# Content of Presentation

1.0
Introduction

2.0
React

3.0
App Structure

4.0
Code Walkthrough

5.0
Data & LexiStats

6.0
Questions


# INTRODUCTION


HTTP://KURTWILLIAM.COM/WORD-SANDWICHES/


# REACT

# USING REACT

Break App up into **Components**
(reusable User Interface components)

# USING REACT

Use **States** (how component renders and behaves) and **Props** (property passed from parent telling component how to render)

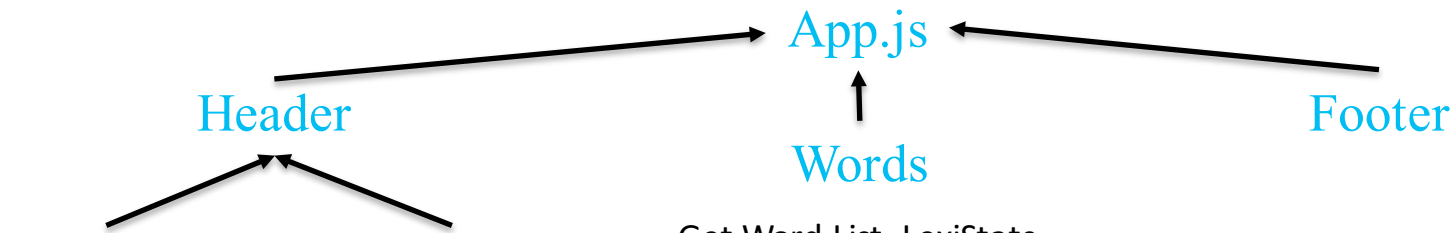
# USING REACT

Changes in Components update on the same page, so no need to refresh or redirect!

Facebook and Instagram use React (think of scrolling on Facebook news feed – the feed refreshes without having to refresh the page)


# APP STRUCTURE

# STRUCTURE




*State from Difficulty is passed up – Difficulty > Header > App, then passed down - App > Words as Props.*


# CODE WALKTHROUGH

# WORDS.JS

On page load (componentDidMount()) pull in the words and render the game box

# WORDS.JS

On click of the ‘feed me’ button filter the word list based off of the settings, generate game string and answer key, and start timer

# GENERATING GAME STRING

1. Take the ~16k words (filtered to be at least 4 characters long)
2. Filter by length, shuffle, then take a few and push them into the answer key array
3. For each word, push another answer key word into a random point in that string. Then recombine string

# WORDS.JS

User types input and timer runs until it reaches 0

# USER INPUT

1. User types a ‘word’ they see in the string
2. Check to see if the input first matches an item in the answer key, then search the string to match characters
3. If both pass the check, add 5 seconds. If not, remove one second

# WORDS.JS

Complete the game, calculate score based on word frequency of use, word length, number of words, and time remaining. Show definitions of words if user requests


# DATA AND LEXISTATS

# WORD LIST

1. Import Word List from .txt file – Google Searches, top 20k words
2. Filter and shuffle Word List depending on the state of the settings and difficulty
3. Store in an answer key array we can use to make LexiStats and Definitions call to Oxford Dictionaries

# WORD LIST

["steak", "demux", "wines”, “wallace”, "armies", "sting", "awards”]

# LEXICAL STATISTICS

1. Make an Ajax call requesting normalizedFrequency for each answer key word
2. Filter data then use it to add points for the multiplier

```
ajax({
        url: `http://proxy.hackeryou.com`,
        type: 'GET',
        dataType: 'json',
        contentType: 'json',
        data:{
                reqUrl: `https://od-
api.oxforddictionaries.com:443/api/v1/stats/frequency/words/en/?corpus=nmc&lemmas=${request}`,

                xmlToJSON: "false",
                proxyHeaders:{
                        "Accept": "application/json",
                        "app_id": app_id,
                        "app_key": app_key
                },
        }
})
```

# DEFINITIONS

1. On click of the word displayed in the game over screen make a call to Oxford Dictionaries to get a definition of the word

# QUESTIONS?