React Lifecycle Reference Sheet for Re-frame

Intro

When you create a Form-3 component in Re-frame, you need to call `reagent/create-class` with a map containing the following methods. Only `:reagent-render` is required.

`{` ;; Information

;; Name your component so that you can see it in error messages.

`:display-name`

```
"my-component"
```

Sets the name of the component so that you can identify it later in error messages.

;; Mounting

;; After constructing the instance of the component, React will mount it into the DOM.

`:component-did-mount`

```
(fn [comp]
  (let [node (reagent/dom-node comp)]
    (js/console.log "DOM node:" node)))
```

Called once just after mounting the component into the DOM. At this point, you can get access to the actual DOM node if you need to modify it.

;; Rendering

;; Just after mounting and any time the arguments, subscriptions, or Reagent Atoms change.

`:reagent-render`

```
(fn [a b c]
  [:div])
```

Called each time the component needs to render. It will be passed the arguments you pass it in the vector in Hiccup. Should return more Hiccup.

;; Updating

;; Updates are caused by changes to the arguments (as passed in Hiccup) and changes to Reagent Atoms.

`:component-did-update`

```
(fn [comp [_ & [a b c :as old-args]]]
  (let [node (reagent/dom-node this)]
    (js/console.log "Updating DOM node:" node)))
```

Called just after re-rendering. The reason to use this is to modify the DOM after a render. You can also check the old arguments against the new ones.

;; Unmounting

;; A component is unmounted when it no longer needs to be in the DOM.

`:component-will-unmount`

```
(fn [comp]
  ;; do some cleanup
  (js/console.log "I am about to unmount."))
```

Call just before the component unmounts from the DOM. Use this opportunity to clean up anything this component has created.

`}`

PurelyFunctional.tv