

**CIS 122**

**Final Review**

# Logistics

- Course Evaluations
  - Fill them out by Wednesday
  - Feedback on lack of textbook
- Assignment 5
  - Some assignments submitted
  - Assignment help after class
- Final times
  - Wednesday 2:00 - 4:00
  - Friday 3:15 - 5:15

# Types

- Integers
- Floats
- Strings
- Booleans
- Lists
  - Nested Lists
- Dictionaries

# Programming Concepts

- Variables
- Functions
- Conditionals
- Recursion
- Iteration
  - Nested Loops
- Classes

# Types - Integers

- Numbers (without a decimal point)
  - 1
  - 42
  - -7
- Integer operations return integer results
  - $1 + 1 \rightarrow 2$
  - $2 * 3 \rightarrow 6$
- Watch out for integer division!
  - $10 / 5 \rightarrow 2$
  - $11 / 5 \rightarrow 2$

# Types - Floats

- Numbers (with a decimal point)
  - 1.5
  - 42.0
  - -7.
- Operations involving floats return floats
  - $1 + 1.5 \rightarrow 2.5$
  - $2 * 3.0 \rightarrow 6.0$
- Useful for float division
  - $10 / 5.0 \rightarrow 2.0$
  - $11 / 5.0 / 2.5$

# Types - Strings

- Sequences of characters (surrounded by quotes)
  - 'abc'
  - "Hello World"
  - '5'
- We can index into them
  - "abcdefg"[ 3 ] → 'd'
  - "abcdefg"[ -2 ] → 'f'
- We can slice them
  - "abcdefg"[ 2 : 5 ] → 'cde'
  - "abcdefg"[ 3 : ] → 'defg'

# Types - Strings (new!)

- We can iterate over them

```
for char in string:  
    print char
```

```
otherString = ""  
for i in range(len(string)):  
    otherString += string[ i ]
```

- We CAN'T modify them (strings are immutable)
  - `string[3] = 'a'`
  - `string.append('a')`

# Types - Booleans

- Only two values
  - True
  - False
- Generate from tests ( >, >=, <, <=, ==, != )
  - $4 < 5 \rightarrow$  True
  - 'x' in 'abcde'  $\rightarrow$  False
- Combine with logical connectives (and, or, not)
  - True and False  $\rightarrow$  False
  - True or False  $\rightarrow$  True
  - not True  $\rightarrow$  False

# Types - Booleans

- We can use them as conditions

- if, elif, else statements

```
if x < 5:  
    return 1  
else:  
    return -1
```

- while loops

```
while x < 5:  
    print x  
    x += 1
```

# Types - Lists

- Sequences of arbitrary elements
  - [ 1, 2, 3 ]
  - [ 'a', True, 42 ]
- We can index into them
  - [10, 20, 30, 40, 50] [ 2 ] → 30
  - [10, 20, 30, 40, 50] [ -2 ] → 40
- We can slice them
  - [10, 20, 30, 40, 50] [ 2 : 4 ] → [ 30, 40 ]
  - [10, 20, 30, 40, 50] [ : 3 ] → [ 10, 20, 30, 40 ]

# Types - Lists

- We can modify them
  - `L [ 2 ] = 100`
  - `L.append(100)`

- We can iterate over them

```
for b in [True, True, False, True]:  
    if b == False:  
        return False  
return True
```

```
for i in range(10):  
    print i
```

# Types - Lists

- We can nest them

```
nestedList = [ [10, 20, 30, 40],  
               [11, 21, 31, 41],  
               [12, 22, 32, 42],  
               [13, 23, 33, 43] ]
```

```
nestedList [2] → [12, 22, 32, 42]
```

```
nestedList [2][3] → [42]
```

# Types - Dictionaries

- Lists with arbitrary keys
  - letterCount = { 'a':5, 'b':7, 'c':2 }
  - sillyDict = { 0:0, 1:1, 2:2 }
- We can index dictionaries by keys
  - letterCount ['a'] → 5
- We can modify entries in dictionaries (they are mutable)
  - letterCount ['a'] = 4
  - letterCount ['c'] += 1
- We can add elements to dictionaries (they are mutable)
  - letterCount['d'] = 3

# Types - Collections

- Three collection types
  - Strings
  - Lists
  - Dictionaries
- Can test whether an element is present with **in** keyword
  - `'a' in 'abcde' → True`
  - `5 in [0, 1, 2] → False`
  - `'rabbit' in {'cat':True, 'dog':False} → False`
    - Search through keys
- Can get size of collection with **len** function
  - `len( [0, 1, 2] ) → 3`