CS101: Intro to Computing Fall 2015

Lecture 16

Feedback

Administrivia

- Homework 11 due on Monday
- Homework 12 will be released on Monday

REVIEW

TypeError: unsupported operand type(s) for +: 'str' and 'int'

What code produces this error?

a)
$$1 + (1,2)$$

b) "3" +
$$len("3")$$

c)
$$5 + ''.join([0,3])$$

TypeError: can only concatenate tuple (not "int") to tuple

```
What code produces this error?
a) for i in zip("ABC"):
    print i[0]
b) for i in range("ABC"):
     print i+1
c) for i in enumerate("ABC"):
    print i+1
```

MODELING AND SIMULATION

Modeling

- Model a program to simulate a real-world process or system over time
- Simulation a run of a computer model
- First "build a model" then "run a simulation"

Modeling design

- 1. Determine equations governing the system and entities involved.
- 2. Determine state variables, units, initial conditions, and constants.
- 3. Determine how state variables are updated through time (transition.)
- 4. Implement model and begin simulation.
- 5. Validate and refine model.

```
v = 0.0
y=1.0
g = -9.8
t=0
dt=.01
while(y>0.0):
   t+=dt
   y += v * dt
   v += g*dt
```

Which of these is *not* state variable?

- a) t
- b) v
- c) y
- d) g

```
v = 0.0
y = 1.0
g = -9.8
t=0
dt=???
while(y>0.0):
   t+=dt
   y+=v*dt
   v += g * dt
```

What will produce the most *accurate* simulation?

- a) 1
- b) .1
- c) .01
- d) .001

```
v = 0.0
y=1.0
g = -9.8
t=0
dt=???
while(y>0.0):
   t+=dt
   y += v * dt
   v += g*dt
```

What will produce the *fastest* simulation?

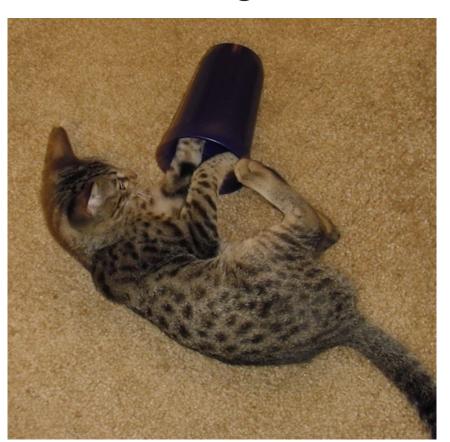
- a) 1
- b) .1
- c) .01
- d) .001

State variables

- A collection of variables describing the current state of the system.
- Describe all information we are interested in.
- Describe all information needed to determine the future state.

Example

- A kitten knocks a cup off of a 1-meter high table. How long until it hits the ground?
- $g=-9.8 \text{m/s}^2$
- $v_0 = 0 \text{m/s}, y_0 = 1 \text{m}$
- $v_{t+1} = v_t + g^* \Delta t$
- $y_{t+1} = y_t + v_t \Delta t$
- $\Delta t = ?$



Time step size

- How much time passes between iterations of our simulation.
- What is ∆t?
- Determines the resolution/fidelity of our model.
- Smaller step size means more accurate reproduction of reality.
- Smaller step size means longer run time.

NUMPY

Numpy

- Module for Python to extend its numerical capabilities
- Designed for more efficient computation
- Designed for manipulating arrays and matrices

```
import numpy as np
```

Arrays

- Numpy arrays are similar to lists:
 - Represent a collection of items
 - Can be indexed
- Numpy arrays are different than lists:
 - Fixed size
 - All elements have the same type
 - Can do operations on all elements

Arrays

```
import numpy as np
x=np.array([4,3,2,1])
x[1]
x[1]=-3
x[1:3]
x+=1
x = x * * 2 + 3
x.size
```

```
x=np.array([1,2])+1
x=x*2
```

```
What is the final value of x?

a) array([1,2,3])

b) array([1,2,1,1,2,1])

c) array([2,4,2]

d) array([4,6])
```

Data type

- Many possible types in numpy
 - Boolean
 - integers (8, 16, 32, 64 bits)
 - floats (16, 32, and 64 bits)
 - complex (64 and 128 bits)

```
a=[3,2,4]
x=np.array(a,dtype=np.float64)
x.dtype
```

arange

- Returns array over a range (like list range)
 - Argument 1: Start
 - Argument 2: End
 - Argument 3: Step size

```
x=np.arange(10,25,5.0)
len(x)
```

```
a=np.arange(0.0,1.0,.25)
b=a+a
a=a+b[2]
x=a[-1]
```

What is the final value of x?

- a) .75
- b) 1.5
- c) 0.0
- d) 1.75

linspace

- Returns array of evenly spaced values
 - Argument 1: start of range
 - Argument 2: end of range
 - Arguemnt 3: number of points in range

```
x=np.linspace(0,1,100)
y=x**2
plt.plot(x,y,'g--')
```

zeros

- Returns array of zeros
 - Argument 1: the number of zeros

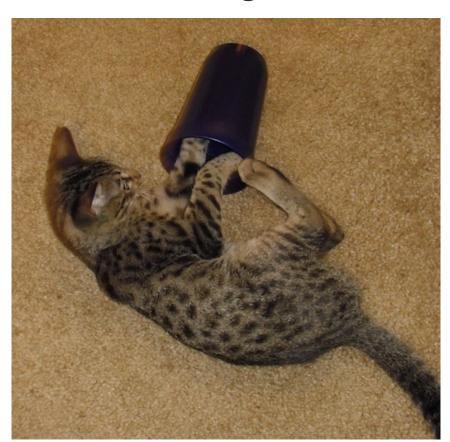
```
x=np.zeros(100)
```

x.dtype

x.size

Example

- A kitten knocks a cup off of a 1-meter high table. How long until it hits the ground?
- $g=-9.8 \text{m/s}^2$
- $v_0 = 0 \text{m/s}, y_0 = 1 \text{m}$
- $v_{t+1} = v_t + g^* \Delta t$
- $y_{t+1} = y_t + v_t \Delta t$
- $\Delta t = ?$



Why use numpy?

Extremely powerful!

```
x=np.linspace(0,2*np.pi,100)
y=np.sin(x)
plt.plot(x,y,'g--')
```