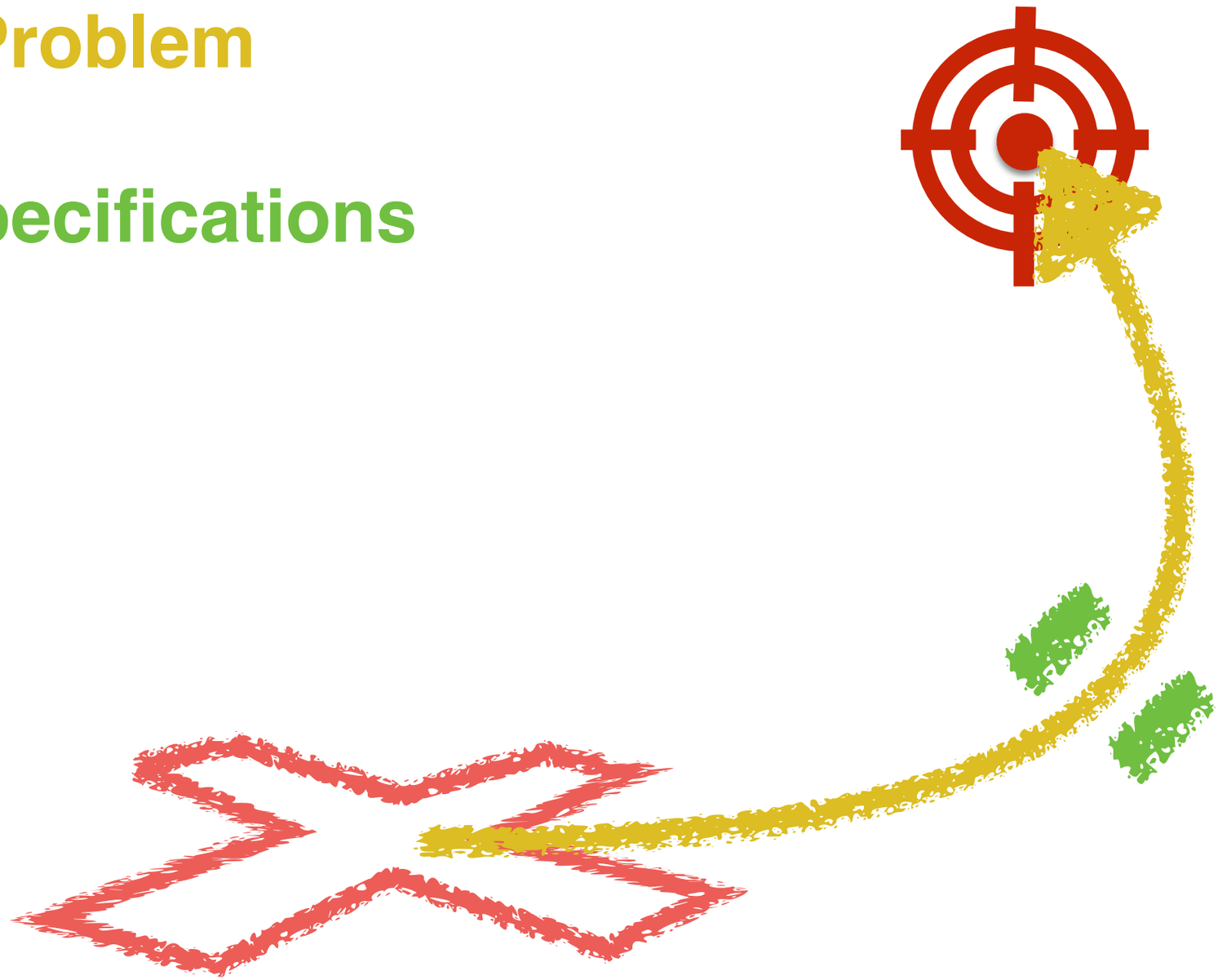
The Programming Process

- Analyze the **Problem**



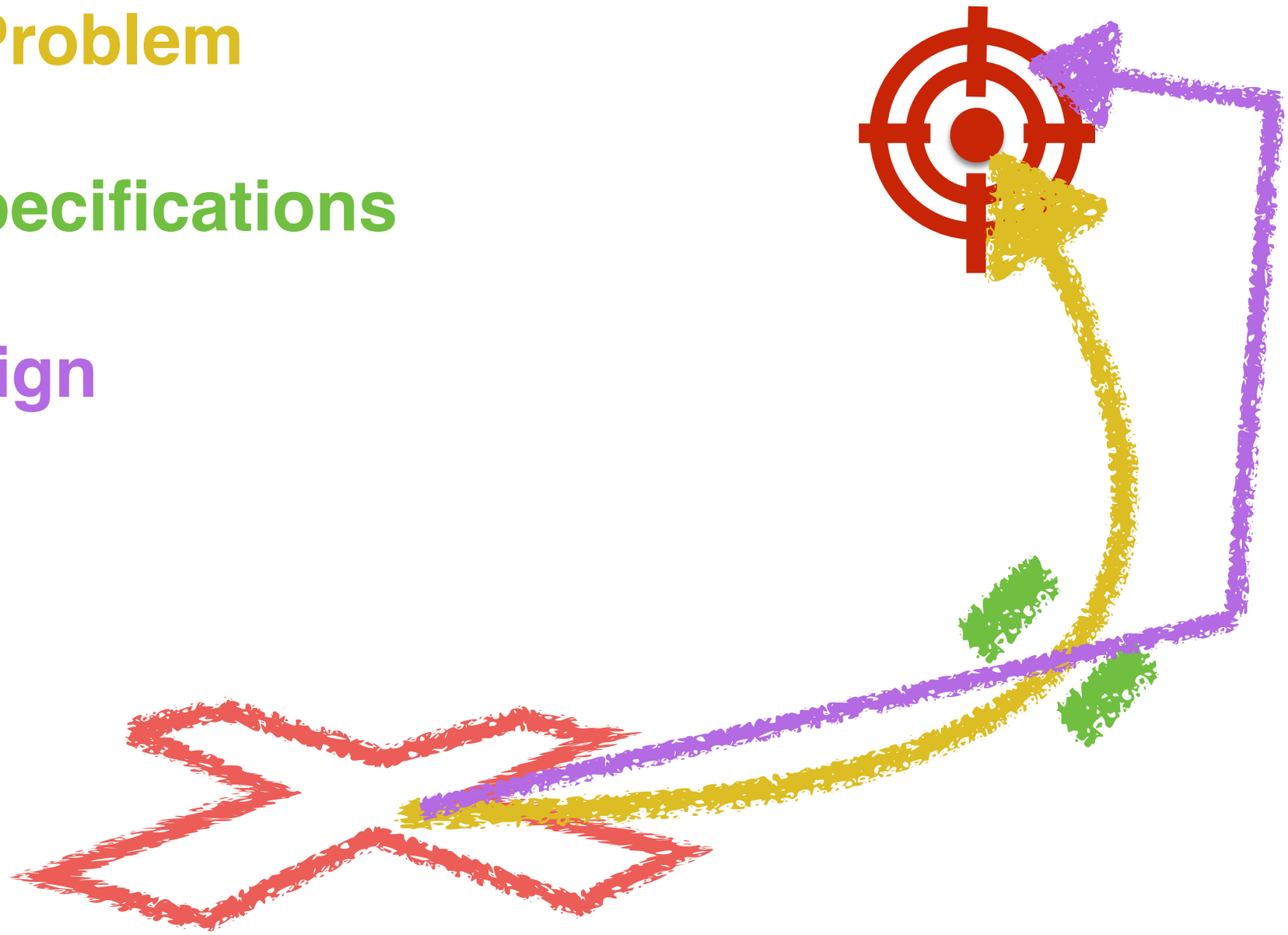
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**



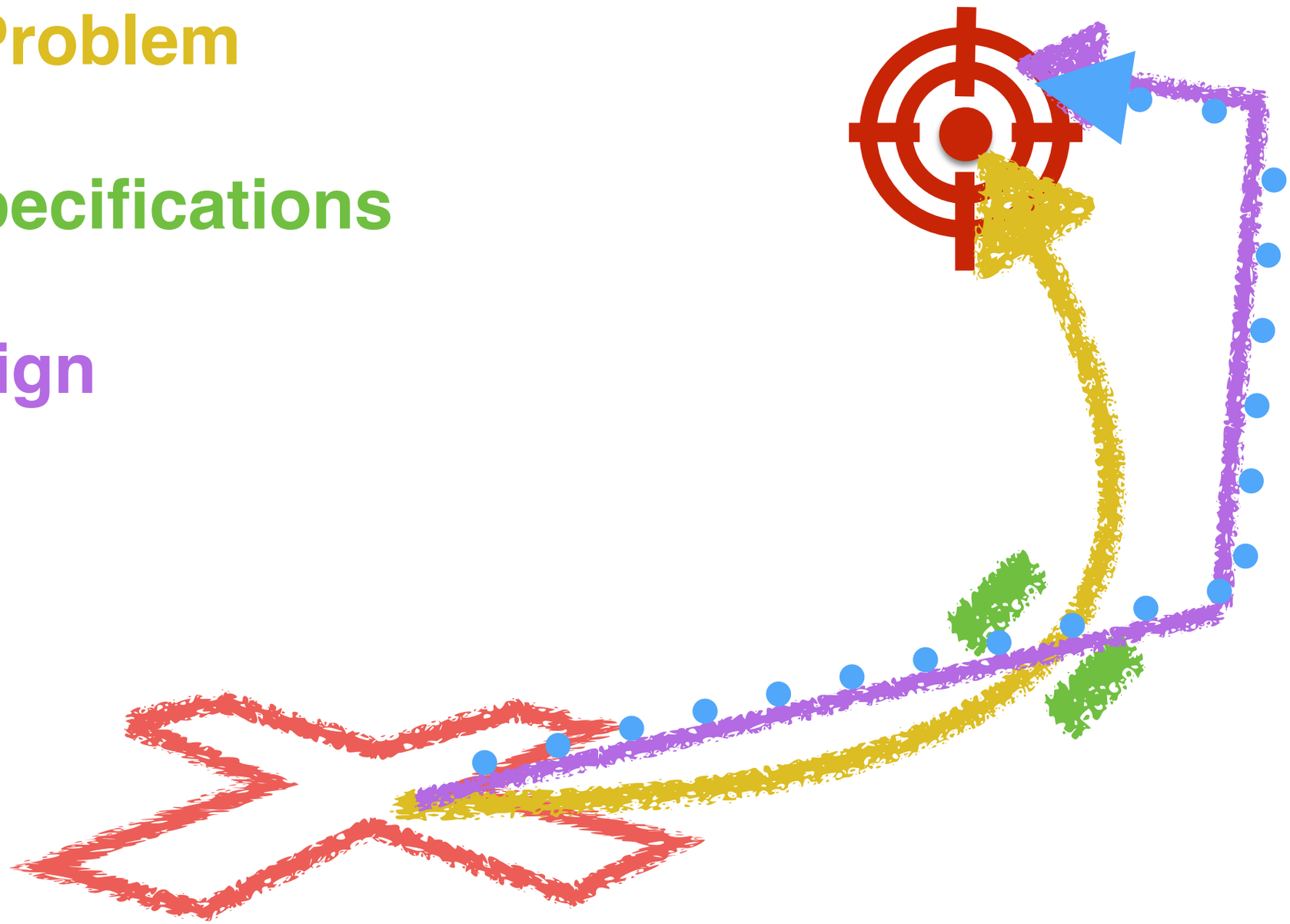
The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**



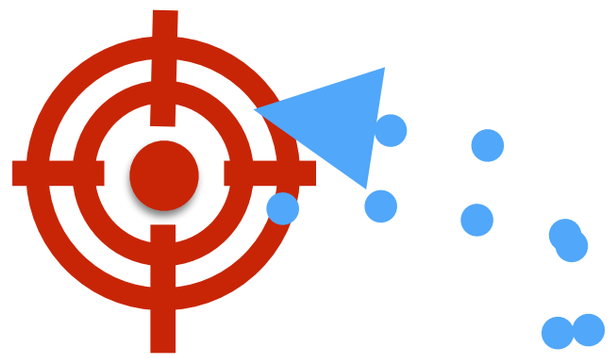
The Programming Process

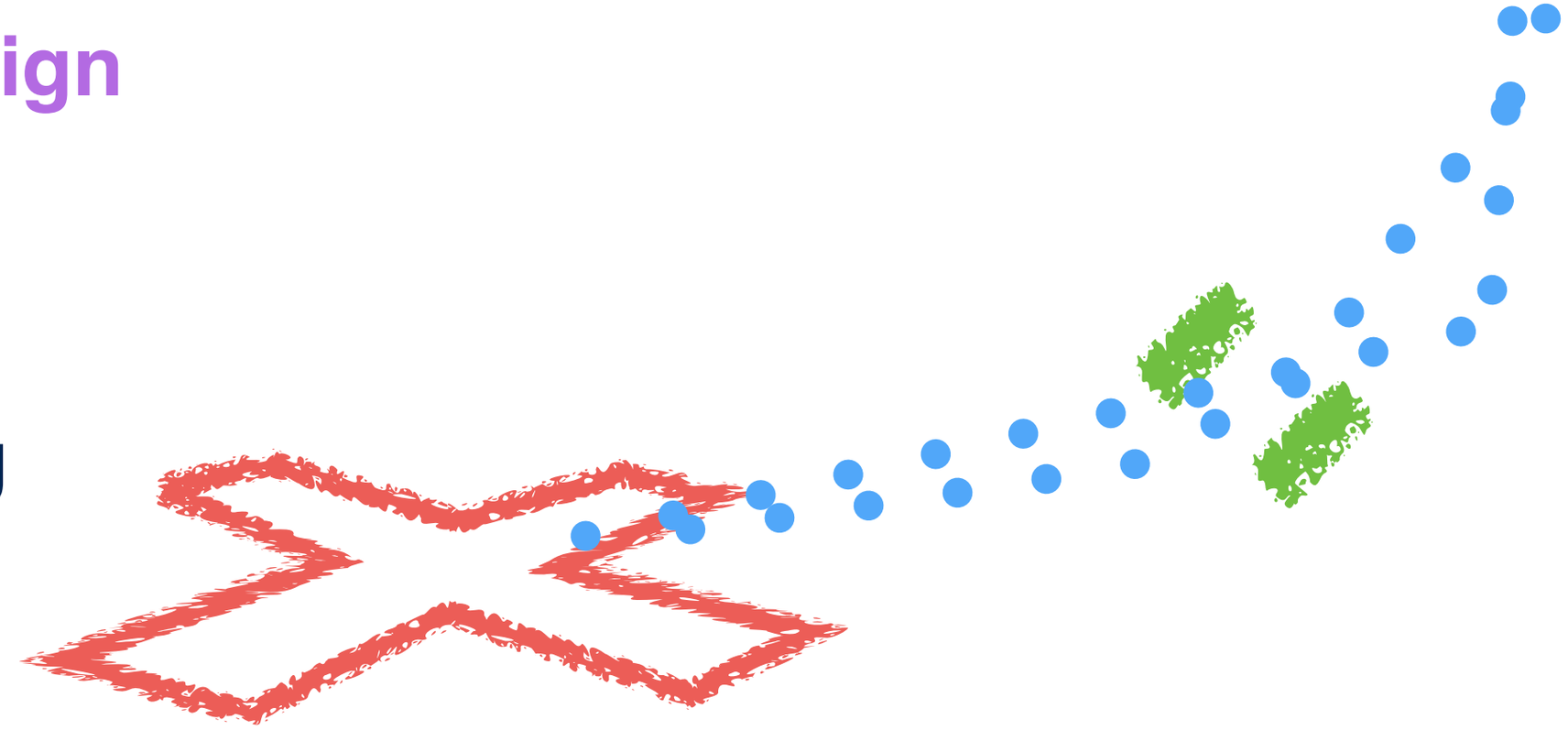
- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**
- **Implement**



The Programming Process

- Analyze the **Problem**
- Determine **Specifications**
- Create a **Design**
- **Implement**
- Test & Debug

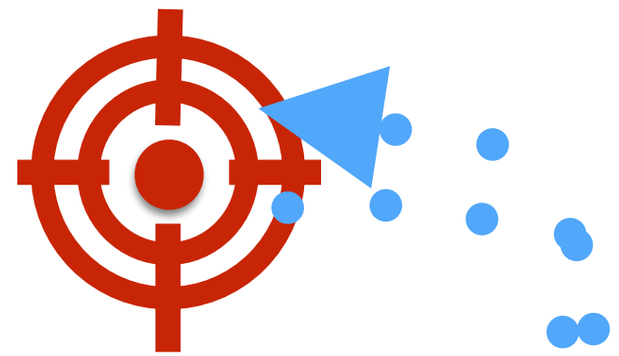


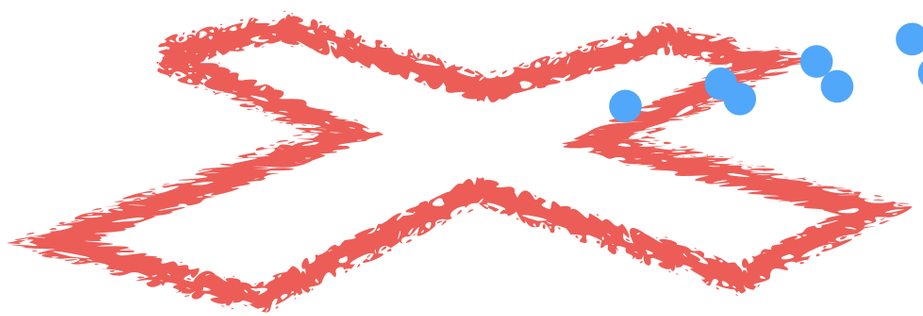


The Programming Process

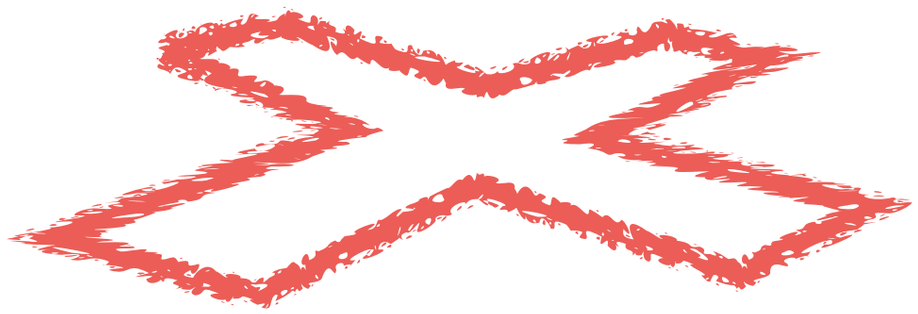
- Analyze the **Problem**
- Determine **Specifications**
- *Refine the*
- ~~Create a~~ **Design**
- **Implement**
- Test & Debug

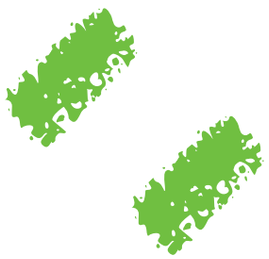
iterate many times





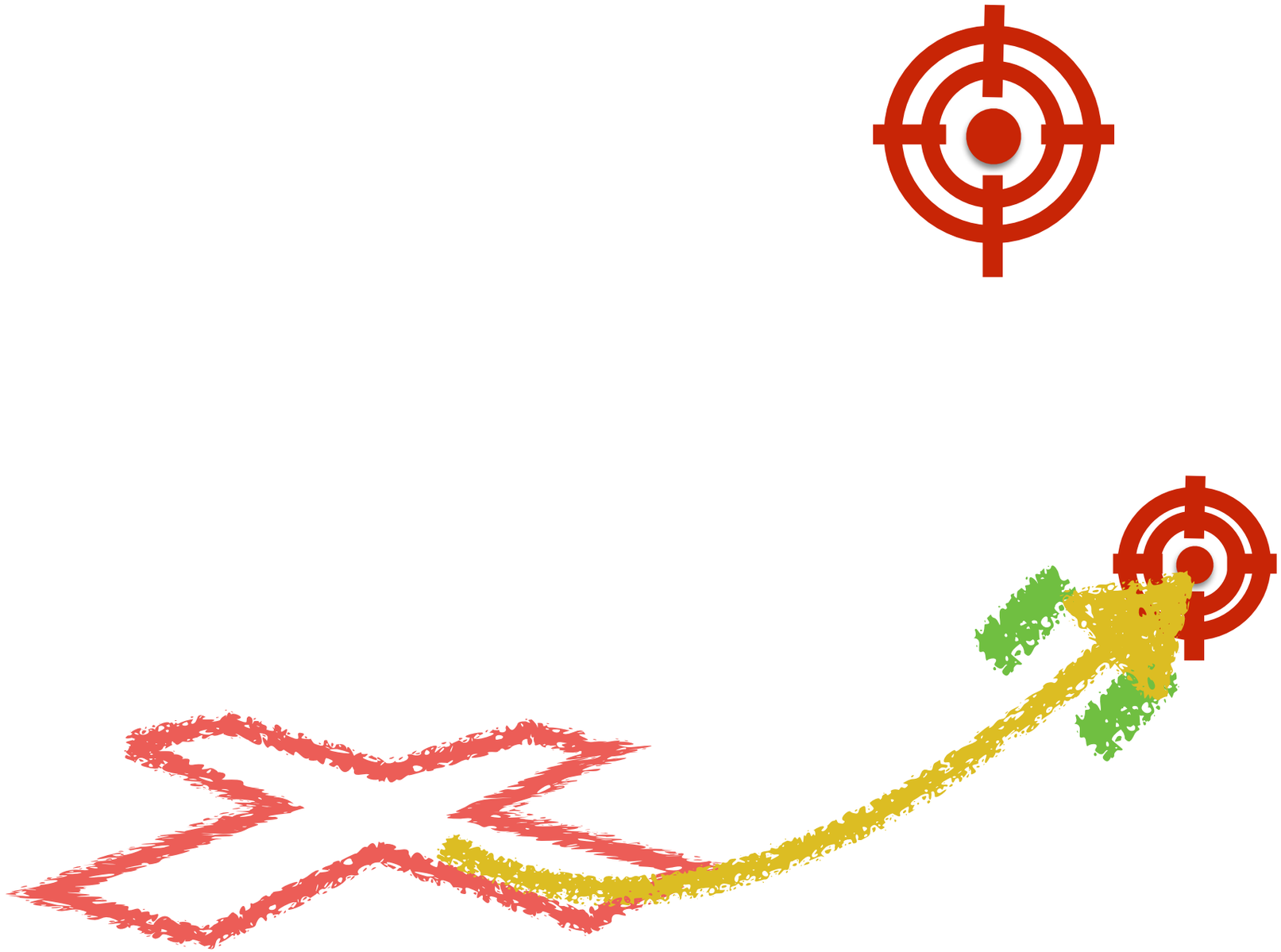
The Programming Process





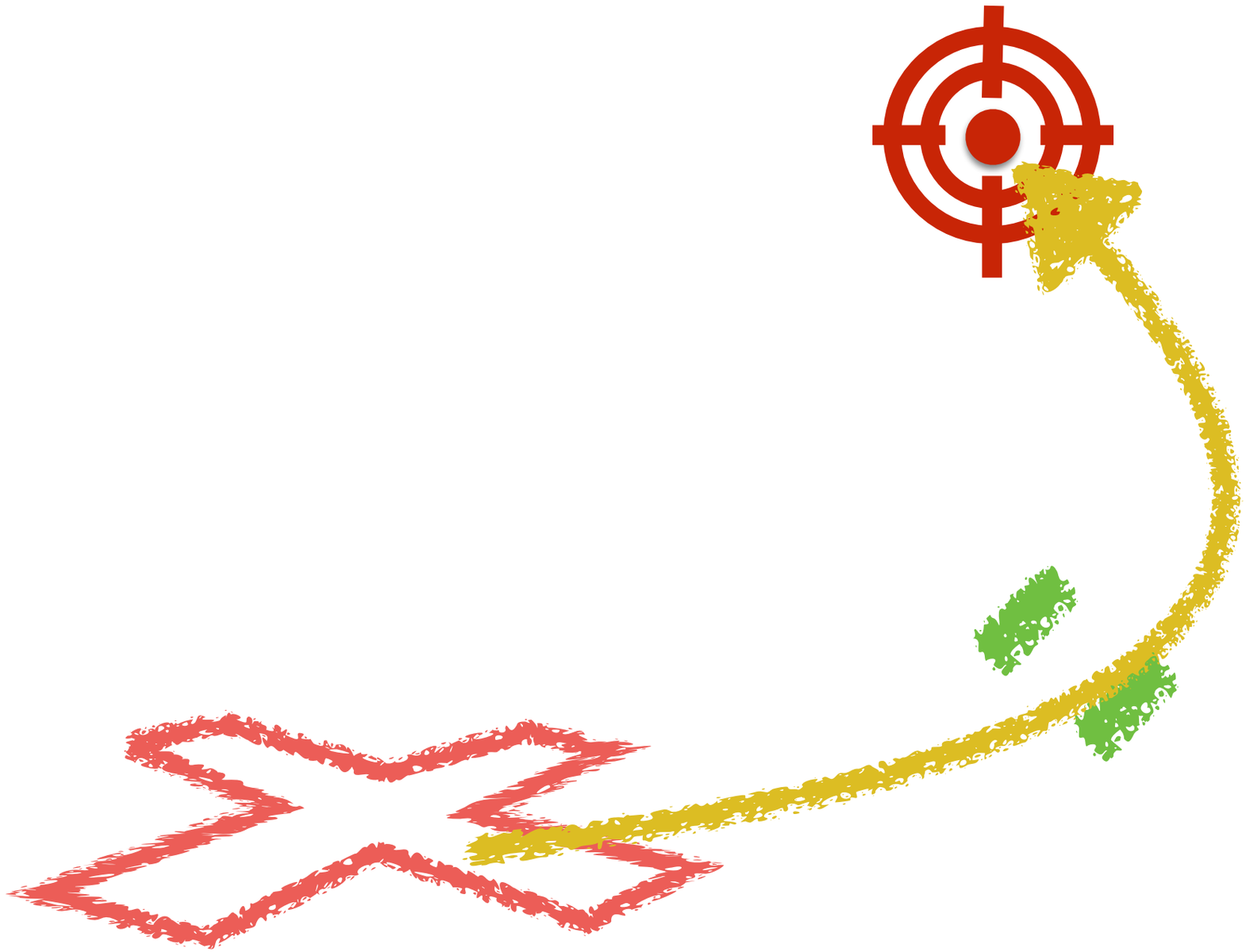
The Programming Process

The Programming Process

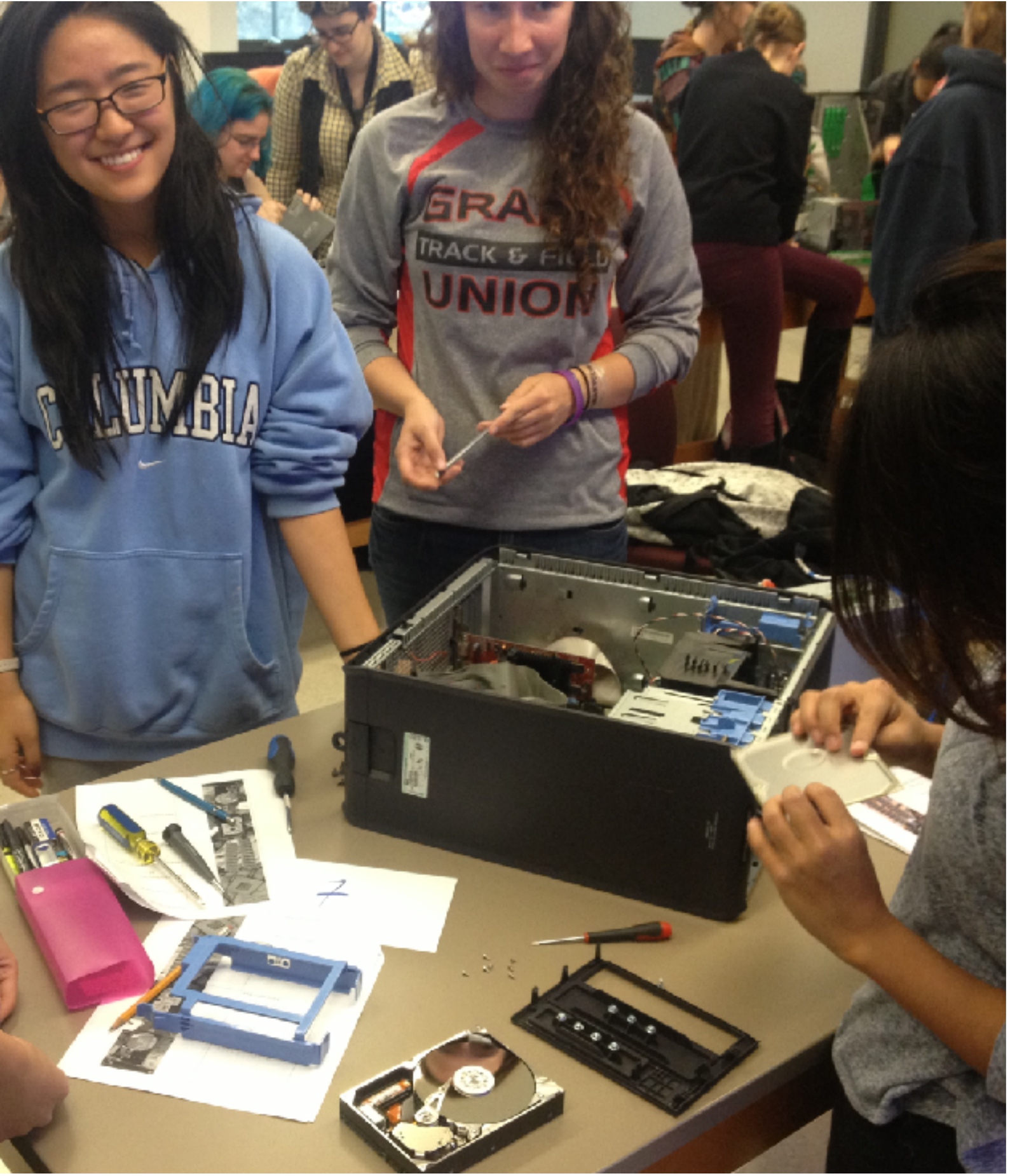


The Programming Process

The Programming Process



Back to the Memory



What does the memory really look like?

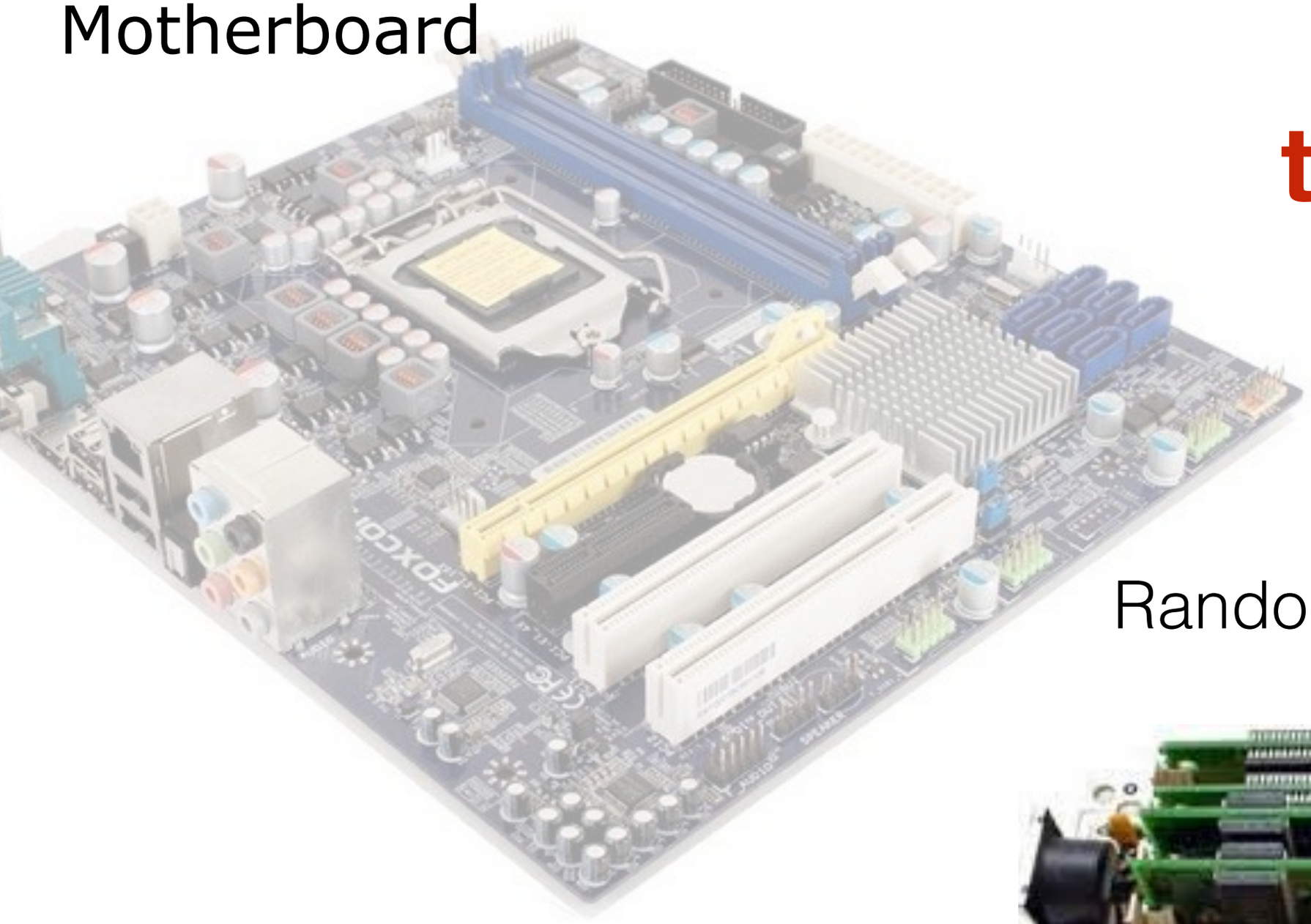
What does the memory really look like?

Motherboard

What does the memory really look like?

What does the memory really look like?

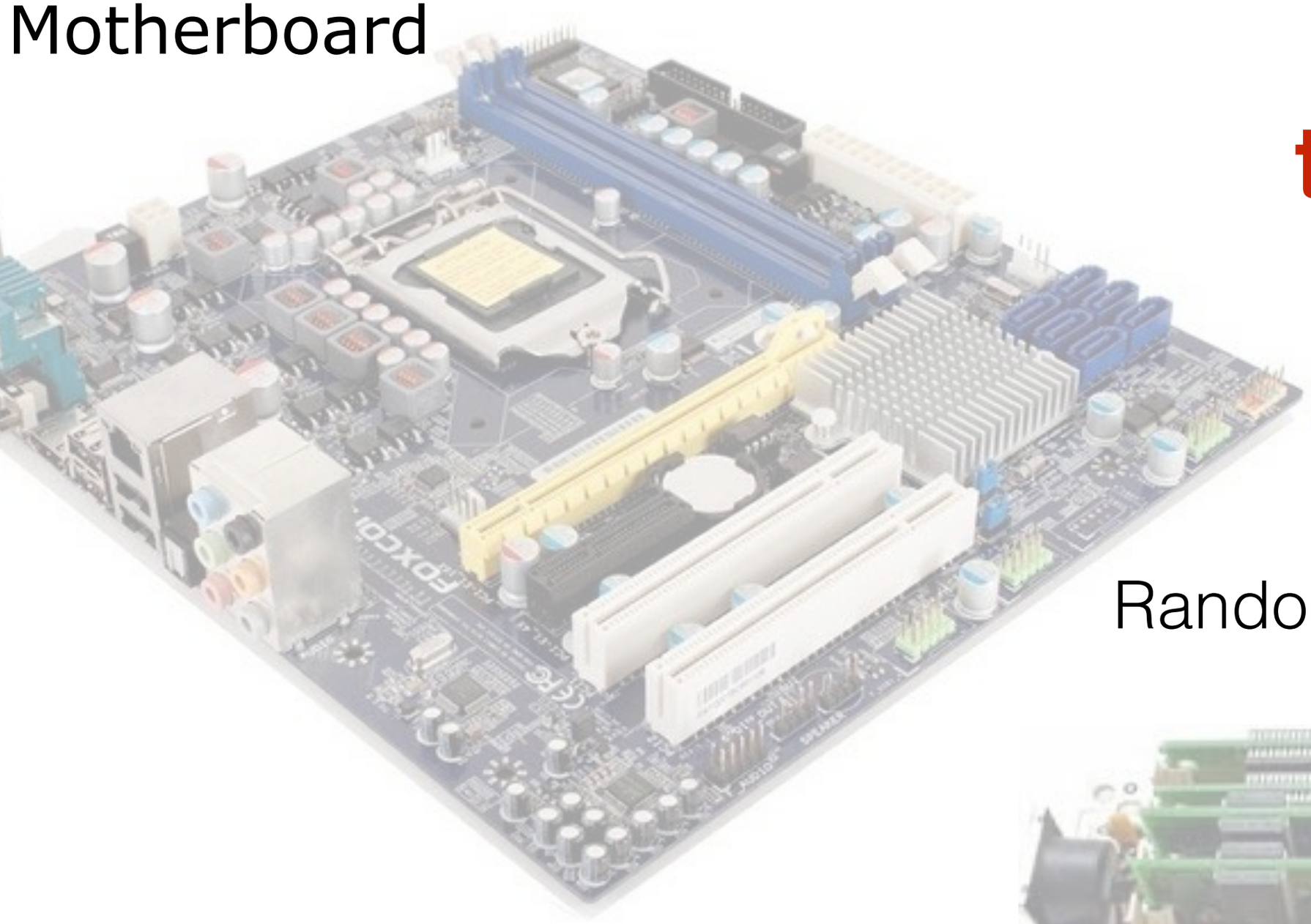
Motherboard



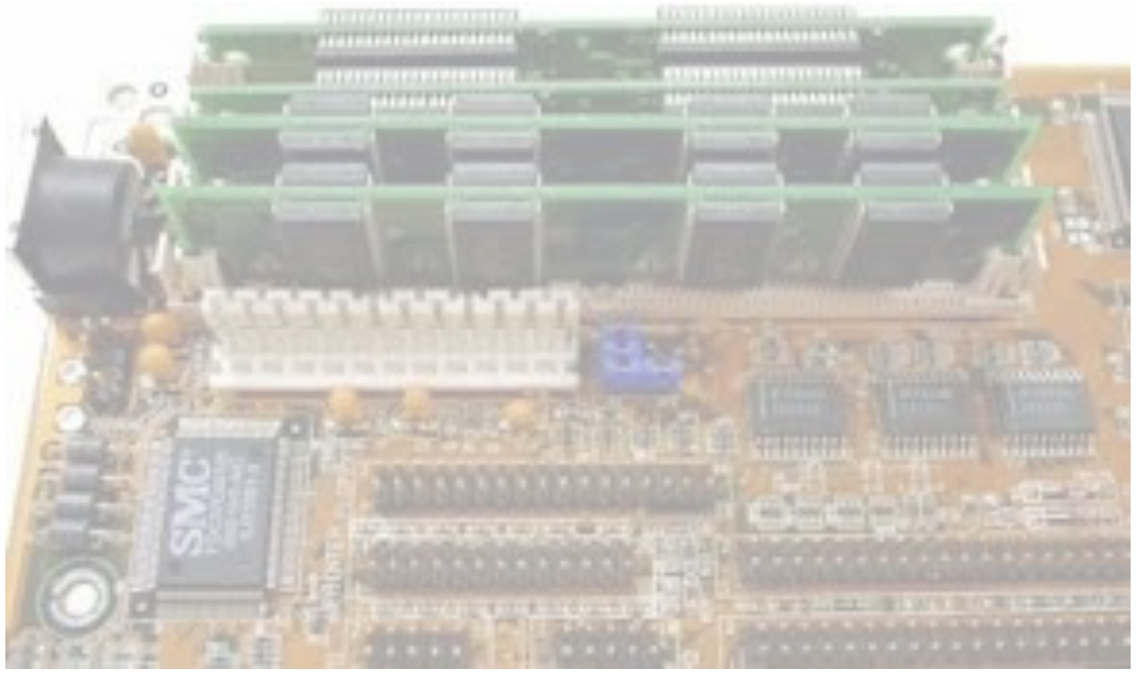
Random Access Memory (RAM)

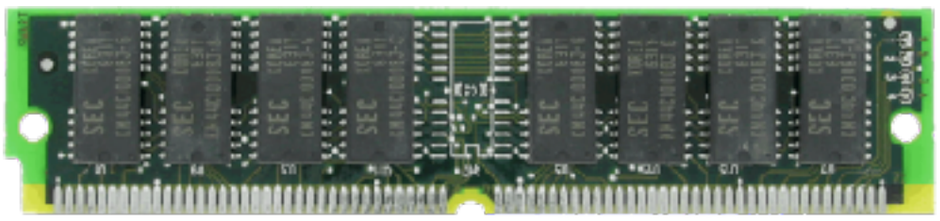
What does the memory really look like?

Motherboard



Random Access Memory (RAM)

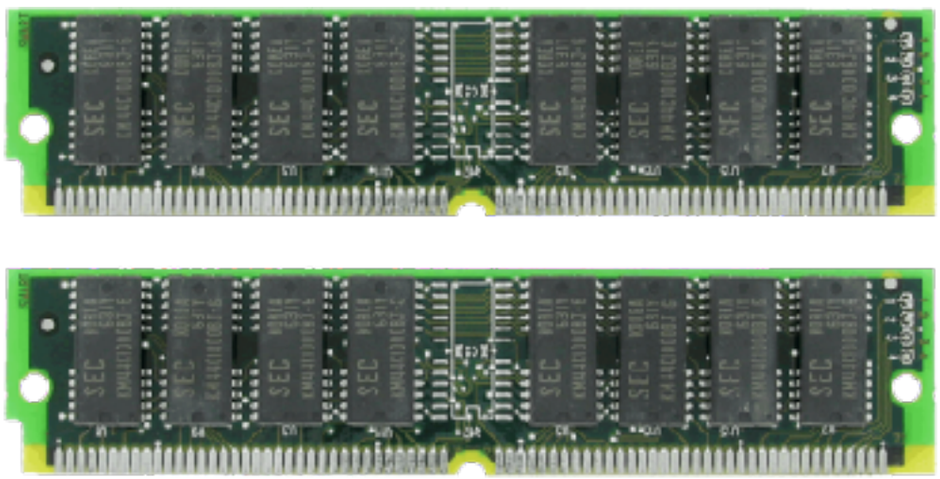




Single In-line Memory Module (SIMM)

What does the memory really look like?

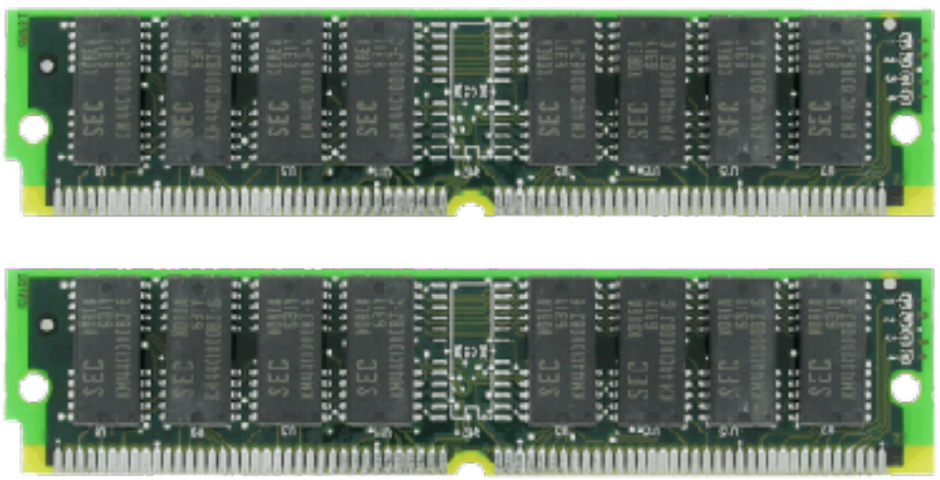
- RAM: 4, 8, 12, 16 GigaBytes
- **Giga** = billion: 10^9 bytes
- In RAM: room for approximately **2 billion** integers



- 1 number takes **4 bytes**
- 1 character takes **1 bytes (sometimes 2 bytes)**

How big is 2 Billion?

2 billion integers

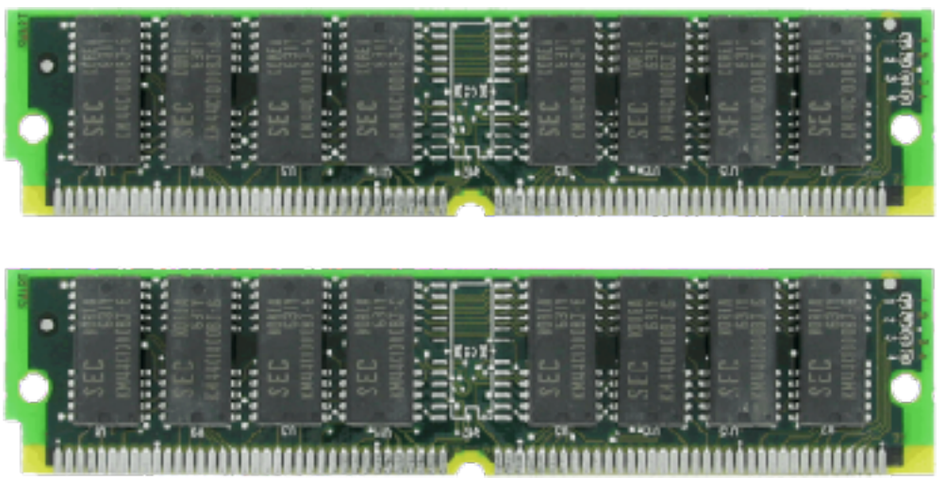


How tall are 2 billion quarters

How big is 2 Billion?

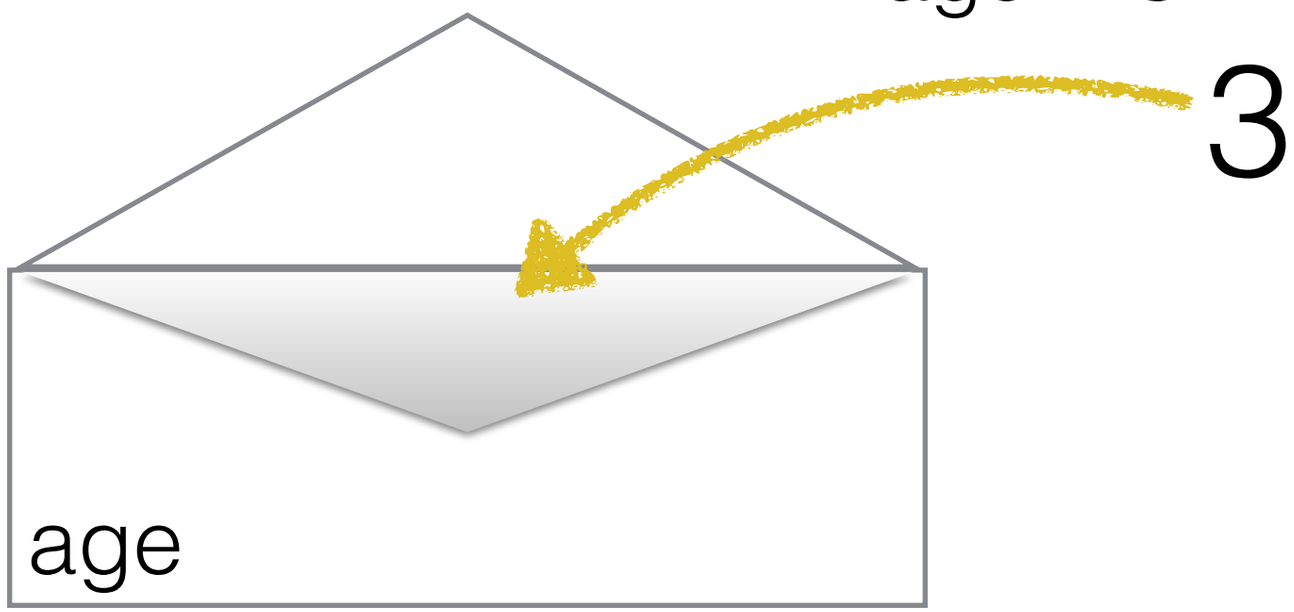
2 miles, or 3.2 km !

2 billion integers



How tall are 2 billion quarters

Variables and Assignment



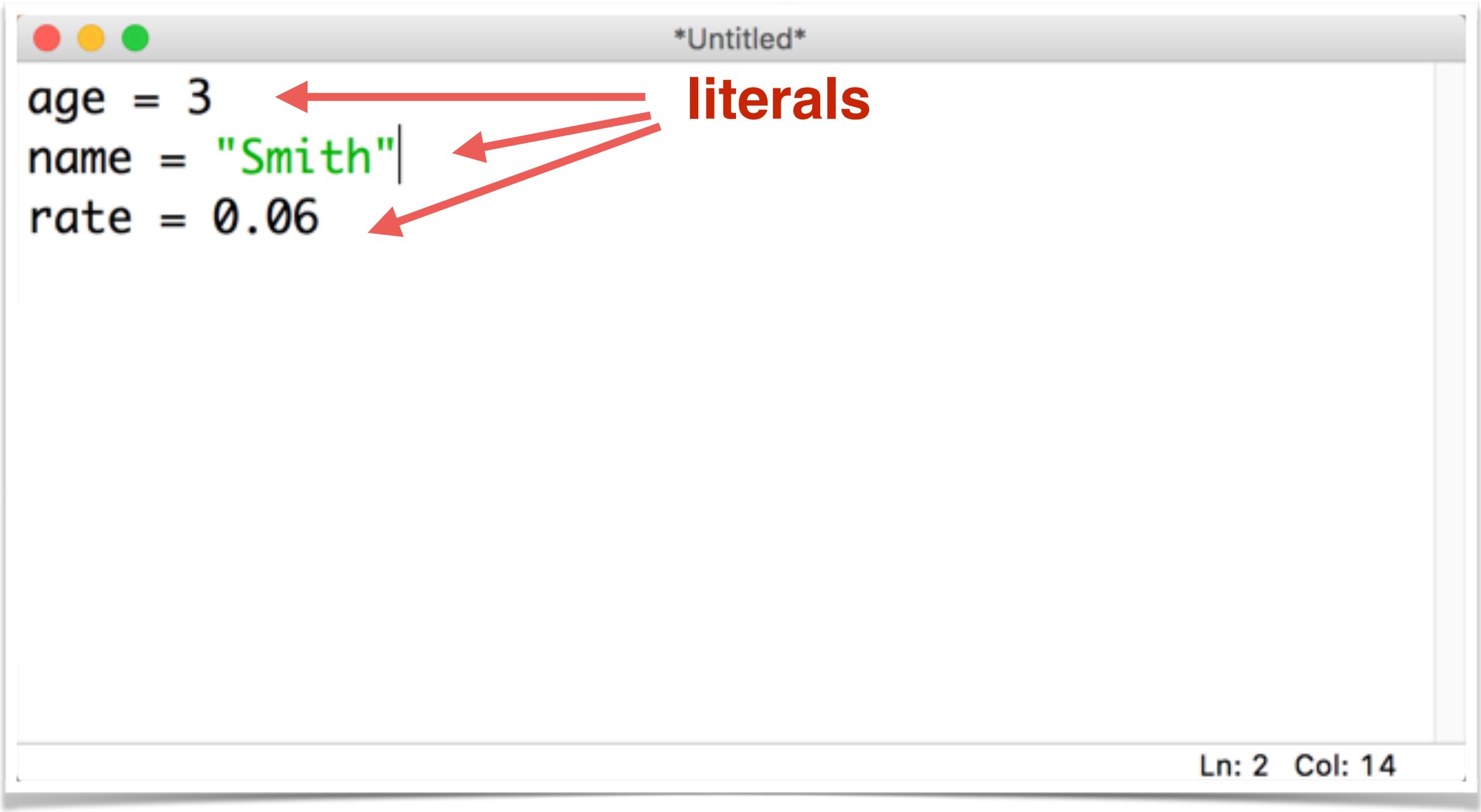
Variables and Assignment

Untitled

```
age = 3
name = "Smith"
rate = 0.06
```

Ln: 2 Col: 14

Variables and Assignment



Variables and Assignment

Untitled

```
age = 3
name = "Smith"
rate = 0.06

age = age * 2                    # double the age
age = age + 1                    # increment the age
```

Ln: 2 Col: 14

Variables and Assignment

Untitled

```
age = 3
name = "Smith"
rate = 0.06

age = age * 2                    # double the age
age = age + 1                    # increment the age

name = name + " College"         # name will contain
                                 # "Smith College"
```

Ln: 2 Col: 14

Variables and Assignment

Untitled

```
age = 3
name = "Smith"
rate = 0.06

age = age * 2                    # double the age
age = age + 1                    # increment the age

name = name + " College"         # name will contain
                                 # "Smith College"
```

Col: 14

In a programming language operators may have different meanings depending on the *context*

Variables and Assignment

Untitled

```
age = 3
name = "Smith"
rate = 0.06

age = age * 2                    # double the age
age = age + 1                    # increment the age

name = name + " College"         # name will contain
                                 # "Smith College"
```

Overloaded operators

Ln: 2 Col: 14

Exercises

Exercises

Guess what Python will do

Untitled

```
age = 3
name = "Smith"
rate = 0.06

age = age * rate
```

Ln: 1 Col: 5

Exercises

Guess what Python will do

Untitled

```
age  = 3
name = "Smith"
rate = 0.06

age = age * rate       # age will contain 0.18
name = "his + hers"    # name will contain "his + hers"
rate = name * rate
```

Ln: 1 Col: 5

Exercises

Guess what Python will do

Untitled

```
age  = 3
name = "Smith"
rate = 0.06

age = age * rate       # age will contain 0.18
name = "his + hers"    # name will contain "his + hers"
rate = name * rate     # TypeError: can't multiply
                       # sequence by 'float'
```

Ln: 1 Col: 5

Exercises

Guess what Python will do

Week1Friday.py - /Users/thiebaut/Desktop/Week1Friday.py (3.5.4)

```
name = "Smith"
col  = name + " College" * 2

print( col )

# output
```

Ln: 10 Col: 0

Exercises

Guess what Python will do

Week1Friday.py - /Users/thiebaut/Desktop/Week1Friday.py (3.5.4)

```
name = "Smith"
col  = name + " College" * 2

print( col )

# output
```

Ln: 10 Col: 0

Exercises

Guess what Python will do

Week1Friday.py - /Users/thiebaut/Desktop/Week1Friday.py (3.5.4)

```
name = "Smith"
col  = name + " College" * 2

print( col )

# output
# Smith College College
```

Ln: 10 Col: 0

Using the Shell…

```
name = "Smith
col  = ( name

print( col )

# output
# Smith Colle
```

Python 3.5.4 Shell

```
Python 3.5.4 (v3.5.4:3f56838976, Aug  7 2017, 12:56:33)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
========== RESTART: /Users/thiebaut/Desktop/Week1Friday.py
==========
SmithCollegeCollege
SmithCollegeSmithCollege
>>>
========== RESTART: /Users/thiebaut/Desktop/Week1Friday.py
==========
Smith College College
Smith CollegeSmith College
>>>
========== RESTART: /Users/thiebaut/Desktop/Week1Friday.py
==========
Smith CollegeSmith College
>>>
```

Ln: 14 Col: 24

Simultaneous Assignments

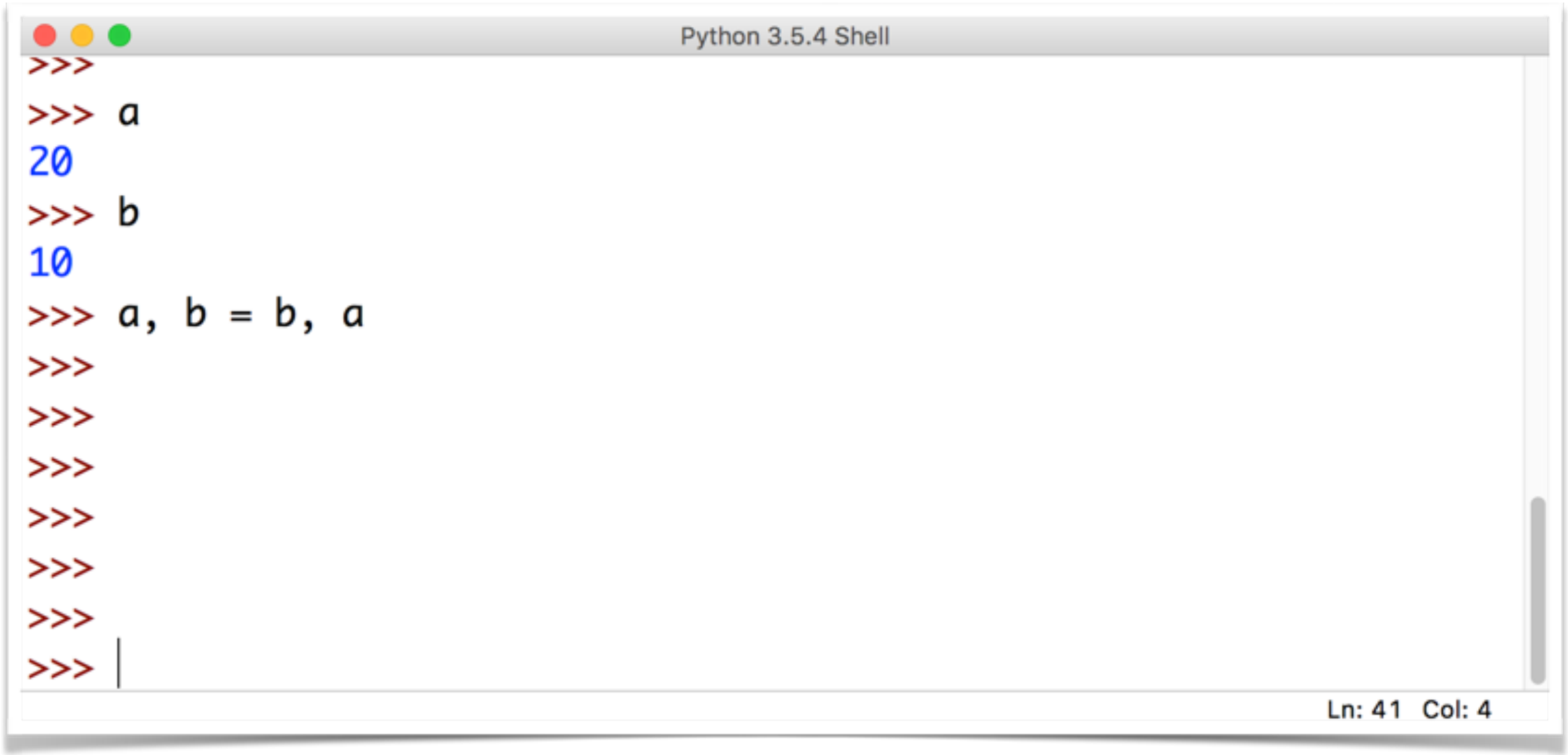
Simultaneous Assignments

Python 3.5.4 Shell

```
Python 3.5.4 (v3.5.4:3f56838976, Aug  7 2017, 12:56:33)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> a, b, c = 10, 20, 30
>>> a
10
>>> b
20
>>> c
30
>>>
```

Ln: 11 Col: 4

Swapping Variables



Lists and Variables

Python 3.5.4 Shell

```
>>>
>>>
>>> a
10
>>> b
20
>>> c
30
>>> a, b, c
(10, 20, 30)
>>> triplet = a, b, c
>>> x, y, z, = triplet
>>>
```

Ln: 51 Col: 4

```
a, b, c = 10, 20, 30
                                # a = 10, b = 20, c = 30
triplet = a, b, c               # triplet = (10, 20, 30)

x, y, z = triplet               # x = 10
                                # y = 20
                                # z = 30
```

- The Programming Process
- Variables
- **Definite Loops**
- Input

```
for <var> in <sequence>:
    <body>
```

```
for <var> in <sequence>:
    <body>
```

```
for can in [  ]:
    open( can )
    drink( can )
    throwAway( can )
```

```
for <var> in <sequence>:
    <body>
```

Sequence

```
for can in [  ]:
    open( can )
    drink( can )
    throwAway( can )
```

Many actions repeated, each group for each can