

# TDT Training Curriculum

## JavaScript concepts:

- Array and object functions
- Destructuring Arrays and Objects
- Rest and Spread Operators
- Promises(Async/Await, catch, ...) in JavaScript
- Callback Functions in JavaScript
- Error handling
- lodash library functions ([link](#))
- Design patterns ([example](#))

[Things to learn before react](#)

## While learning React:

Things to focus on:

- What is "state" and "Props"?
- Class and functional components
- Hooks([video](#))
- different component life cycles and how to interact with them
- What are component re-renderings and how do they work?
- how to trigger re-renderings
- Virtual DOM
- High level component
- Error boundary
- Redux
- Testing

[Tips for better code](#)

## Code structure

Things to focus on while Coding

- FileName/FolderName
  - File name should represent the component/class
  - Code file should not be large
  - Split main component into few small components
    - Ex: DropDownComponent -> dropdownList -> Item
  - Move logical functionalities to helper file
    - Ex: Sort/filter array or formatting logic
  - Inline css should be avoided
  - Scss/Css file should have parent class with component-based-name
- Variable/Function name should be readable and understandable

- Ex: showNotification - its a flag to show notification(contains boolean value)
- notificationMessage - its a string
- - Function
- EX: const formattedNotification = formatNotification()
- Variable name should maintain singular/plural status based on value
  - Ex: Notifications - array of notification
  - notificationEntry - its an object/hash
- Variable name should be case sensitive
  - Variable as camelCase (for react)
  - Component name should start with capital letter
- Documentation/comments for code
  - Ex: document input/output structure for formatting function
  - Add comments for extra logic (or special case logic)
- Indentation for coding is importation(follow es-lint structure)
 

```

      <div>
        <span> inside div</span>
      </div>
      
```
- Add **TODO** comment for future implementation/Refactoring
- 

#### Backend:

- Security: Users should not be able to view/edit items that do not belong to them.
- Session: Server Side session. What it is.
- When to use GET and where not to use.
- Server side caching, CDN caching and client side caching
- Migration and Rollback
- Database transaction in SQL