Administrivia
Complete homework before THIS Friday at 6:00 p.m.
A set of instructions executed by a computer to achieve a goal is called:

A  a process
B  a program
C  a procedure
D  an algorithm
A group of eight bits is called:
A  a nybble
B  a chomp
C  a byte
D  a gobble
Question #3

Python is:
A  a high-level language
B  a low-level language
Python is:
A  an interpreted language
B  a compiled language
Elements of Programming
What is a literal?

- Fixed value (noun)
- Represents data that doesn’t change (3 or 'firefly')
Executing a literal?

processor
Executing a literal?
Executing a literal?
What is an **operator**?

- Manipulates data (verb)
Executing an operator?
It needs a statement to make sense!
What is an expression?

- Combines literals and operators (phrase)
What is an expression?

- Combines literals and operators (phrase)
- Produce a new value
  - $3 \times 5$
  - $100 - 23$
Executing an expression?

3 + 5

processor
Executing an expression?

3 + 5

processor

8
What is an expression?

- Can be arbitrarily complicated
  - $3 + 8\times5 + 4 - 7/100$
1 + 1 * 2 = ?
A 4
B 3
C Something else
\[ 23 + 6/2 - 4 = \]

A 22
B 18
C -9
D Something else
Use parentheses!

23 + (6/2) − 4 is always clearer.
What are some other operators?

- exponentiation, **
What are some other operators?

- exponentiation, **
- modulus, % (important)
What are some other operators?

- exponentiation, **
- modulus, % (important)
- floor division, //
What are some other operators?

- bitwise OR, | 
- bitwise XOR, ^ 
- bitwise AND, & 
- bitwise left shift, << 
- bitwise right shift, >>
Example

\[ 1 \sim 2 \equiv \]

A 0
B 1
C 2
D 3
The machine state hasn’t changed.
So what?

- The machine state hasn’t changed.
- Programs are complex, and we need to remember results.
How do we keep values around?
How do we keep values around?

Elements of Programming
How do we reuse values?

- Low-level languages refer directly to memory address:
  - ADD DATA AT 10101101 11010100
  - TO DATA AT 11010100 01001001
  - STORE RESULT AT 00001101 01001110
What is a variable?

- The solution: name memory locations!
What is a variable?

- The solution: name memory locations!
- Variables name a memory location
What is a variable?

- The solution: name memory locations!
- Variables name a memory location
- Variables store a value
What is a variable?

- The solution: name memory locations!
- Variables name a memory location
- Variables store a value
- This value can change over time—it is a placeholder.
What new operator do we need?

- assignment, = (single equals sign)
How do we reuse values?

\[ x = 5 \]

memory

processor
How do we reuse values?

$x = 5$

processor

memory

$x = 5$
How do we reuse values?

\[ x + 1 \]

memory

\[ x = 5 \]

processor
How do we reuse values?

$x + 1$

processor

memory

$x = 5$
How do we reuse values?

```
x + 1
```

```
memory
x = 5
```

```
processor

6
```

Elements of Programming
What value is stored in the variable $x$?

$x = 17 + 7 \times 9$

A 3
B 31
C 55
D 78
Example

What value is stored in the variable $x$?

$x = 17 + 7*9$

$x = 3$

A 0
B 1
C 2
D 3
What is a statement?

- A statement changes the state of the computer (sentence)
What is a *statement*?

- A statement changes the state of the computer (sentence)
- Example: an assignment
What is a program?

Programs consist of series of statements:
What is a program?

- Programs consist of series of statements:
  - A script is a file containing a series of Python statement.
What is a program?

- Programs consist of series of statements:
  - A script is a file containing a series of Python statements.
  - A notebook (as we use in the lab) also collects series of Python statements.
What is a *program*?

- Programs consist of series of statements:
  - A script is a file containing a series of Python statement.
  - A notebook (as we use in the lab) also collects series of Python statements.
  - These are stored in text (there’s no magic, just text).
What is a program?

- Programs consist of series of statements:
  - A script is a file containing a series of Python statements.
  - A notebook (as we use in the lab) also collects series of Python statements.
  - These are stored in text (there’s no magic, just text).
- Each instruction is executed in order from top to bottom—together, these statements make up a program.
Our first program

x = 10
y = x ** 2
y = y + y
Reminders
Homework #1 due Friday, Sept. 30, 6:00 p.m.