Preliminaries: Additional List Operations

- List membership: `in`
  - returns Boolean
  - `e in L`

- List of consecutive integers: `range`
  - returns a list
  - `range(10) → [0, 1, …, 9]`
  - `range(100, 200) → [100, 101, …, 199]`

- Assignment: `=`
  - creates an alias

Preliminaries: Random

- `random` provides a number of helpful methods
  - `random()` returns a float in [0, 1]
  - `randint(x, y)` returns an int in [x, y]
  - `shuffle(L)` permutes L in place

- Multiple ways to import
  - `import random`
    - `random.random, random.randint, random.shuffle`
  - `import random as rd`
    - `rd.random, rd.randint, rd.shuffle`
  - `from random import random, randint, shuffle`
    - `random, randint, shuffle`

Iteration

- while loops
  - similar to other languages with minor syntactic differences

- for loops
  - primarily list based

```python
i = 0
while i < 10:
    instructions
    i += 1

for i in range(x):
    instructions
```
Iteration

• for loops with range and len
  • `range(len(L))` returns list with entry for each element in a list L

• iterator
  • operate on each element of list
  • `e` takes value of each element in list in turn

```python
for i in range(len(L)):
    instructions

for e in L:
    instructions
```

Important note about iterators

• Consider code this code fragment

```python
for e in L:
    e = 2 * e
```

• `e` is really a reference to each element in list
• `e = 2 * e` reassigns the reference but doesn’t affect the value stored in the list
• Thus L is unchanged