

### Lecture 13

Conditionals and Iteration

## **Weekly Goals**

- Monday
  - No class

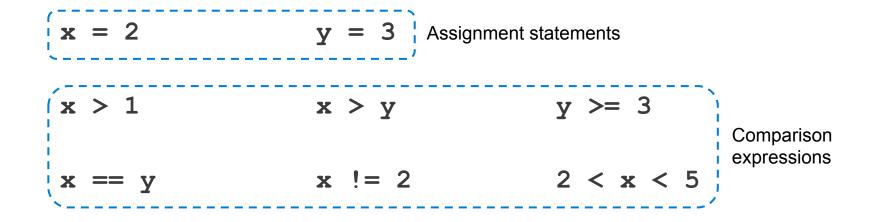
- Wednesday
  - Table review
  - Table examples
- Today
  - Booleans, conditionals, and iteration
  - Simulation

### **Announcements**

## **Comparison and Booleans**

### **Comparison Operators**

The result of a comparison expression is a bool value



## **Aggregating Comparisons**

Summing an array or list of bool values will count the True values only.

```
1 + 0 + 1 == 2
True + False + True == 2
sum([1 , 0 , 1 ]) == 2
sum([True, False, True]) == 2
(Demo)
```

### **Control Statements**

#### **Control Statements**

These statements *control* the sequence of computations that are performed in a program

- The keywords if and for begin control statements
- The purpose of if is to define functions that choose different behavior based on their arguments

### **Random Selection**

#### **Random Selection**

#### np.random.choice

- Selects uniformly at random
- with replacement
- from an array,
- a specified number of times

# **Appending Arrays**

## **A Longer Array**

- np.append(array\_1, value)
  - new array with value appended to array\_1
  - value has to be of the same type as elements of array\_1
- np.append(array\_1, array\_2)
  - o new array with array 2 appended to array 1
  - o array\_2 elements must have the same type as array\_1 elements

### **Iteration**

#### for Statements

• for is a keyword that begins a control statement

 The purpose of for is to perform a computation for every element in a list or array

## Optional: Advanced where

#### A Closer Look at where

```
t.where(array_of_bool_values)
```

returns a table
with only the rows of t for which
the corresponding bool is True.