# Lab 3
# Basic Python Programming: Selection and Control

# *for* and Iteration

- One of Python's strength is its rich set of built in data structures.
- The for statement iterates through each element of a collection (list, etc.)

```
for element in collection:
    suite
for num in range(a,b):
    suite
```

# *try-except* Block

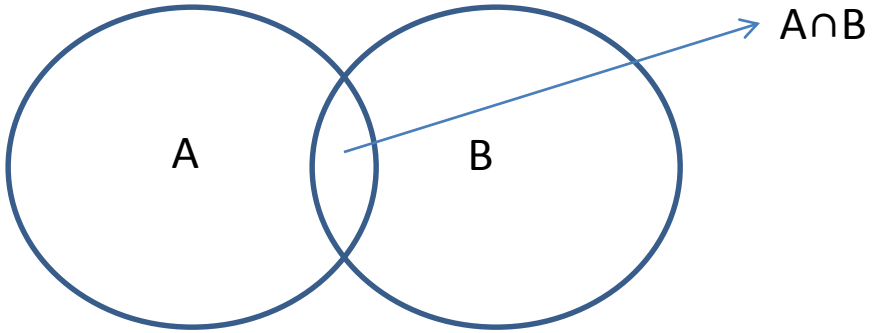
- The try statement works as follows.

```
while True:
    try:
        x = int(raw_input("Please enter a number: "))
        break
    except ValueError:
        print "Oops! That was no valid number. Try again..."
```

- First, the *try clause* (the statement(s) between the try and except keywords) is executed.
- If no exception occurs, the *except clause* is skipped and execution of the try statement is finished.
- If an exception occurs during execution of the try clause, the rest of the clause is skipped. Then if its type matches the exception named after the except keyword, the except clause is executed, and then execution continues after the try statement.

# Dice Game Problem

- Assuming we are playing a game with throwing two dice. What is the probability of getting ***at least*** one 6?
- Of course, this can be solved analytically:




$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$= \frac{1}{6} + \frac{1}{6} - \frac{1}{36}$$

$$= \frac{11}{36}$$

# Monte Carlo Simulation

- Create a script that in a loop from 1 to n draws two uniform random numbers between 1 and 6 and counts how many times ***p*** a 6 shows up.
- Write out the estimated probability ***p/float(n)*** together with the exact result 11/36. Run the script a few times with different ***n*** values (preferably read from the command line) and determine from the experiments how large ***n*** must be to get the first three decimals (0.306) of the probability correct.

# A More Complicate Dice Game Problem

- Somebody suggests the following game. You pay 1 unit of money and are allowed to throw four dice. If the sum of the eyes on the dice is less than 9, you win 10 units of money, otherwise you lose your investment. Should you play this game? Write a program to run simulations to answer this question.