Web Development Tasksheet

Thursday, September 2, 2021 10:41 PM

Relative, Fixed and Absolute Position:

- Static is the default value of the element. When we don’t set define element's position value, default value is static according to the normal flow of the document.
- Absolute position is relative to it's nearest parent element. Elements with position absolute are basically removed from the normal document flow, and other elements behave as that element is not in the document.
- Fixed position is similar to absolute position, fixed element is also removed from the normal document flow. But the difference is, unlike absolute positioned elements, fixed positioned elements are always relative to <html> .

Var, Let and Const:

- Var and let are mutable, we can reassign/update their values later. But once we assigned a value to a variable using const, we can not change is, it is immutable.
- Var is an old way of defining variables, and it is function scoped. Let is introduced in 2015(ES6), and it is block scoped.

What must be developed first, mobile or desktop?

I think mobile should develop first, for mobile development CSS is shorter and less chance of errors, less cluttered. When you first make a mobile website, later you just have to expand upon the CSS you just wrote for desktop…but if you do opposite, and go desktop first then you override. For large projects messy CSS with many overrides can go out of hand and is more error prone. Starting with only desktop view might give you hard time making it mobile friendly later.

Alert and Alert dialog:

- Alert displays specific message, error or arrival of information, or gives you some kind of a warning, and we need to click okay to proceed. For example "Congratulations, your account is created successfully"
- Alert dialog is used to notify the user of some urgent information that demands his/her attention.

It displays the dialog message with OK and Cancel button. For Example "Are you sure you want to continue?"

Link, Generic Button and Submit Button:

- Link take you to other place/website, button does something, perform some action.
- Generic button won't submit the form, but one with type="submit" will, submit form submit the forms they are in by default. Normal button don’t do anything by default.

Component Cycle:

So every react component has it's own life cycle when created. Like humans, the born, grow, and then die. Different hooks/phases get trigger on different events.

1. componentWillMount(): very first method, it gets executed before initial rendering of the component.
2. componentDidMount(): next method, it runs after initial rendering.
3. componentWillRecieveProps(): whenever component is about to receive new props this method gets executed.
4. shouldComponentUpdate(): it gets executed after receiving new props, but before rendering. It can return false from preventing it from rendering. But if it returns true then next method…
5. componentWillUpdate(): it gets executed after receiving new props, before rendering.
6. componentDidUpdate(): it gets executed after component is updated and flushed to DOM.
7. componentWillUnmount(): whenever you remove component from DOM this method runs.

If I have experince with redux, webpack, code efficiency, and performance?

I have experience with webpack, code efficiency and performance. But I am relatively new with redux, still learning.