

Introduction to Python I

Fall 2013

Carola Wenk

Python Programming

```
def f(a, b):  
    print "this is function f"  
    return a+b;
```

function declaration

```
evens = 0; odds = []  
print f(1, 2)  
print f('z', f('a', 'b'))  
for i in range(1,10):  
    if (i % 2 == 0):  
        evens += i  
    else:  
        odds.append(i)  
print evens; print sum(odds)
```

sequence of statements

A Python script is a sequence of function declarations followed by a sequence of statements.

A function is just a way to reuse useful blocks of statements.

Python Programming

```
def f(a, b):  
    print "this is function f"  
    return a+b;
```

```
evens = 0; odds = []  
print f(1, 2)  
print f('z', f('a', 'b'))  
for i in range(1,10):  
    if (i % 2 == 0):  
        evens += i  
    else:  
        odds.append(i)  
print evens; print sum(odds)
```

function declaration

variable assignment

system library calls

loop and conditional
statements

Variables can take on integer, string, floating point values, and can also be lists of these.

Python provides a large set of libraries with useful functions.

Python Syntax

There are four primary categories:

- Conditional and Logical Expressions

`if, elif, else, <, >, ==, and, or, not`

- Operators and Assignment

`+, -, *, /, =, ...`

- Looping Constructs

`for, while, range()`

- Functions

```
def <name>(<arguments>) :  
    <statement block>
```

We will learn syntax as we go, so we won't go into exhaustive detail.

Nested Statements

- Python syntax is designed to be uncluttered; unlike most languages, blocks of statements are defined only by how they are indented.

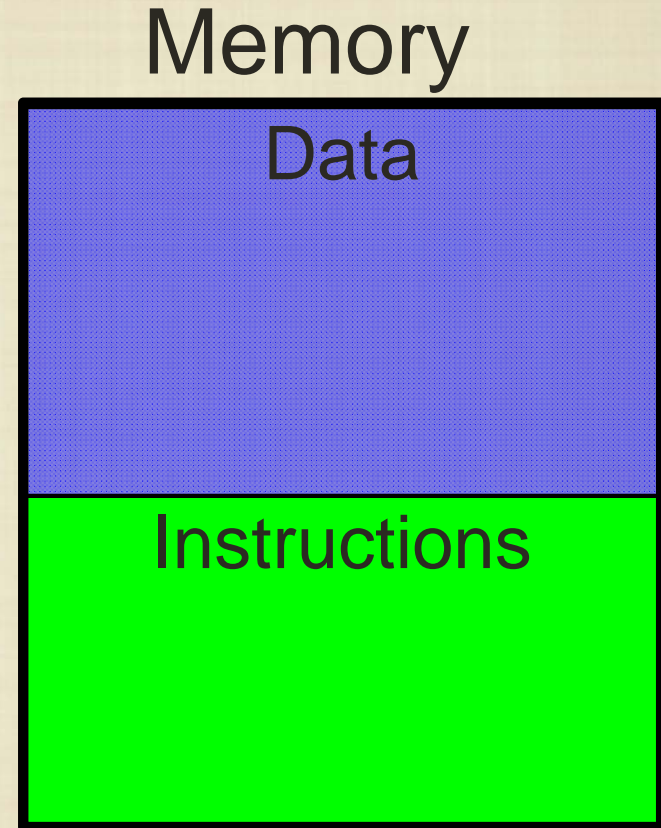
```
for i in range(1,10):  
    if (i % 2 == 0):  
        evens += i  
    else:  
        odds.append(i)
```

The first statement in a nested block determines how many spaces the entire block should be indented.

Variables and Python “Types”

Unlike most languages, we do not need to “declare” variables up front.

```
x = 25  
y = [1, 2, 3]  
z = "cmps1500"  
x = "hello"
```

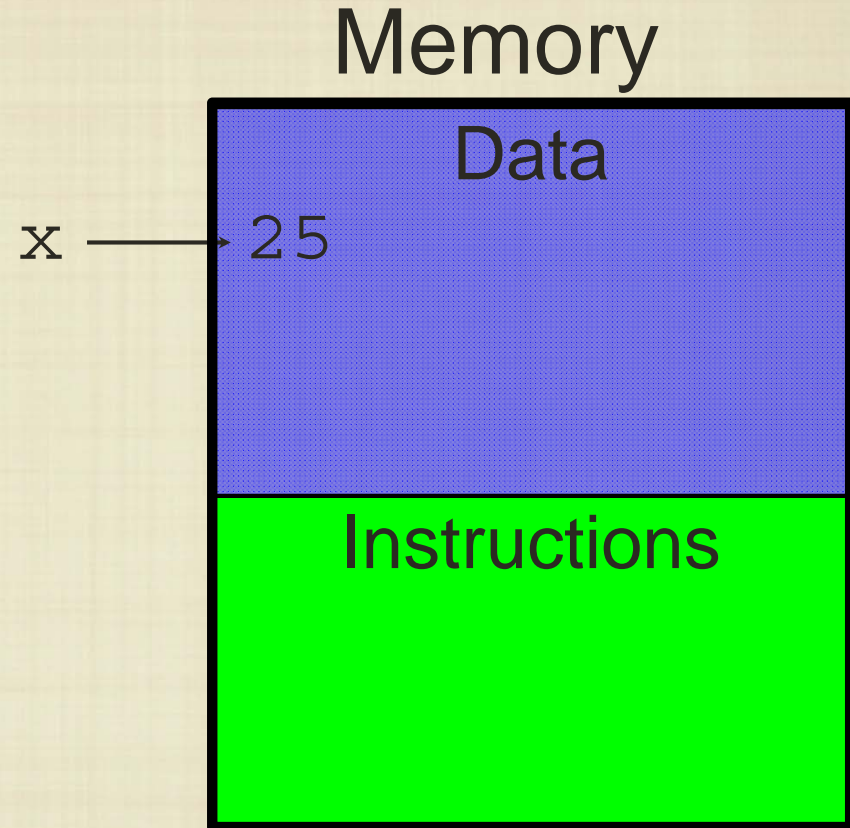


Python allows types of variables to be unspecified and allocates storage as necessary.

Variables and Python “Types”

Unlike most languages, we do not need to “declare” variables up front.

```
x = 25  
y = [1, 2, 3]  
z = "cmps1500"  
x = "hello"
```



Python allows types of variables to be unspecified and allocates storage as necessary.

Variables and Python “Types”

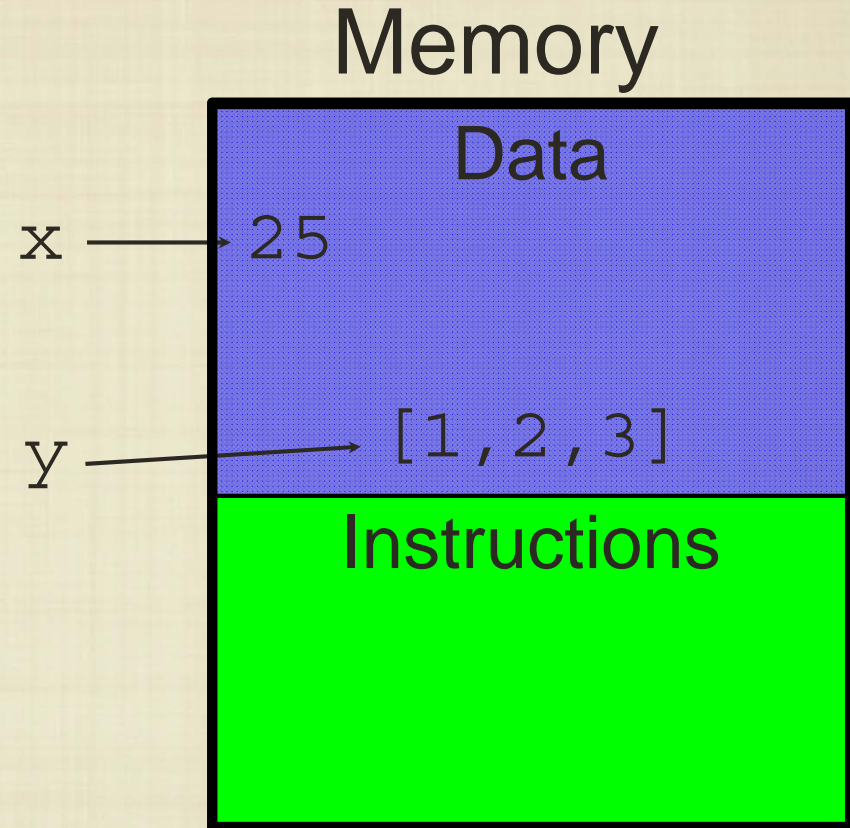
Unlike most languages, we do not need to “declare” variables up front.

```
x = 25
```

```
y = [1, 2, 3]
```

```
z = "cmps1500"
```

```
x = "hello"
```

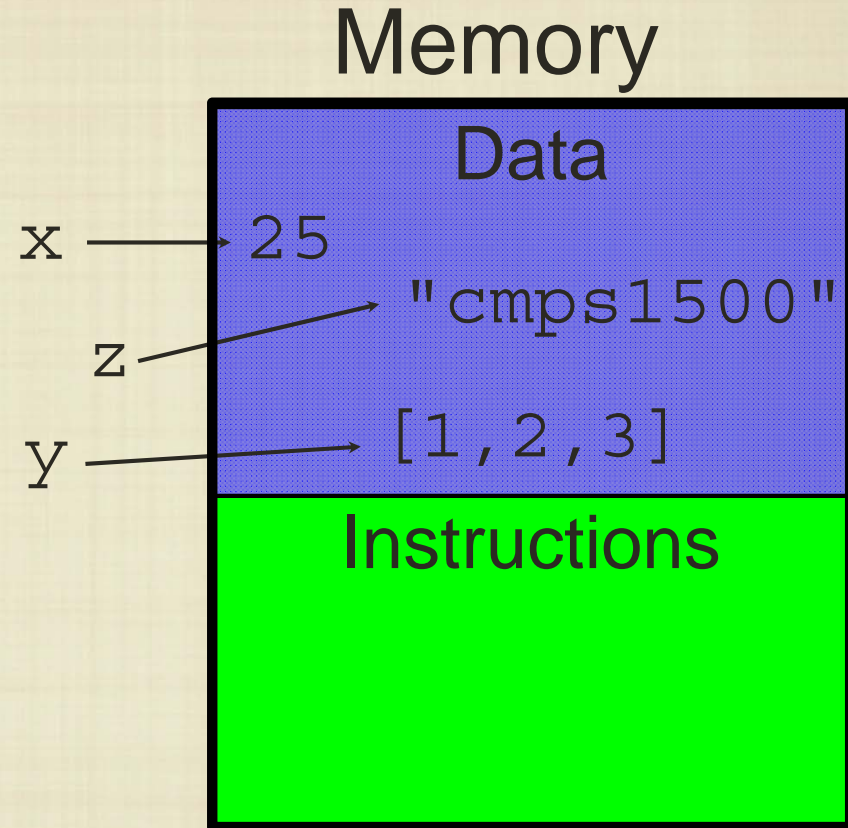


Python allows types of variables to be unspecified and allocates storage as necessary.

Variables and Python “Types”

Unlike most languages, we do not need to “declare” variables up front.

```
x = 25  
y = [1, 2, 3]  
z = "cmps1500"  
x = "hello"
```

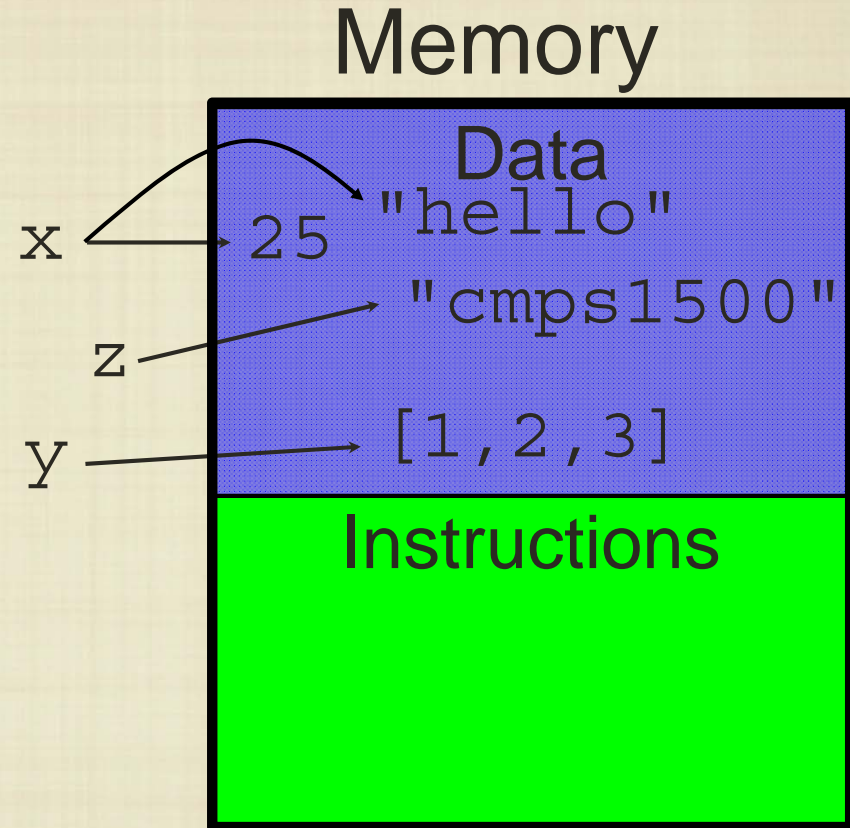


Python allows types of variables to be unspecified and allocates storage as necessary.

Variables and Python “Types”

Unlike most languages, we do not need to “declare” variables up front.

```
x = 25  
y = [1, 2, 3]  
z = "cmps1500"  
x = "hello"
```



Python allows types of variables to be unspecified and allocates storage as necessary.