Front-End Framework

Junghoo Cho

cho@cs.ucla.edu

Example: Google Suggest

- "No USC" extension to Google Suggest
  - No USC queries are allowed!
- Q: How should we modify the code? What should we add?
- Review the changes in the new code
- As app becomes more complex, the code becomes uglier an difficult to maintain

Challenges of Web Development

- Q: What will Gmail code look like?
- Challenges
  1. Code complexity
     - 10,000s of lines of JavaScript, HTML, CSS code
  2. Lack of modularity
     - Many global variables (both in JS and CSS) and name conflicts
     - Code maintenance difficulty
  3. Code reusability
     - Do we have to start from scratch all the time?
- Idea
  - Can we develop and provide “library” of commonly-used independe modules?
  - Can we program at a higher-level than DOM elements?
- Q: How can we split complex program into independent “mo

Components

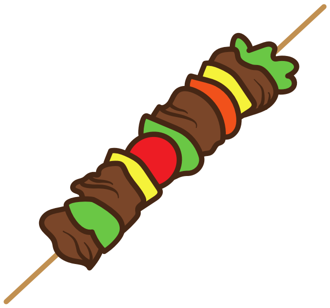
- Basic idea: Any complex UI app consists of simple “compone
    - Each component is mostly generic and independent of others
    - Structure and develop the app to exploit this independence and reu

Component-Based Development

1. Split the app into hierarchy of simpler components
2. Develop each component independently (with unit testing)
3. Combine simple components into more complex ones

- Advocated first by React, now adopted by all popular front-e frameworks
  - Reduces development and maintenance complexity
  - Local changes are limited to a particular component

kebab-case vs camelCase



kebab-case

camelCase

- Many frameworks often use
  - camelCase in JavaScript code: e.g., `SearchBoxComponent`
  - kebab-case in HTML (and filenames): e.g., `<app-search-box>`
- This convention is because
  - HTML is *not* case sensitive, but dash *is* allowed in identifiers
  - JavaScript *is* case sensitive, but dash is *not* allowed in identifiers

What We Learned

- Need for front-end framework
  - Any help for modular and reusable development of front-end code?
- Component-based development
  - Modern frameworks split code into a collection of modular “compo
- kebab-case vs camelCase
- Next
  - We will learn Angular as example front-end framework
    - Similar to other popular frameworks
    - Requies TypeScript