CS 161: Introduction to Programming and Problem-solving

Warren Harrison

Lists & Tuples

Part B

PORTLAND STATE UNIVERSITY

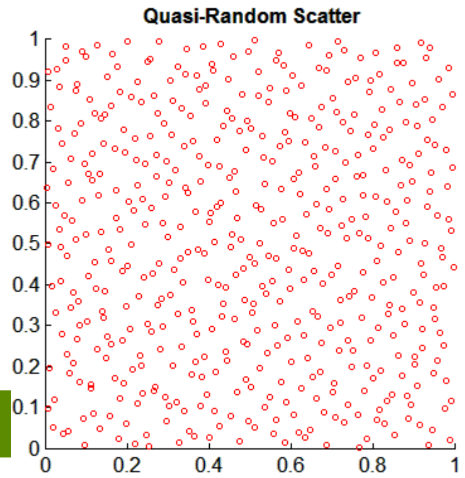
Rolling Dice

- Which side comes up when you roll dice is random – each side should be equally likely
- We can use Python's random module to tell us which side – one, two, three, four, five or six comes up after a roll:

```
import random
roll = random.randint(1,6)
print("You rolled a",roll)
```

PORTLAND STATE UNIVERSITY

The random Module



- Modules are files that group together a collection of *functions* related to a specific area.
- Modules must be *imported* in order to use their functions.
- The random module contains functions related to generating random numbers.
- Functions from a module are called by using the module's name, a dot, and the function name.

PORTLAND STATE UNIVERSITY

randint

- Call as:

```
import random
random.randint(low,high)
```

- Returns a "random" integer between *low* and *high*
- You must import random before you can use it – *to save space, I'll omit it in the examples, but it must be there*

High-Low: a Dice Game

- In the game of high-low, a pair of dice are rolled. The outcome is
  - High if the sum is 8, 9, 10, 11, or 12.
  - Low if the sum is 2, 3, 4, 5, or 6
  - Seven if the sum is 7
- A player can bet on any of the three outcomes. The payoff for a bet of high or for a bet of low is 1:1. The payoff for a bet of seven is 4:1.

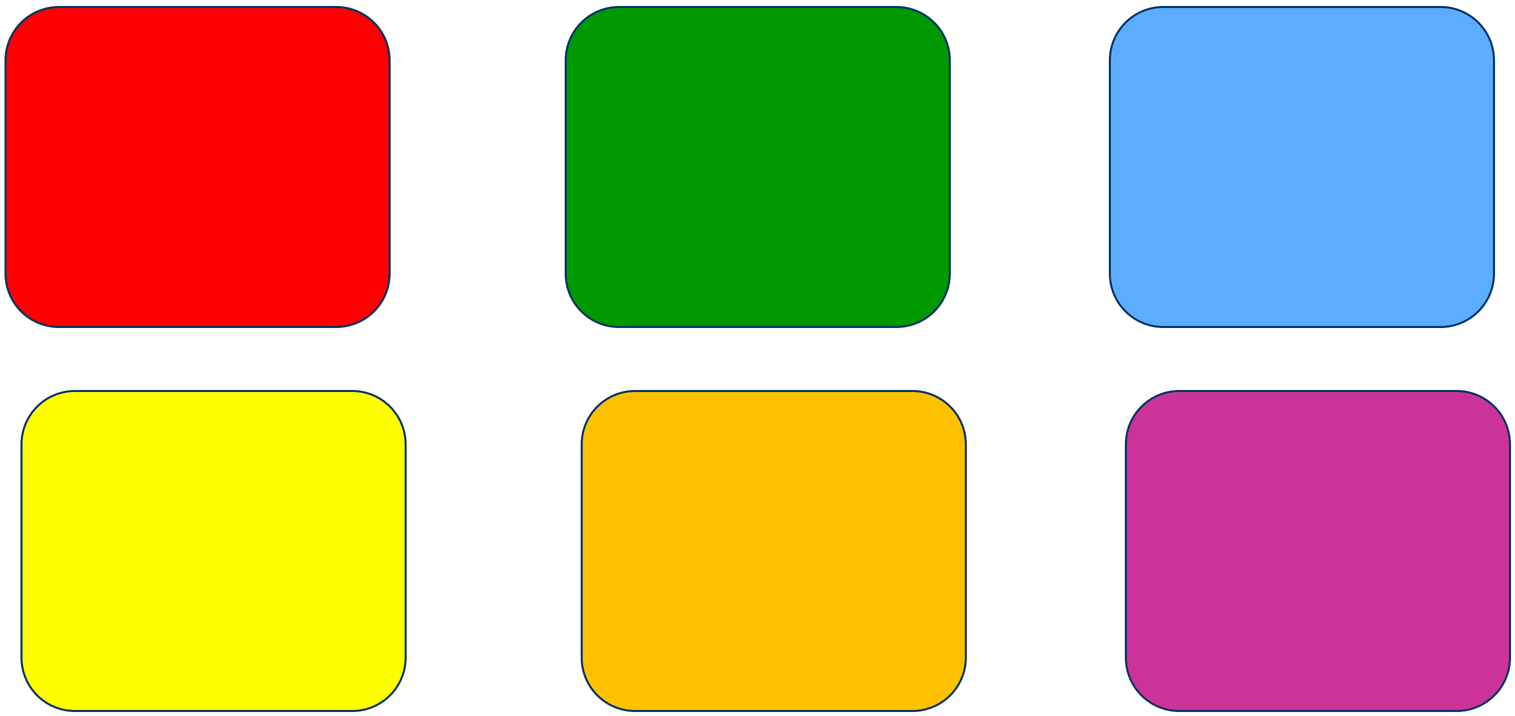
High-Low

```
outcome = "ERROR"
roll_1 = random.randint(1,6)
roll_2 = random.randint(1,6)
sum = roll_1 + roll_2
if((sum >= 8) and (sum <= 12)):
    outcome = "High"
elif((sum >= 2) and (sum <= 6)):
    outcome = "Low"
elif(sum == 7):
    outcome = "Seven"
print("You rolled a",roll_1,"and a",roll_2)
print("which is",outcome)
```

What If Our Dice Has Colors Rather than Numbers?

- Red
- Yellow
- Blue
- Green
- Orange
- Violet

Call Sides by Color Rather than Pips



Representing Colored Dice

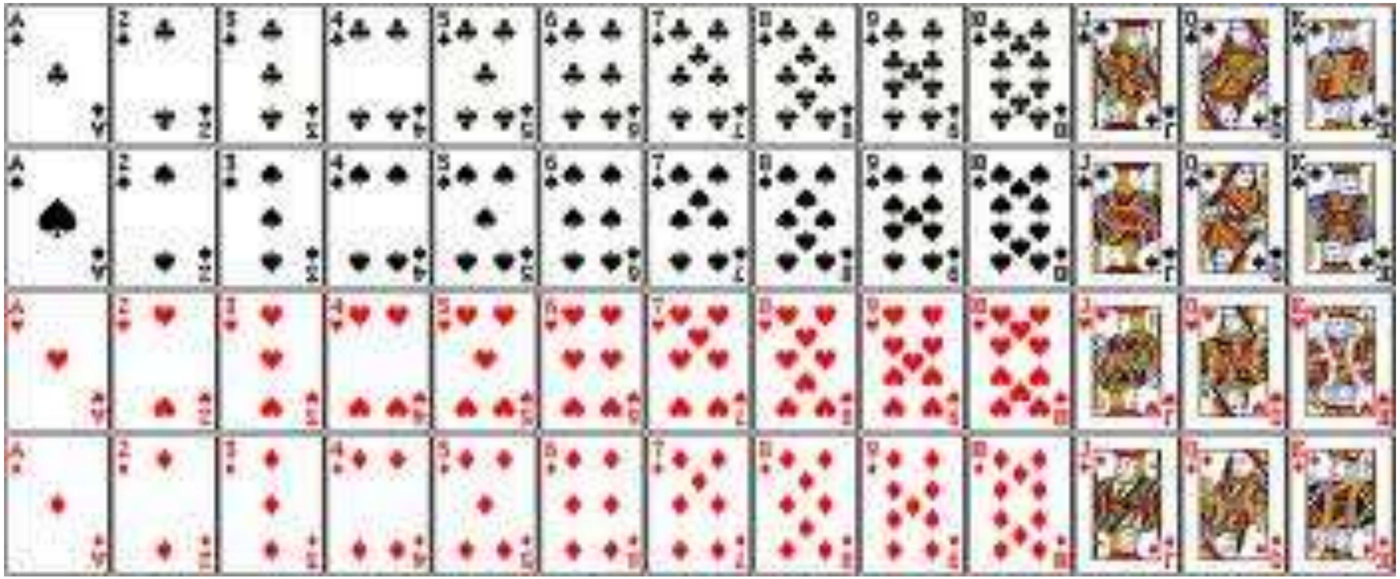
- Use a list to represent a six-sided die
- We can associate numbers with each side of the die, and use those numbers to index into the list

```
dice=["red","green","blue","yellow","orange","violet"]
roll_1 = random.randint(0,5)
roll_2 = random.randint(0,5)
print("You rolled a")
print(dice[roll_1],"and a",dice[roll_2])
```

PORTLAND STATE UNIVERSITY

How About Dealing Cards?

- 4 suites of 13 cards each – a total of 52 cards to represent



PORTLAND STATE UNIVERSITY

A Deck of Cards

```
deck = [“Ace of Clubs","2 of Clubs","3 of
   ...
```

… do I really have to do this 52 times?

Of course not …

… there is a pattern: "*n* of *suite*" where *n* is Ace to King

A Program to Populate a List Representing a Card Deck

```
deck = []
for suite in
  ("Clubs","Spades","Hearts","Diamonds"):
    for number in ("Ace","2","3","4","5",
      "6","7","8","9","10","Jack","Queen",
      "King"):
        card = number + " of " + suite
        deck = deck + [card]
print(deck)
```

A Card Deck …

```
>>>
['Ace of Clubs', '2 of Clubs', '3 of Clubs', '4 of Clubs', '5 of
   Clubs', '6 of Clubs', '7 of Clubs', '8 of Clubs', '9 of
   Clubs', '10 of Clubs', 'Jack of Clubs', 'Queen of Clubs',
   'King of Clubs', 'Ace of Spades', '2 of Spades', '3 of
   Spades', '4 of Spades', '5 of Spades', '6 of Spades', '7 of
   Spades', '8 of Spades', '9 of Spades', '10 of Spades', 'Jack
   of Spades', 'Queen of Spades', 'King of Spades', 'Ace of
   Hearts', '2 of Hearts', '3 of Hearts', '4 of Hearts', '5 of
   Hearts', '6 of Hearts', '7 of Hearts', '8 of Hearts', '9 of
   Hearts', '10 of Hearts', 'Jack of Hearts', 'Queen of Hearts',
   'King of Hearts', 'Ace of Diamonds', '2 of Diamonds', '3 of
   Diamonds', '4 of Diamonds', '5 of Diamonds', '6 of Diamonds',
   '7 of Diamonds', '8 of Diamonds', '9 of Diamonds', '10 of
   Diamonds', 'Jack of Diamonds', 'Queen of Diamonds', 'King of
   Diamonds']
>>>
```

Nested Loops

- One loop inside another loop
- Every time the outside loop goes through one iteration, the inside loop iterates through its entire range

```
for i in (1,2,3,4):
   for j in (1,2,3):
      print("do something")
```

Running a Nested Loop

3 iterations inside 4 iterations

```
>>>
i is 1 and j is 1
i is 1 and j is 2
i is 1 and j is 3
i is 2 and j is 1
i is 2 and j is 2
i is 2 and j is 3
i is 3 and j is 1
i is 3 and j is 2
i is 3 and j is 3
i is 4 and j is 1
i is 4 and j is 2
i is 4 and j is 3
>>>
```

```
for i in (1,2,3,4):
    for j in (1,2,3):
        print("i is",i,"and j is",j)
```

PORTLAND STATE UNIVERSITY

Dealing Five Cards to Two Players

- Use a similar approach to deal cards as we used with rolling dice
- Generate a random index, and use that to select a "card"
- But once we deal a card, we can't deal it a second time … when a card is dealt, it must be removed from the list

Dealing Five Cards to You and I From the List

```
for hand in (1,2,3,4,5):
    for player in ("You","I"):
        draw = random.randint(0,len(deck))
        card = deck[draw]
        print(player,"got the",card)
        del deck[draw]
print(deck)
```

Here Are Our Hands

```
You got the 6 of Hearts
I got the 10 of Diamonds
You got the Jack of Spades
I got the King of Clubs
You got the Ace of Hearts
I got the King of Diamonds
You got the 8 of Spades
I got the 10 of Spades
You got the 9 of Clubs
I got the 2 of Clubs
['Ace of Clubs', '3 of Clubs', '4 of Clubs', '5 of Clubs', '6 of Clubs', '7 of
    Clubs', '8 of Clubs', '10 of Clubs', 'Jack of Clubs', 'Queen of Clubs', 'Ace of
    Spades', '2 of Spades', '3 of Spades', '4 of Spades', '5 of Spades', '6 of
    Spades', '7 of Spades', '9 of Spades', 'Queen of Spades', 'King of Spades', '2 of
    Hearts', '3 of Hearts', '4 of Hearts', '5 of Hearts', '7 of Hearts', '8 of
    Hearts', '9 of Hearts', '10 of Hearts', 'Jack of Hearts', 'Queen of Hearts', 'King
    of Hearts', 'Ace of Diamonds', '2 of Diamonds', '3 of Diamonds', '4 of Diamonds',
    '5 of Diamonds', '6 of Diamonds', '7 of Diamonds', '8 of Diamonds', '9 of
    Diamonds', 'Jack of Diamonds', 'Queen of Diamonds']
>>>
```

You Probably Want Each Player's Hand to Be Represented as a List

```
for hand in (1,2,3,4,5):
    draw = random.randint(0,len(deck))
    yourHand = yourHand + [deck[draw]]
    del deck[draw]
    draw = random.randint(0,len(deck))
    myHand = myHand + [deck[draw]]
    del deck[draw]

print("Your Hand",yourHand)
print("My Hand",myHand)
```

Manipulating List Items:
A Summary

- The items in a list can be replaced by simply assigning the indexed location a new value

```
deck[2] = "Joker"
```

- You can delete a list item using the index

```
del deck[2]
```

- Or the value

```
deck.remove("Joker")
```