

# DWA\_07.4 Knowledge Check\_DWA7

---

1. Which were the three best abstractions, and why?

- Populating drop down function in data.js

```
/**
 * populates a drop down list
 *
 * @param {HTMLElement} DropDownElement
 * @param {string} DropDownName
 * @param {object} dataObject
 */
const populateDropDown = (DropDownElement, DropDownName, dataObject)
=> {
  const generateHtml = document.createDocumentFragment()
  const firstElement = document.createElement('option')
  firstElement.value = 'any'
  firstElement.innerText = `All ${DropDownName}`
  generateHtml.appendChild(firstElement)

  for (const [id, name] of Object.entries(dataObject)) {
    const element = document.createElement('option')
    element.value = id
    element.innerText = name
    generateHtml.appendChild(element)
  }

  DropDownElement.appendChild(generateHtml)
}
```

I hid all the functionality of how to populate a drop down element in a separate file and just exported it so you can just call it and use it in the imported file.

- HTML object

```
/**
 * An object literal that contains references to all the HTML elements
 * referenced through the operation of the app either upon initialization
or
 * while its running (via event listeners). This ensure that all UI
elements can
 * be accessed and seen in a structured manner in a single data structure.
 */
export const html = {
```

---

## 2. Which were the three worst abstractions, and why?

- HTML object

```
/**
 * An object literal that contains references to all the HTML elements
 * referenced through the operation of the app either upon initialization
or
 * while its running (via event listeners). This ensure that all UI
elements can
 * be accessed and seen in a structured manner in a single data structure.
 */
export const html = {... }
```

- setThemeProperty function

```
/**
 * Function that sets the app theme property
 * @param {string} theme - string theme you want to set
 */
const setThemeProperty = (theme) => {
  if(theme === 'night') {
    document.documentElement.style.setProperty('--color-dark',
html.theme.night.dark);
```

```
        document.documentElement.style.setProperty('--color-light',
html.theme.night.light);
    } else {
        document.documentElement.style.setProperty('--color-dark',
html.theme.day.dark);
        document.documentElement.style.setProperty('--color-light',
html.theme.day.light);
    }
}
```

---

3. How can The three worst abstractions be improved via SOLID principles.

- HTML object:
    - Can be improved by creating separate object objects for each functionality separately which is the main problem with this implementation of abstraction without SOLID principle.
  - setThemeProperty:
    - This function cant be extended without changing the fundamental logic of the function, it violates the open-closed principle.
-