🏠 Home(/)

0x04. React inline styling

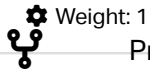Front-end My Plan(planning#me) JavaScript ES6 React

⚙️ Weight: 1

Projects(/projects/current)

📅 Project over - took place from Aug 5, 2024 6:00 AM to Aug 7, 2024 6:00 AM

✅ An auto review will be launched at the deadline
QA Reviews (/corrections/to_review)

? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

🎓 Curriculums(/dashboards/my_curriculums)

📄 Concepts(/concepts)

💬 Conference rooms(/dashboards/video_rooms)

🖥️ Servers(/servers)

>_ Sandboxes(/user_containers/current)

🔧 Tools(/dashboards/my_tools)

🎞️ Video on demand(/dashboards/videos)

👥 Peers(/users/peers)

Resources

Discord(https://discord.com/app)

Read or Watch:

- Aphrodite (/rltoken/DfGvHrI6zsKtKFEYWajLoA)
- Inline styling (/rltoken/s2il-pXJvk6Lxj6BmAWG9Q)
- Enzyme Render (/rltoken/hX2sEidBZzVuGUNhCv3MWA)
- Enzyme Prop (/rltoken/lhm4Qeyjz3oyCa6C4zgzCQ)
- CSS Profile(/users/key_profile)FC5DpSqXeHIccDZDg)
- CSS Media queries (/rltoken/6-GPqaJjRsdE9qgy_8ZTpg)

- CSS Animations (/rltoken/z02z0ouci-gTwD0zDk5Kcw)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/1SdjdzLeEkHh5JYyHEb3kw), **without the help of Google**:

- the differences between using a CSS file and inline styling
- how to use a CSS-in-JS tool like Aphrodite
- how to use conditions within JS to apply different styles
- how to use responsive design and make the application show a different UI according to the screen size
- how to create small animations within the app

Requirements

- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using `node 12.x.x` and `npm` `6.x.x`
- Allowed editors: `vi`, `vim`, `emacs`, `Visual Studio Code`
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory

Tasks

0. Inline styling

mandatory

- Copy over the `task_5` directory from the `0x04. React components` project (We'll be using it as the base for this project)
- Rename the `task_5` directory to `task_0`

Modify the `CourseListRow` component in `task_0/dashboard/src/CourseList/CourseListRow.js`:

- Using inline styling, change the background color of a row to `#f5f5f5ab`
- Using inline styling, change the background color of a header row to `#deb5b545`
- If needed, modify the test file so every test pass

Tips:

- For better performances, never create and pass an object to an element directly. Use a constant instead
- Use the `isHeader` prop to easily pick the style you want to apply to the `tr` tag

Requirement:

- Even if the modification is small, make sure that your test suites are still passing. Especially the file `CourseListRow.test.js`

Repo:

- GitHub repository: `alx-react`

- Directory: `0x04-React_inline_styling`
- File: `task_0/dashboard/src/CourseList/CourseListRow.js`, `task_0/dashboard/src/CourseList/CourseListRow.test.js`

1. install Aphrodite

`mandatory`

Install Aphrodite using npm with:

```
npm install --save aphrodite
```

Modify the `App` component in `task_1/dashboard/src/App/App.js`:

- Modify the component to use Aphrodite within the js file
- Define the styling for the body and the footer within the file
- Delete the file `App.css` and the import

Modify the `BodySectionWithMarginBottom` component in

`task_1/dashboard/src/BodySection/BodySectionWithMarginBottom.js:`

- Modify the component to use Aphrodite within the js file
- Define the styling for the margin within the file
- Delete the file `BodySection.css` and the import

Modify the `CourseList` component in `task_1/dashboard/src/CourseList/CourseList.js`:

- Modify the component to use Aphrodite within the js file
- Define the styling for the list within the file
- Remove the styling for the list within the `CourseList.css` file

Modify the `Header` component in `task_1/dashboard/src/Header/Header.js`:

- Modify the component to use Aphrodite within the js file
- Define the styling for the logo and the header within the file
- Delete the file `Header.css` and the import

Modify the `Login` component in `task_1/dashboard/src/Login/Login.js`:

- Modify the component to use Aphrodite within the js file
- Define the styling for the margin within the file
- Delete the file `Login.css` and the import

Modify the `Notifications` component in

`task_1/dashboard/src/Notifications/Notifications.js:`

- Modify the component to use Aphrodite within the js file
- Define the styling for the notifications panel within the file
- Remove the styling for the notifications panel from the `Notifications.css`

Make sure the test suites are still passing!

Tips:

- Look into using `StyleSheetTestUtils.suppressStyleInjection` at the top of your test file, to prevent issues with style injections

Requirements:

- At this point, the UI should look exactly the same with the inline styling as it was with the CSS files

Repo:

- GitHub repository: `alx-react`
- Directory: `0x04-React_inline_styling`

- File: `task_1/dashboard/src/App/App.js`,

  `task_1/dashboard/src/BodySection/BodySectionWithMarginBottom.js`,

  `task_1/dashboard/src/CourseList/CourseList.js`,

  `task_1/dashboard/src/Header/Header.js`, `task_1/dashboard/src/Login/Login.js`,

  `task_1/dashboard/src/Notifications/Notifications.js`,

  `task_1/dashboard/src/App/App.test.js`, `task_1/dashboard/src/BodySection/BodySectionWit`

  `task_1/dashboard/src/CourseList/CourseList.test.js`,

  `task_1/dashboard/src/Header/Header.test.js`,
  `task_1/dashboard/src/Login/Login.test.js`,

  `task_1/dashboard/src/Notifications/Notifications.test.js`

Check submission

1. Conditionally applying style

`mandatory`

Modify the `NotificationItem` component in
`task_2/dashboard/src/Notifications/NotificationItem.js`:

- Modify the component to use Aphrodite within the js file
- Define the stylings for the urgent and default items
- Using condition, apply the styling to the `li` element
- Delete the `Notifications.css` file and remove any import

Modify the `NotificationItem.test` suite in

`task_2/dashboard/src/Notifications/NotificationItem.test.js`:
- Make sure that tests are still passing

Modify the `CourseListRow` component in `task_2/dashboard/src/CourseList/CourseListRow.js`:
- Modify the component to use Aphrodite within the js file
- Define the styling for the different type of rows (default rows, header rows)
- Define the styling for the different type of `th` elements
- Delete the `CourseList.css` file and remove any import

Modify the `CourseListRow.test` suite in

`task_2/dashboard/src/CourseList/CourseListRow.test.js`:

- Make sure that tests are still passing
- Test properties one by one if easier

Requirements:

- Use conditions as much as you can, do not repeat elements
- At this point, the UI should look exactly the same with the inline styling as it was with the CSS files

Tips:

- You can either use conditions or use an array to apply the different styling. Conditions are usually more robust
- Testing might become difficult with the different variations. Look into the Enzyme documentation for `render`, `contains`, `prop`, and `html`

Repo:

- GitHub repository: `alx-react`
- Directory: `0x04-React_inline_styling`
- File: `task_2/dashboard/src/Notifications/NotificationItem.js`,
  `task_2/dashboard/src/Notifications/NotificationItem.test.js`,
  `task_2/dashboard/src/CourseList/CourseListRow.js`,
  `task_2/dashboard/src/CourseList/CourseListRow.test.js`

Check submission

3. Responsive design

mandatory

Let's make the application responsive to the screen size using media queries. We are going to only focus on large screen and screens with a width under 900px

Modify the component `Login` in `task_3/dashboard/src/Login/Login.js`:

- Make sure that a label and an input are on each line
- Make sure that the button is on a new line
- The screen should look like the image below:

Modify the component `Notifications` in `task_3/dashboard/src/Notifications/Notifications.js`:

- When the panel is open, it should take over the entire screen
- There should be no padding because of the `ul` element
- The font size of the text should be 20px

Modify the component `NotificationItem` in `task_3/dashboard/src/Notifications/NotificationItem.js`:

- The item should take the entire screen width
- A black border should be displayed at the bottom
- The font size of the text should be 20px
- The padding for the item should be `10px 8px`

Requirements:

- When the notifications panel is open, the screen should look like the image below:

Repo:

- GitHub repository: `alx-react`
- Directory: `0x04-React_inline_styling`
- File: `task_3/dashboard/src/Login/Login.js`, `task_3/dashboard/src/Notifications/Notifications.js`, `task_3/dashboard/src/Notifications/NotificationItem.js`

4. Animation

mandatory

Let's create an animation that we can display when the user hovers on the `Notifications` menu or when there is a new notification. In `task_4/dashboard/Notifications/Notifications.js`:

- Create one object containing the CSS frames to make the opacity change from `0.5` to `1`
- Create one object containing the CSS frames to make the element bounce. You can play with translateY and alternate from 0px to -5px and 5px

Modify the styling for the menu item to:

- Float on the right of the screen over every element
- The background color should be `#fff8f8`
- Show the pointer cursor when hovering the element
- On hover, animate the element with the two new animations. The duration for the opacity change should be 1s, and the duration for the bouncing effect should be 0.5s. The animation should repeat 3 times only
- When the list of notifications is visible, hide the menu item

Requirements:

- When the notifications panel is hovered or opened, the UI should look like the image below:

Repo:

- GitHub repository: `alx-react`
- Directory: `0x04-React_inline_styling`
- File: `task_4/dashboard/src/Notifications/Notifications.js`

☰
(/)

🏠 Home(/)

📅 My Planning(/planning/me)

⑃ Projects(/projects/current)

✓ QA Reviews I can make(/corrections/to_review)

? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

🎓 Curriculums(/dashboards/my_curriculums)

📄 Concepts(/concepts)

💬 Conference rooms(/dashboards/video_rooms)

🖥 Servers(/servers)

>_ Sandboxes(/user_containers/current)

🔧 Tools(/dashboards/my_tools)

🎞 Video on demand(/dashboards/videos)

👥 Peers(/users/peers)

🎮 Discord(https://discord.com/app)

🔍

My Profile(/users/my_profile)