👾

Persistent Storage

Created	@January 31, 2022 3:11 PM
Type	Lecture
Week	6
Lecture #	1
Lecture URL	https://youtu.be/yyh0kjqA5_Q
Notion URL	https://21f1003586.notion.site/Persistent-Storage-25e0fca725244a279da8609a2ebf7f08

What is persistent storage?

- Vue data is lost (or set to initial stage) whenever page is reloaded
- Persistent state without the need for server

Why persistent storage?

- True persistence is possible at the server
- Offline →
    - Work without requiring a server connection
- Simple apps →
    - No need to go to the server for simple requests
- Configuration →
    - Local configurations possible — not needed by the server

How to persistent storage?

- Cookies →
    - JS `setCookie()` can be used for simple data
    - Limited storage — usually session temporary — removed on browser restart
- `localStorage` →
    - API to save simple key → value entries
    - Complex objects should be stringified — use JSON
- IndexedDB →

    - Transactional database system
    - Object-oriented JS-based DB
    - store and retrieve objects with a key

`localStorage`

- WebStorage API
    - https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
- sessionStorage
    - storage ends with session (browser restart)
    - more storage than cookies ~5MB
- localStorage
    - persists across browser restarts
    - browsers may implement limits to avoid overload

Example: https://vuejs.org/v2/cookbook/client-side-storage.html

Vue - Form Validation

Created	@January 31, 2022 3:36 PM
Type	Lecture
Week	6
Lecture #	2
Lecture URL	https://youtu.be/J_b2hmWUX44
Notion URL	https://21f1003586.notion.site/Vue-Form-Validation-25ab5089c6e14b2586131ea5db752f8e

Form Validation

- Check whether the data entered in a form meets certain criteria
- Simple checks in the browser
  - Text field contains number, email address etc.
  - Select field has at least one entry
  - Empty fields
- Server side checks
  - *Never trust the client*
  - Essential in many cases for security
  - More costly, increase in server load

Vue & Validation

- Data binding and reactivity
  - Easy updates of parts of the DOM
  - Selectively display error messages: `v-if, v-show`
- `v-model` connects fields with JS variables for easy processing
- `preventDefault()` — stop normal processing of the form unless check successful
  - connect as `submit` event handler

Custom Validation

- Example → custom email check

    - Specific domain, specific number of characters, etc.
- Example → check for certain overall condition
    - All numbers add up to given value
- Need to prevent regular form validation
    - `novalidate=true` in form definition

https://vuejs.org/v2/cookbook/form-validation.html

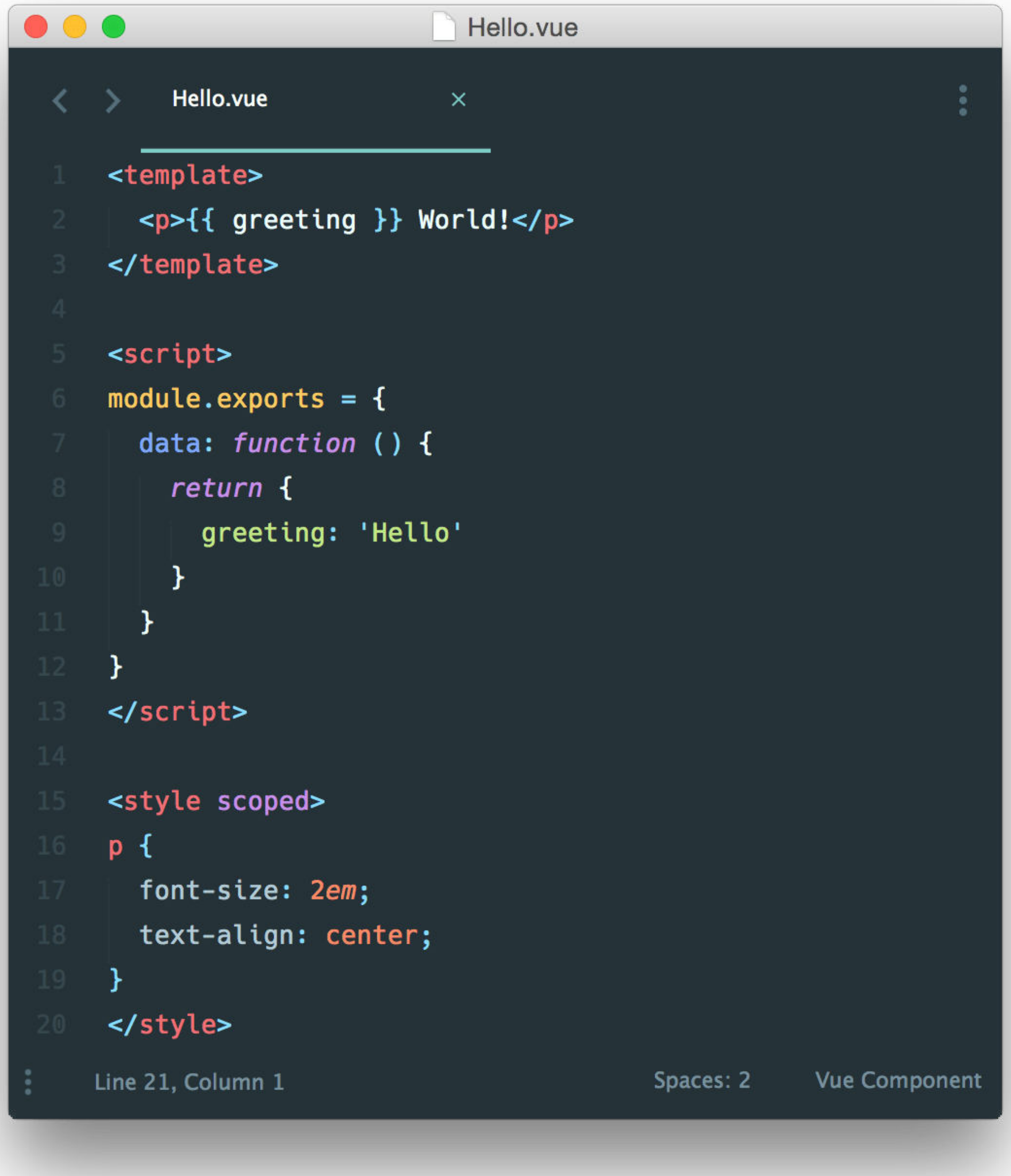
Vue - Managing Components

Created	@January 31, 2022 4:22 PM
Type	Lecture
Week	6
Lecture #	3
Lecture URL	https://youtu.be/zjGtbOZAS9A
Notion URL	https://21f1003586.notion.site/Vue-Managing-Components-a08853cdc818414d99931208e2200aec

Need for Single File Components

- Global namespace
    - Unique names needed for each components
        - can be difficult when importing from other sources
- String templates
    - Harder to edit and manage with regular editors
- CSS
    - No block scoping — only global CSS
    - Not modular unlike HTML (templates) and JS (components)
- No build step
    - Backwards compatibility not easy, cannot use tools like Babel

SFC Structure



```
<template>
  <p>{{ greeting }} World!</p>
</template>

<script>
module.exports = {
  data: function () {
    return {
      greeting: 'Hello'
    }
  }
}
</script>

<style scoped>
p {
  font-size: 2em;
  text-align: center;
}
</style>
```

Source: https://vuejs.org/v2/guide/single-file-components.html

Separation of Concerns

- We would normally like separation
  - Semantic content — HTML
  - Presentation — CSS
  - Logic — JS
- This does not require separate files

Extra “Tooling”

- JS cannot directly read `.vue` files
- Compilation step needed
  - Convert `.vue` to `.js` + `.css` + `.html`
- Webpack, ESBuild, Vite etc. — package files
- `npm` → Node package manager

  - Systematically managed JS modules
  - Import new modules as needed

Mostly managed from the Command Line interface

Vue - Testing

Created	@January 31, 2022 4:33 PM
Type	Lecture
Week	6
Lecture #	4
Lecture URL	https://youtu.be/D8snI7H0TZk
Notion URL	https://21f1003586.notion.site/Vue-Testing-a1bd2a6f0576452eac327d69fa1382db

Testing Vue applications

- Unit testing
    - As with any other software process
    - Components are good units to test
    - Can “mount” into a testing DOM
- E2E (End-to-End) testing
    - Full application including backend
- Cross-browser testing
    - Compatibility with older browsers
    - Diminishing returns — is it worth supporting all older browsers

Test mechanisms

- Set up fixtures →
    - prepared data
- Test suite →
    - Collect several tests together
    - Test one component
- Tools →
    - mocha, chai, jest → supporting functions to test for presence/absence of elements

https://vuejs.org/v2/cookbook/unit-testing-vue-components.html