Objects
Announcements
Review: Aggregation Functions
Project 2 Question 10: Fastest Words

```python
def time(match, player_num, word_index):
    """Returns the time it took player_num to type the word at word_index in match."""

def fastest_player(match, num_players, word_index):
    """Return which player typed word_index fastest.

>>> p0 = [2, 2, 3]
>>> p1 = [6, 1, 2]
>>> words = ['what', 'great', 'luck']
>>> [fastest_player(match(words, [p0, p1]), 2, i) for i in range(3)]
[0, 1, 1]
"""

fastest_p, fastest_time = None, None
for p in range(num_players):
    p_time = time(match, p, word_index)
    if fastest_time is None or fastest_time > p_time:
        fastest_p, fastest_time = p, p_time
return fastest_p

return min(range(num_players), key=lambda p: time(match, p, word_index))
```
Class Statements
Classes

A class describes the behavior of its instances

**Idea:** All bank accounts have a balance and an account holder; the Account class should add those attributes to each newly created instance

```python
>>> a = Account('John')
>>> a.holder
'John'
>>> a.balance
0

>>> a.deposit(15)
15
>>> a.withdraw(10)
5
>>> a.balance
5
>>> a.withdraw(10)
'Insufficient funds'
```

**Idea:** All bank accounts share a withdraw method and a deposit method
The Account Class

```python
class Account:
    def __init__(self, account_holder):
        self.balance = 0
        self.holder = account_holder
    def deposit(self, amount):
        self.balance = self.balance + amount
        return self.balance
    def withdraw(self, amount):
        if amount > self.balance:
            return 'Insufficient funds'
        self.balance = self.balance - amount
        return self.balance
```

__(Demo)__

```python
>>> a = Account('John')
>>> a.holder
'John'
>>> a.balance
0
>>> a.deposit(15)
15
>>> a.balance
15
>>> a.withdraw(10)
5
>>> a.balance
5
>>> a.withdraw(10)
'Insufficient funds'
```
Animation Demos by Hany Farid