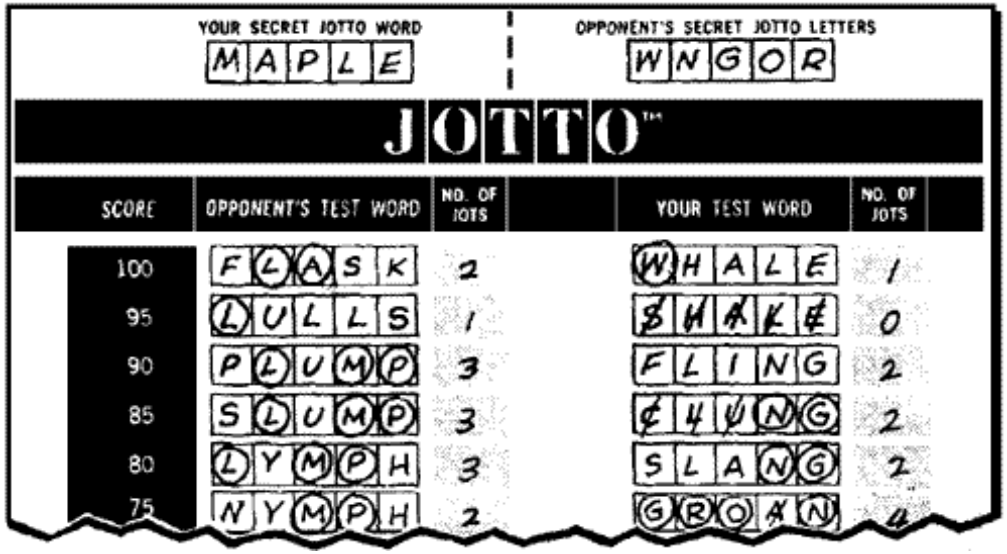
Compsci 101
Dictionaries, Jotto
Live Lecture



Susan Rodger
Nicki Washington
March 23, 2021

Announcements

- APT-5 due tonight! March 23
- Nothing due on Thursday this week, consulting hours shorter, get ahead on Assign 4!

- Assignment 4 Hangman due Tues. March 30
  - ASGN4 Sakai quiz – do early! Tests understanding
- APT-6 is now out, due Thurs. Apr 1

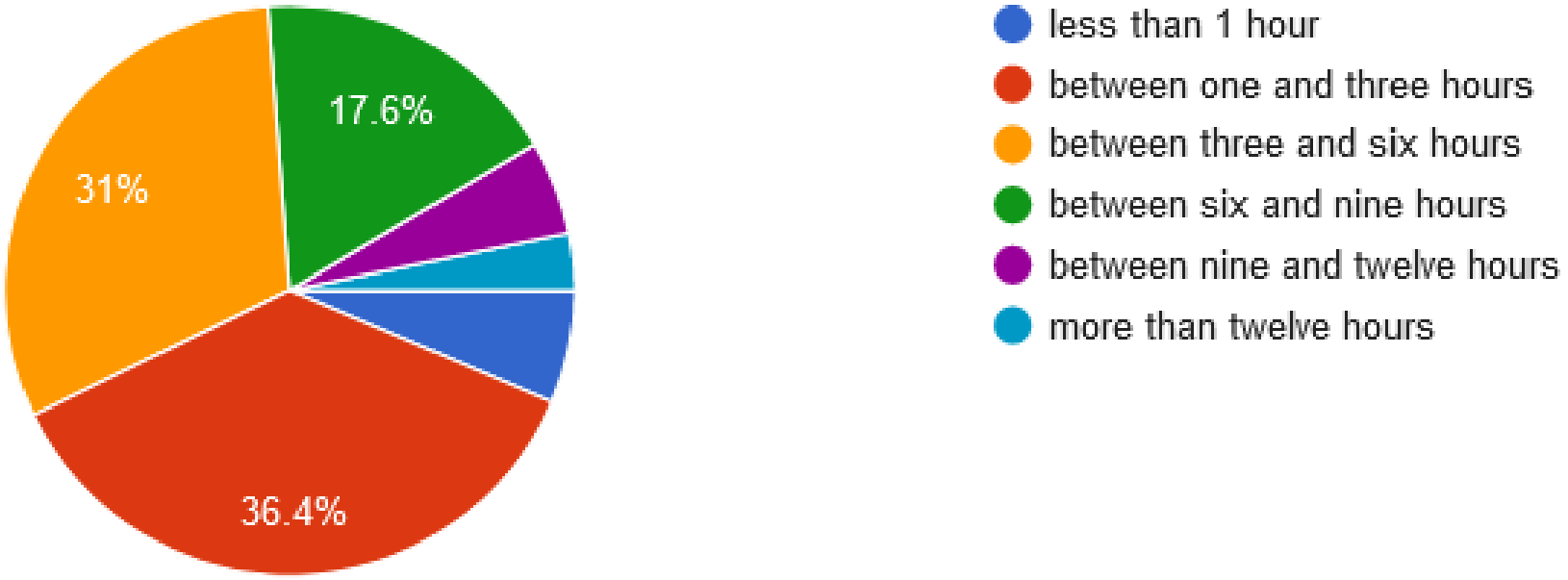
- Assign 5 is out Thursday, it builds on Assgn 4

- Lab 8 Friday, there is no prelab

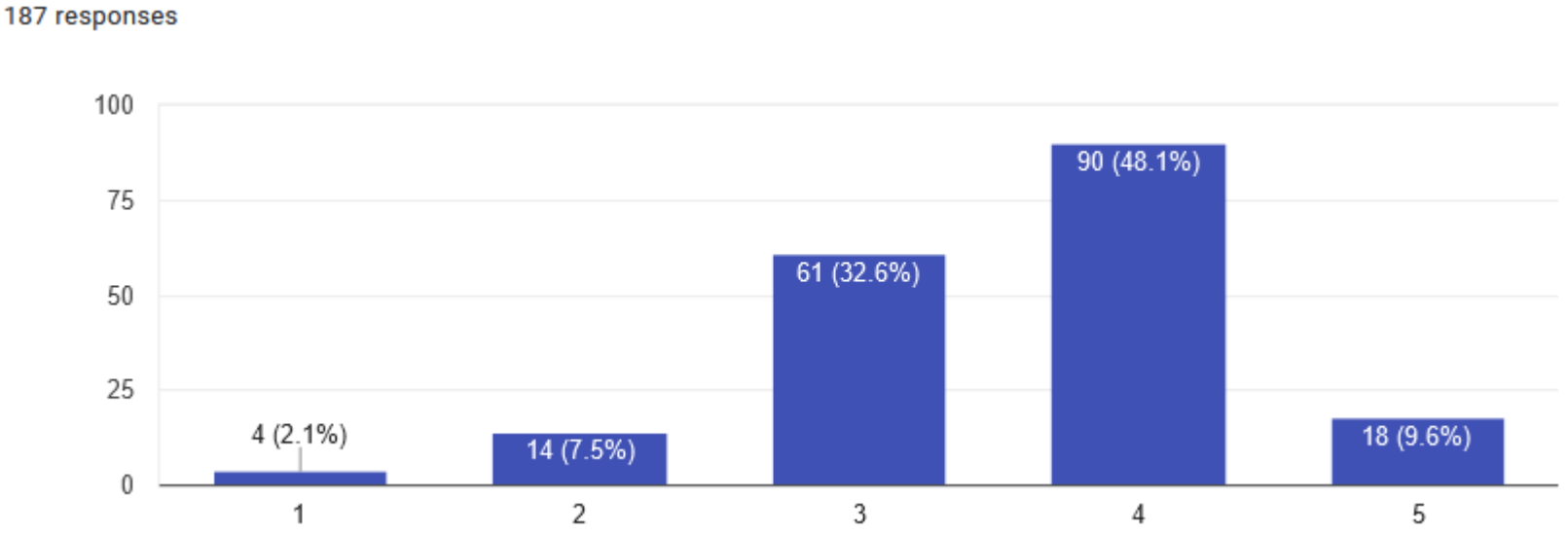
Exam 2….

- Exam 2 – not back yet, do not discuss with anyone til we hand it back.

Exam 2 Time Preparing

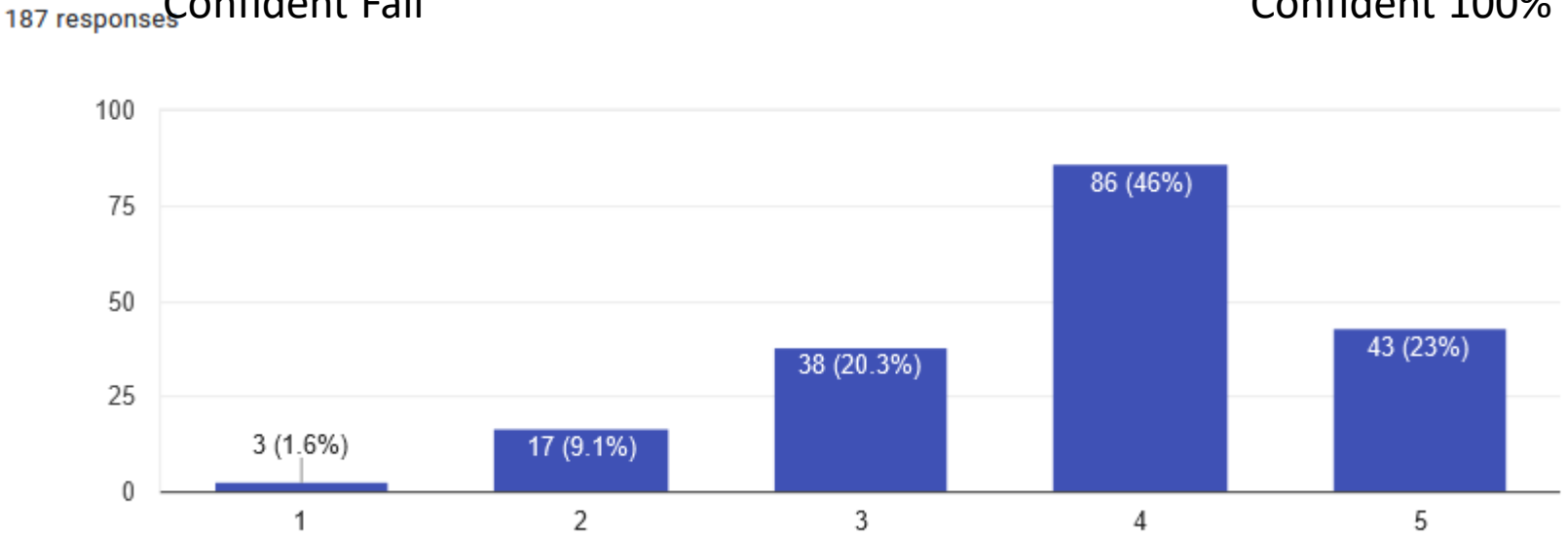


Thoughts Before and After taking Exam 2



Confident Fail

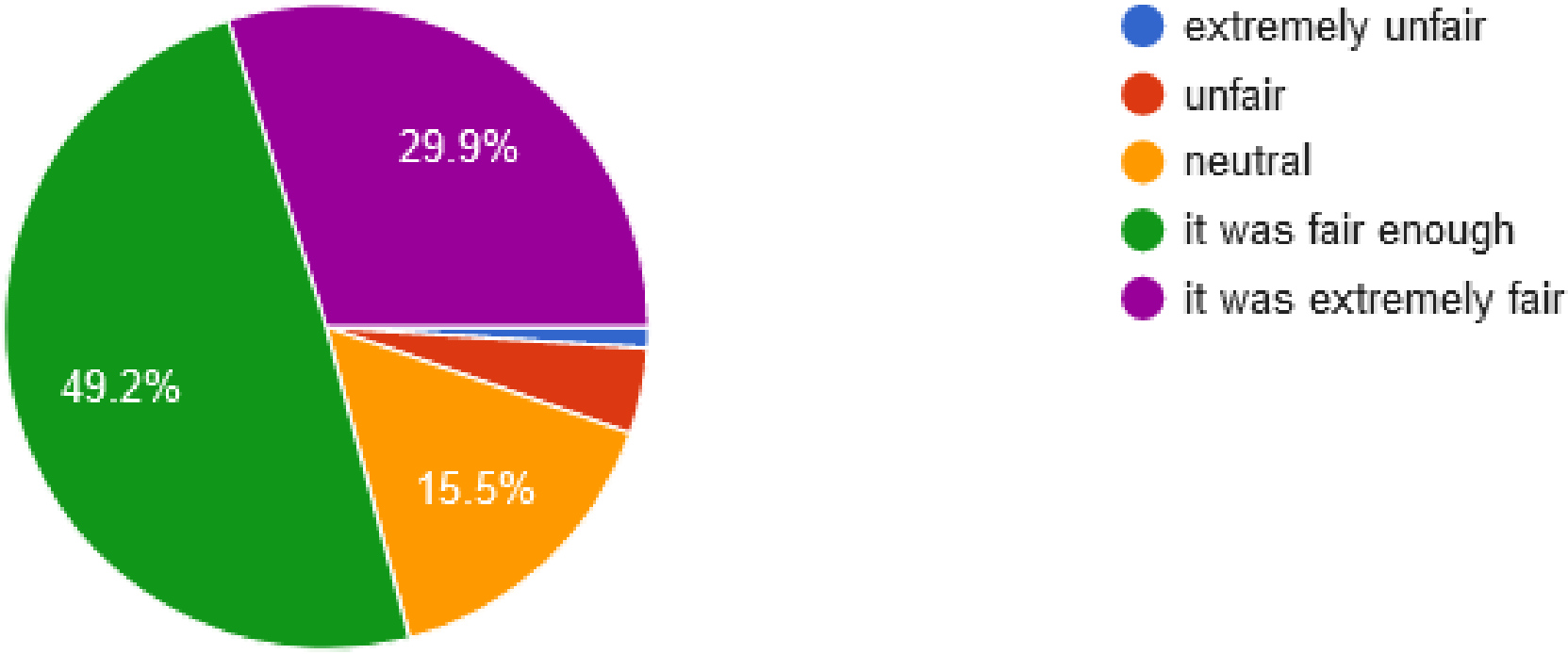
Confident 100%



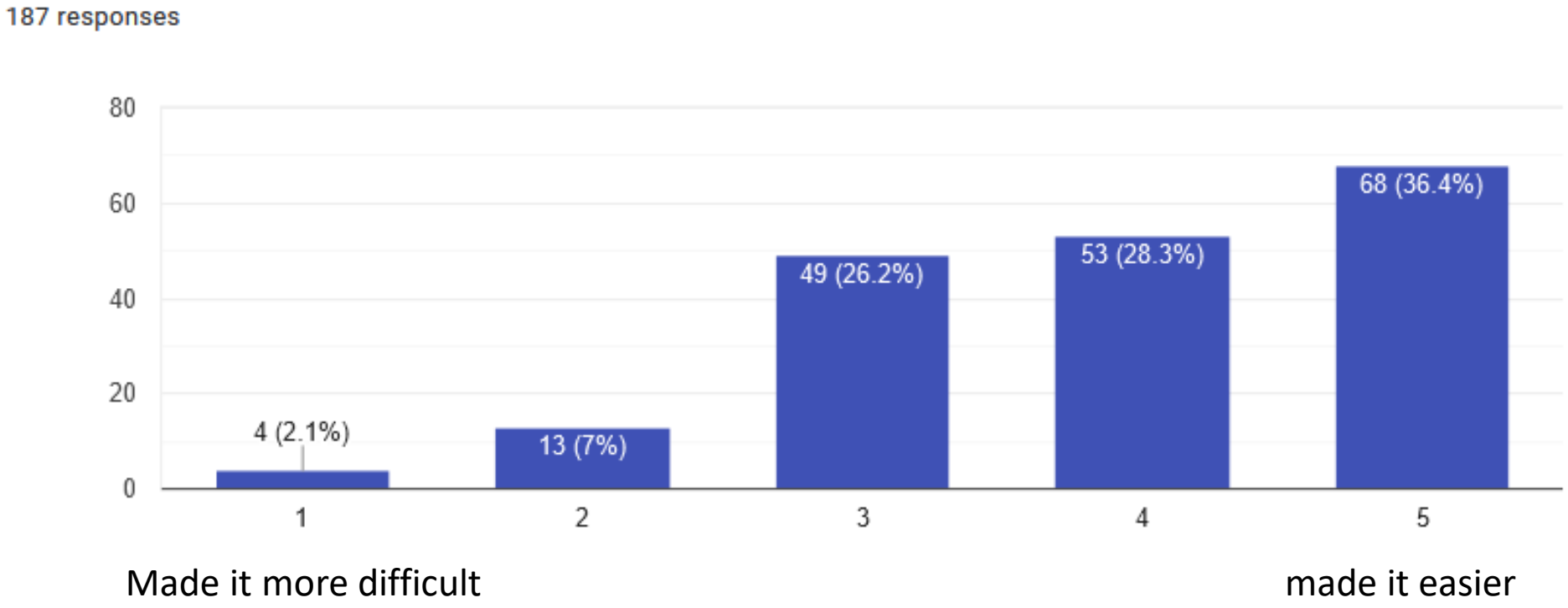
Did Poorly

think did well

Was Exam 2 Fair?



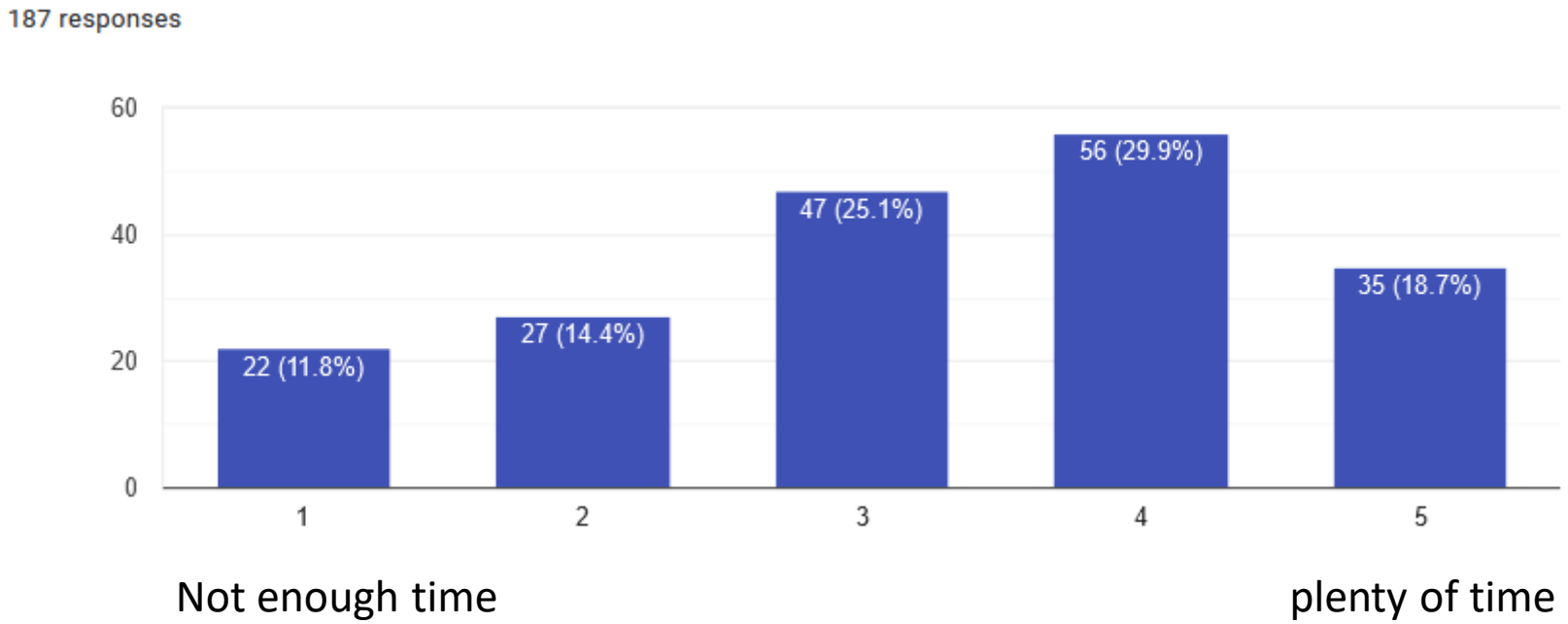
How was it taking Exam 2 online?



Made it more difficult

made it easier

Did you have enough time for Exam 2?



Not enough time

plenty of time

PFTD

- Dictionaries cont.
  - Functions
- A little on sorting
- Jotto!
  - How to approach a large project
  - Splitting functionality
  - Putting it all together

Short Code and Long Time

- See module WordFrequencies.py
  - Find # times each word in a list of words occurs
  - We have tuple/pair: word and word-frequency

```
37  def slowcount(words):
38      pairs = [(w,words.count(w)) for w in set(words)]
39      return sorted(pairs)
```

- Think: How many times is **`words.count(w)`** called?
  - Why is **`set(words)`** used in list comprehension?

WordFrequencies with Dictionary

- If start with a million words, then…
  - We look at a million words to count # "cats"
  - Then a million words to count # "dogs"
  - Could update with parallel lists, but still slow!
- Look at each word once: dictionary!

- Key idea: use word as the "key" to find occurrences, update as needed
  - Syntax similar to **`counter[k] += 1`**

Using fastcount

- Update count if we've seen word before
  - Otherwise it's the first time, occurs once

```
28  def fastcount(words):
29      d = {}
30      for w in words:
31          if w in d:
32              d[w] += 1
33          else:
34              d[w] = 1
35      return sorted(d.items())
```

WOTO-1 Counting Dictionaries
http://bit.ly/101s21-0323-1

- In your groups:
  - Come to a consensus

Dictionary Syntax and Semantics

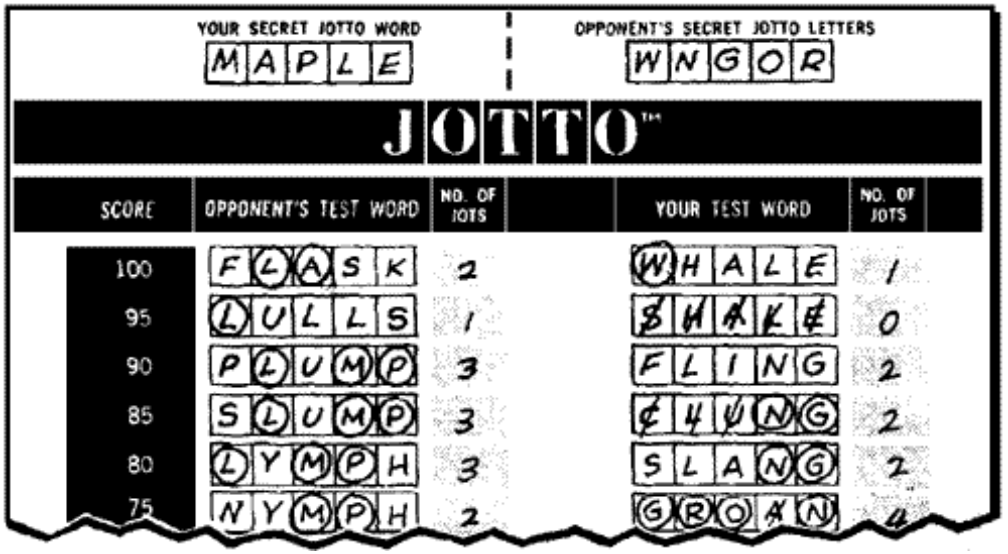
Syntax/Function	Meaning
`d = {}`	Initialize empty dictionary d
`d.keys()`	Collection of keys in dictionary
`d.values()`	Collection of values
`d[key]`	Value associated with key (error if key not in d)
`d.get(key,dv)`	Value associated with key (dv if key not in d, dv is optional)
`d.items()`	Collection of (key,value) tuples in d

WOTO-2 Dictionary Functions
http://bit.ly/101s21-0323-2

- In your groups:
  - Come to a consensus

How to approach Hangman: Jotto

- https://en.wikipedia.org/wiki/Jotto
- http://jotto.augiehill.com/single.jsp
- No letters repeat – have to agree on this
- Shall we play a game?



Write program where Computer Guesses Your Word

- **Brute force, no thinking or eliminating letters**
  - Pick a word at random, guess it
  - If x letters in common? Only keep words with x letters in common
  - Repeat until guessed

WOTO-3 Approaching Implementation http://bit.ly/101s21-0323-3

- In your groups:
  - Come to a consensus

- What is needed?
- What order should the code do things?

Start with Blank Screen

1. Computer gets a list of words

Initialization

2. Computer chooses a word at random
3. User/player enters # letters in common
4. Only keep words with that # in common

Loop

Iterative Programming!

- Start with a task
- Implement only that task
- Write some code to check that the code works
  - Run and debug until it works
- Repeat

Should you write all the code first and then run it?

"Debugging is twice as hard as writing the code in the first place. Therefore, If you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."
- Brian Kernighan (original Unix contributor)

SimpleJotto.py

- We have a file of five letter words: **`kwords5.txt`**
  - Would you like to play a game?

- Let's start! Simple version that sort of works ☺

Jotto Step 1

- Read the file kwords5.txt
  - Read the file kwords5.txt
  - Don't continue until we know this works

Jotto Step 2

- Pick a word at random
  - Pick a word at random, show the user
  - Don't continue until we know this works

Jotto Step 3

- Get number of letters in common
  - Get # letters in common, do something?
  - Don't continue until we know this works

WOTO-4 More on Jotto http://bit.ly/101s21-0323-4

- In your groups:
  - Come to a consensus

- What is needed?
- What order should the code do things?

Jotto Step 4

- Only keep words with that number in common

- Example: User enters 0, keep only words with 0 letters in common (replace 0 with 1, 2, …, N)
  - What are the steps here?
  - What's similar?
  - What do we do to solve this?
- Helper function: commonCount

commonCount

- Given two words, return # letters in common
- Similar to SandwichBar?
  - Sandwich ingredients -> letters in a word

Finishing SimpleJotto

- When is the game over? How to signal that
  - Interaction is via # letters in common
- Add functionality: number of guesses?
  - Remind the user where they are

Writing and Testing functions

- We'd like to test the function isolated from game
  - Ensure we don't have to play to test it
  - Unit testing similar to APT tests

- Can do this in your main!
  - Remove for final submission for hand grading
- Also use the APT testing framework and Gradescope

Summary: Jotto

- Break down entire game into steps
- Recognize what steps belong where
  - Initialization: Before loop
  - Inside loop
  - Anything after loop? End of game stuff
- Code one step at a time (or one function)
  - Test if that step worked (or submit to Gradescope)