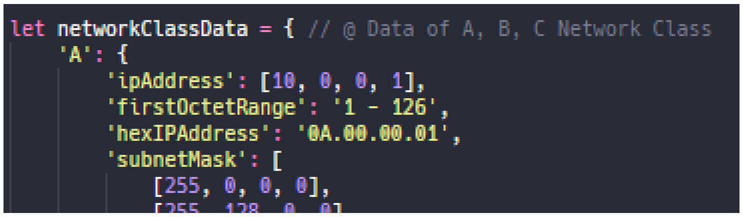
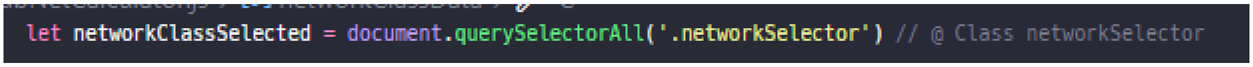
Subnet Calculator



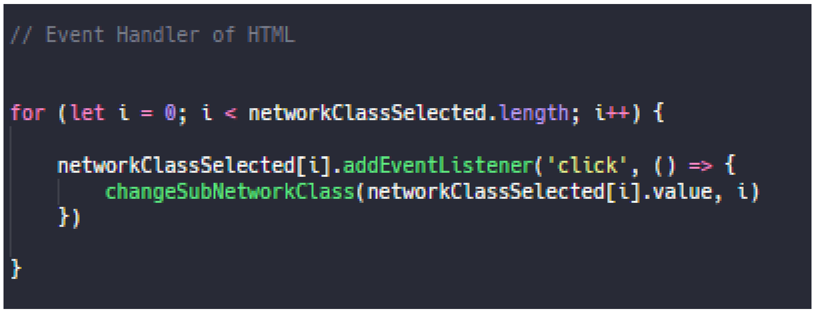
```
let networkClassData = { // @ Data of A, B, C Network Class
    'A': {
        'ipAddress': [10, 0, 0, 1],
        'firstOctetRange': '1 - 126',
        'hexIPAddress': '0A.00.00.01',
        'subnetMask': [
            [255, 0, 0, 0],
            [255, 128, 0, 0]
```

networkClassData : Object data of 3 Class



```
let networkClassSelected = document.querySelectorAll('.networkSelector') // @ Class networkSelector
```

networkClassSelected : Radio input HTML



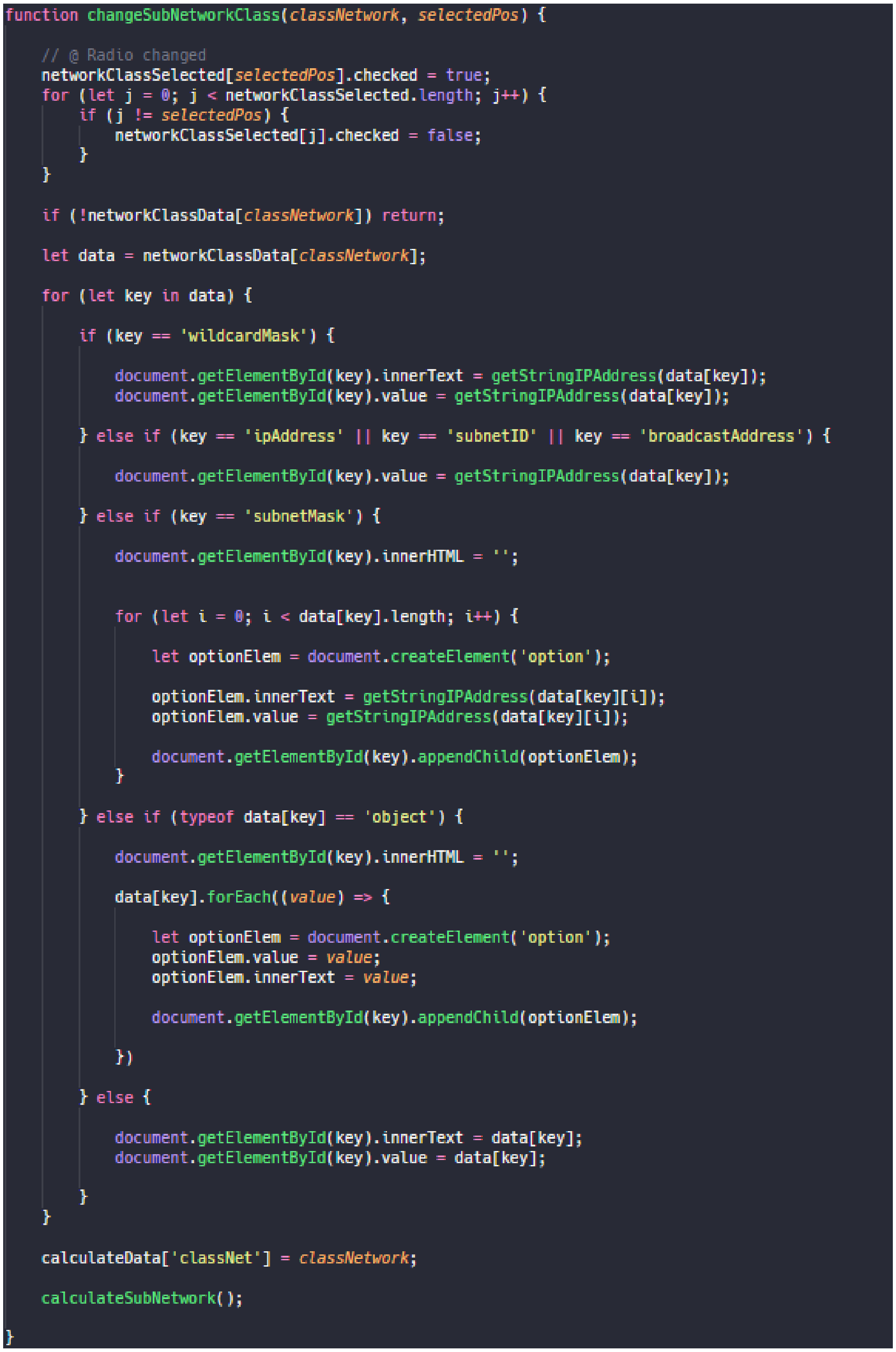
```
// Event Handler of HTML


for (let i = 0; i < networkClassSelected.length; i++) {

    networkClassSelected[i].addEventListener('click', () => {
        changeSubNetworkClass(networkClassSelected[i].value, i)
    })

}
```

.addEventListener to 3 Class Radio input

if select class X then loop will uncheck another class

except the selected class

and call function changeSubNetworkClass(X)



```
function changeSubNetworkClass(classNetwork, selectedPos) {

    // @ Radio changed
    networkClassSelected[selectedPos].checked = true;
    for (let j = 0; j < networkClassSelected.length; j++) {
        if (j != selectedPos) {
            networkClassSelected[j].checked = false;
        }
    }

    if (!networkClassData[classNetwork]) return;

    let data = networkClassData[classNetwork];

    for (let key in data) {

        if (key == 'wildcardMask') {

            document.getElementById(key).innerText = getStringIPAddress(data[key]);
            document.getElementById(key).value = getStringIPAddress(data[key]);

        } else if (key == 'ipAddress' || key == 'subnetID' || key == 'broadcastAddress') {

            document.getElementById(key).value = getStringIPAddress(data[key]);

        } else if (key == 'subnetMask') {

            document.getElementById(key).innerHTML = '';


            for (let i = 0; i < data[key].length; i++) {

                let optionElem = document.createElement('option');

                optionElem.innerText = getStringIPAddress(data[key][i]);
                optionElem.value = getStringIPAddress(data[key][i]);

                document.getElementById(key).appendChild(optionElem);
            }

        } else if (typeof data[key] == 'object') {

            document.getElementById(key).innerHTML = '';

            data[key].forEach((value) => {

                let optionElem = document.createElement('option');
                optionElem.value = value;
                optionElem.innerText = value;

                document.getElementById(key).appendChild(optionElem);

            })

        } else {

            document.getElementById(key).innerText = data[key];
            document.getElementById(key).value = data[key];

        }
    }

    calculateData['classNet'] = classNetwork;

    calculateSubNetwork();

}
```

If networkClassData[X] is null then return

but if it’s not null

We’ll loop for set the interface for X class data