Python Basics!

operators, expressions, computing

CS101 Lecture #2

2016-09-28

Warmup Quiz

Question #1

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

Question #2

A group of eight bits is called:

A a nybble
B a gobble
C a byte
D a bite

Question #3

Python is:

A a high-level language

B a low-level language

Question #4

Python is:

A an interpreted language

B a compiled language

Question #4

Python is:

A *an interpreted language*

B a compiled language

Interpreted vs Compiled

A *compiled* language has a compiler to compile its code into machine instructions, e.g., C, C++

A *interpreted* language executes its program directly – one line at a time, e.g., python, matlab

Elements of Programming

Literal

What is a literal?

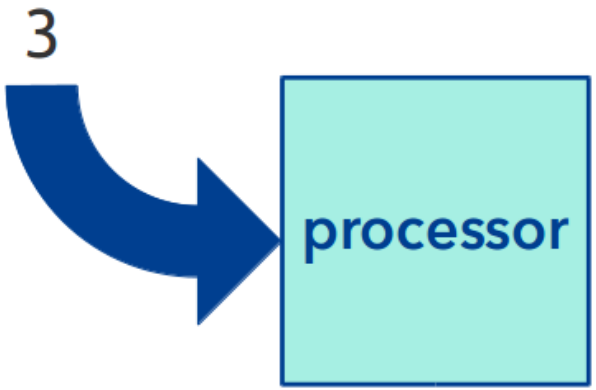
Literal

What is a literal?

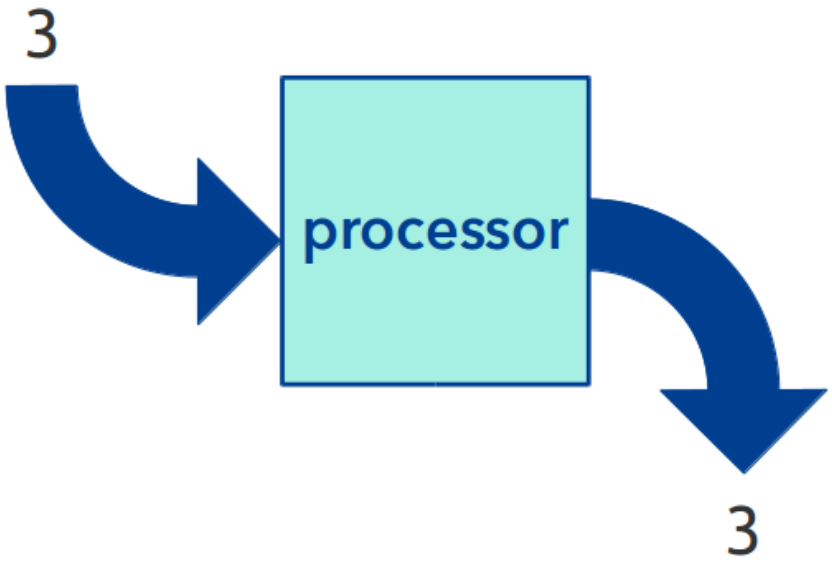
A fixed value, e.g. 5, 'apple'

Executing a literal?

Executing a literal?



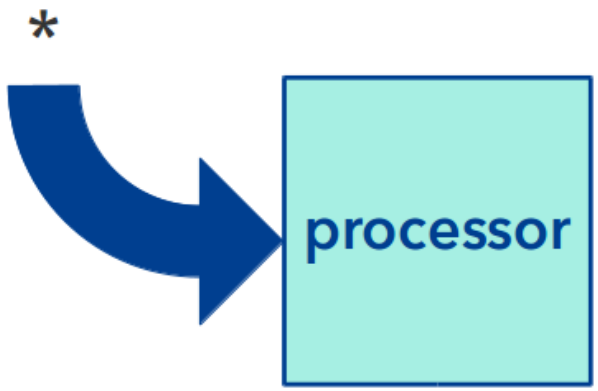
Executing a literal?



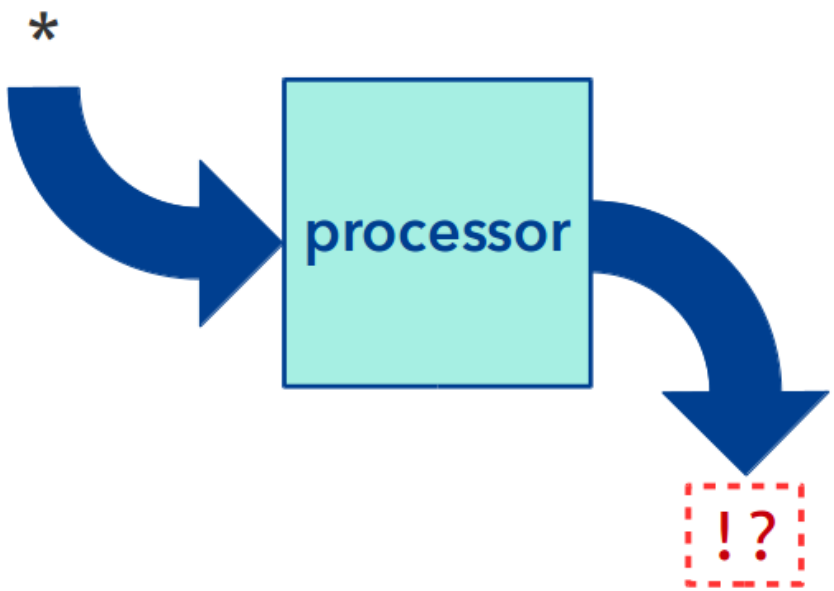
What is an ***operator****?*

- Manipulates data (verb)
- +,−,∗,/

Executing an operator?



It needs operands to make sense!



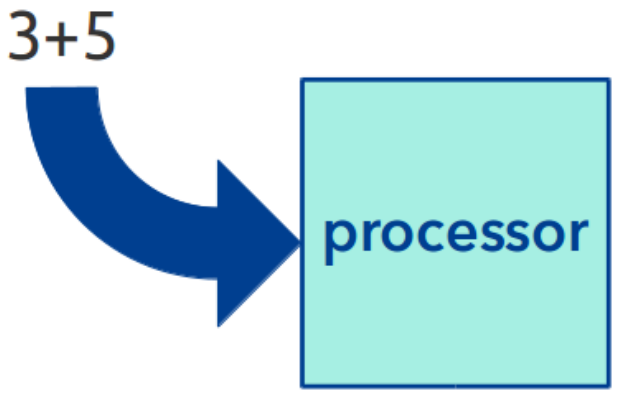
*What is an **expression**?*

- The combination of operands and operators

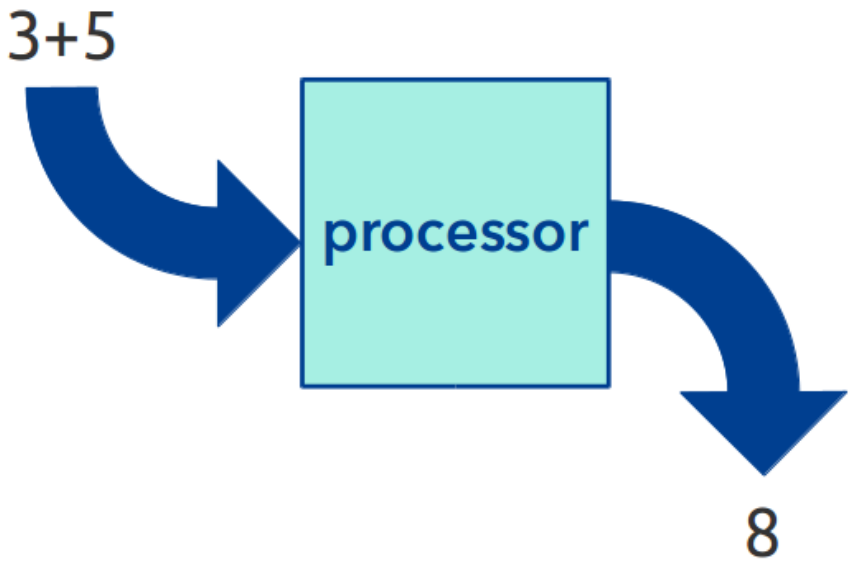
*What is an **expression**?*

- The combination of operands and operators
- Produce a new value (another literal)
  - `3 * 5`
  - `100 - 23`

Executing an expression?



Executing an expression?



*What is an **expression**?*

- Can be arbitrarily complicated
  - `3 + 8*5 + 4 - 7/100`

Order of Operator

(also called: Operator Precedence)

$1 + 1 * 2 \stackrel{?}{=}$

A 4

B 3

C Something else

Change order of operator by parentheses

$(1 + 1) * 2 \stackrel{?}{=}$

A 4
B 3
C Something else

Question

$23 + 6/2 - 4 \stackrel{?}{=}$

A 22
B -1
C 35
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)
  - $5\%2 = 1$

What are some other operators?

- exponentiation, **
- modulus, % (important)
  - $5\%2 = 1$
- floor division, //

What are some other operators?

- exponentiation, **
- modulus, % (important)
  - $5\%2 = 1$
- floor division, //
  - $5//2 = 2$

What are some other operators?

- exponentiation, **
- modulus, % (important)
  - $5\%2 = 1$
- floor division, //
  - $5//2 = 2$
  - (-5)//2 = ?

What are some other operators?

- bitwise `OR`, `|`
- bitwise `AND`, `&`
- bitwise `XOR`, `^`

OR	0	1
0	0	1
1	1	1

AND	0	1
0	0	0
1	0	1

XOR	0	1
0	0	1
1	1	0

Example

$1 \hat{} 2 \stackrel{?}{=}$

A 0
B 1
C 2
D 3

What are some other operators?

- bitwise left shift, <<
- bitwise right shift, >>

What are some other operators?

- bitwise left shift, <<
- bitwise right shift, >>

$$00001111 << 1 = 00011110$$

What are some other operators?

- bitwise left shift, <<
- bitwise right shift, >>

$$00001111 << 1 = 00011110$$
$$00011110 >> 2 = 00000111$$

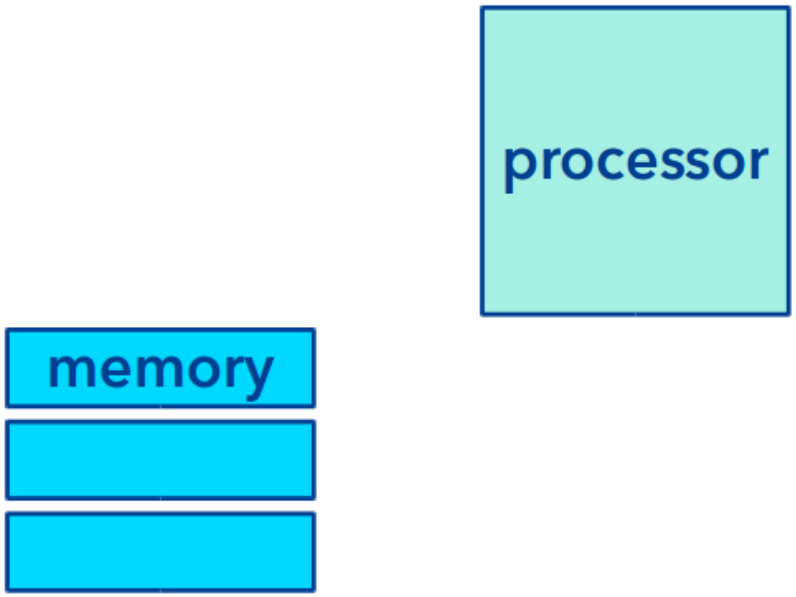
So what?

- The machine state hasn't changed.

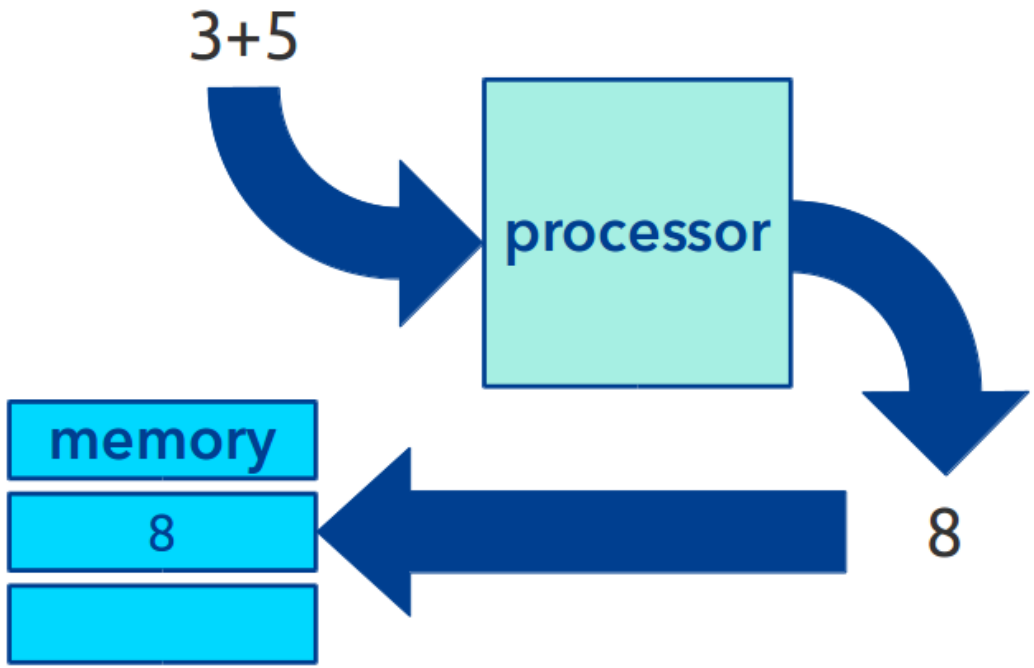
So what?

- 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?



*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

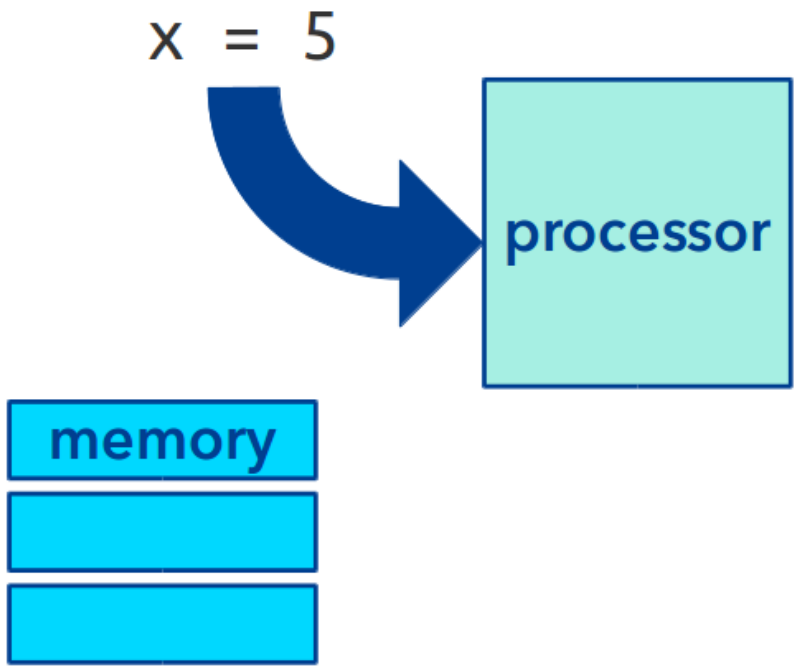
*What is a **variable**?*

- The solution: name memory locations!
- Variables name a memory location
- Variables store a value
- This value can change over time—it is a placeholder.

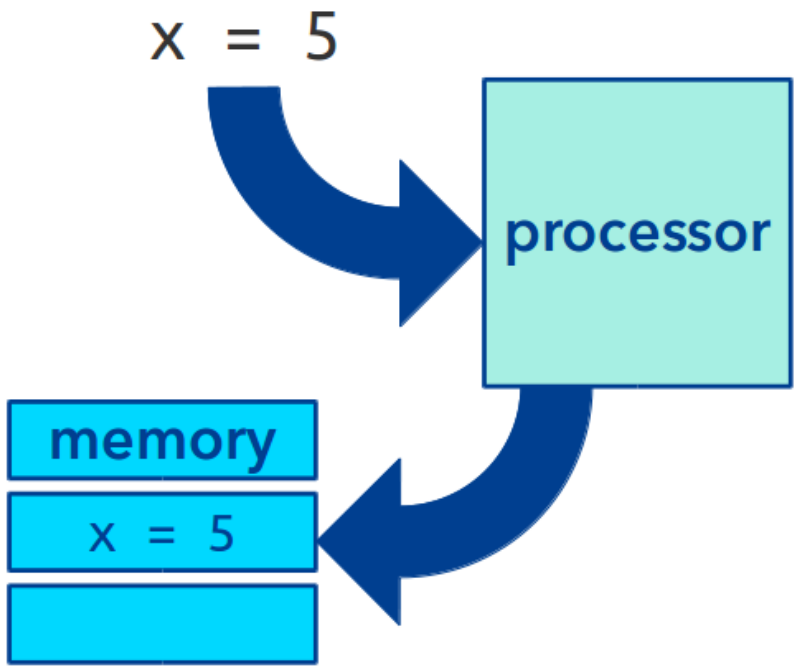
What new operator do we need?

- assignment, = (single equals sign)

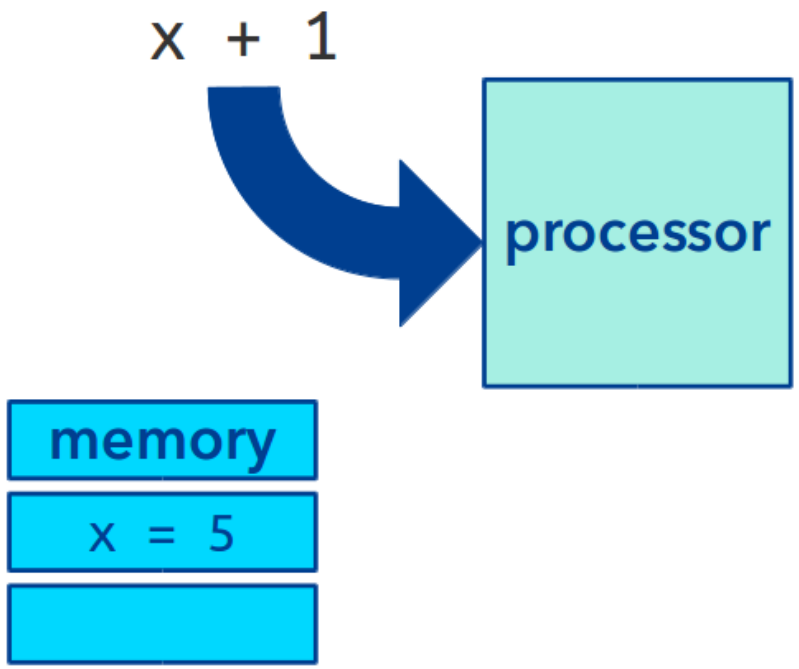
How do we reuse values?



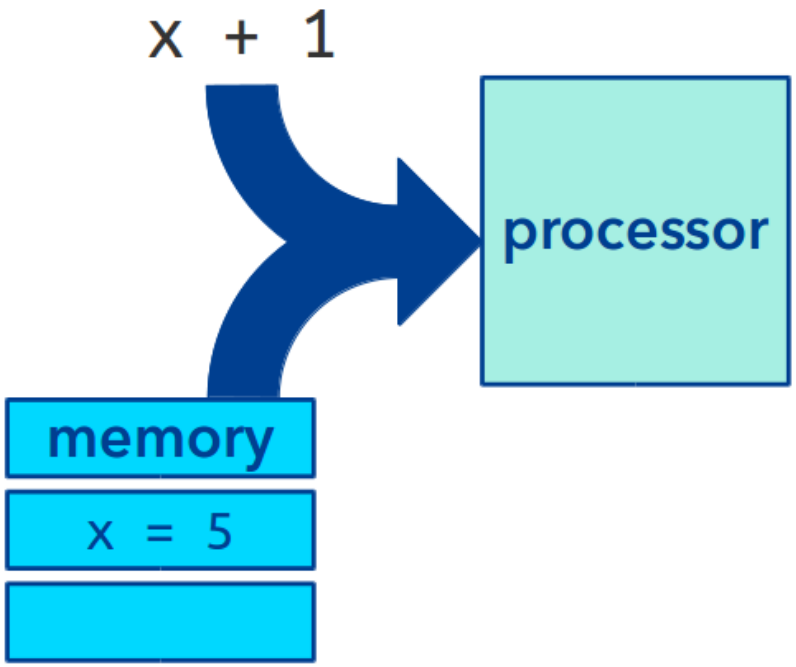
How do we reuse values?



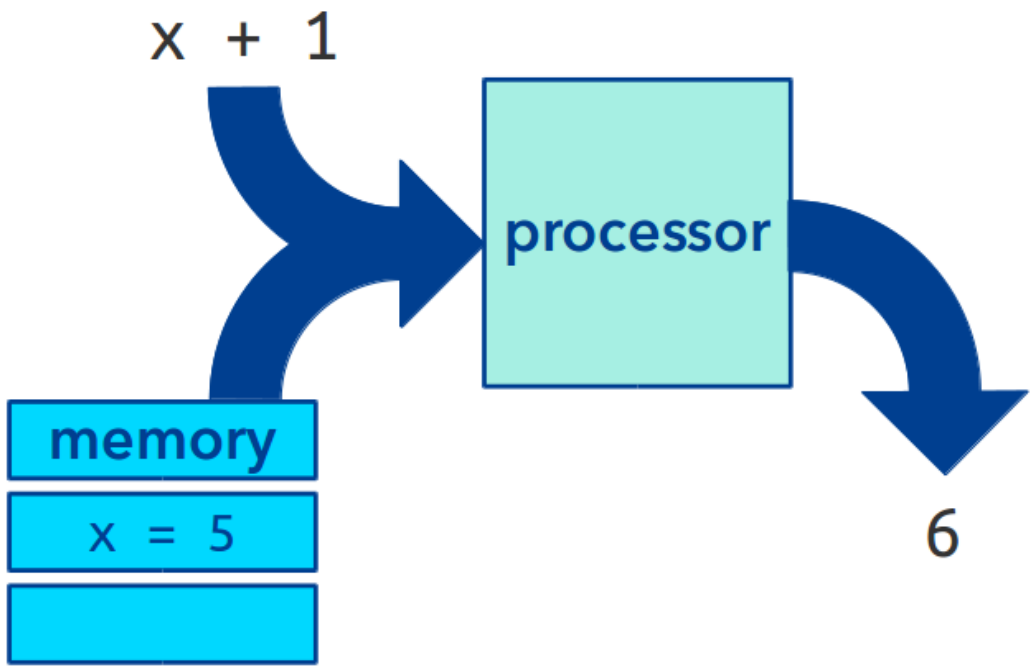
How do we reuse values?



How do we reuse values?



How do we reuse values?



Example

What value is stored in the variable `x`?

```
x = 17 + 7*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 1
B 3
C 6
D Something Else

Example

What value is stored in the variable `x`?

```
x = 17 + 7*9
x = 3
x + 3
```

A 1
B 3
C 6
D Something Else

Example

What value is stored in the variable `x`?

```
x = 17 + 7*9
x = 3
x + 3
```

A 1
B 3
C 6
D Something Else

`x + 3` is a literal
The execution of a literal does not change machine state

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 series of python statements is also called a python *script*.

*What is a **program**?*

- Programs consist of series of statements:
  - A series of python statements is also called a python *script*.
  - A python script is stored in text that you can edit (there's no magic, just text).

*What is a **program**?*

- Programs consist of series of statements:
  - A series of python statements is also called a python *script*.
  - A python script is stored in text that you can edit (there's no magic, just text).
- Each statement is executed in order from top to bottom by the python interpreter – together, these statements form a program.

Our first program

```
x = 10
y = x ** 2
y = y + y
```

Our first program

```
x = 10
y = x ** 2
y = y + y
```

Save the code to file: my_first_script.py
Run from your command line window (Linux or iOS, Windows also OK):

» python my_first_script.py

Reminders

Reminders

Homework #1 due Monday, Sep. 25, 6:00 p.m.