HTML5
드래그 앤 드롭
단위 변환기

김기성

1. 아이디어

- 우리 주변에 다양한 단위가 많아서, 모든 단위를 알기 쉽지 않습니다. 특히 미국 단위
- 저는 앱인벤터처럼 간단히 드래그 앤 드롭으로 계산할 수 있는 방식을 생각해 보았습니다.

기획

- 길이
- 속도
- 각도
- 데이터
- 무게
- 넓이
- 에너지
- 힘
- 압력
- 시간
- 부피

1. 왼쪽에 있는 사이드바를 눌렀을 때 단위 박스 생성

km m mm mile inch

2. 계산 박스 생성 버튼을 눌렀을 때

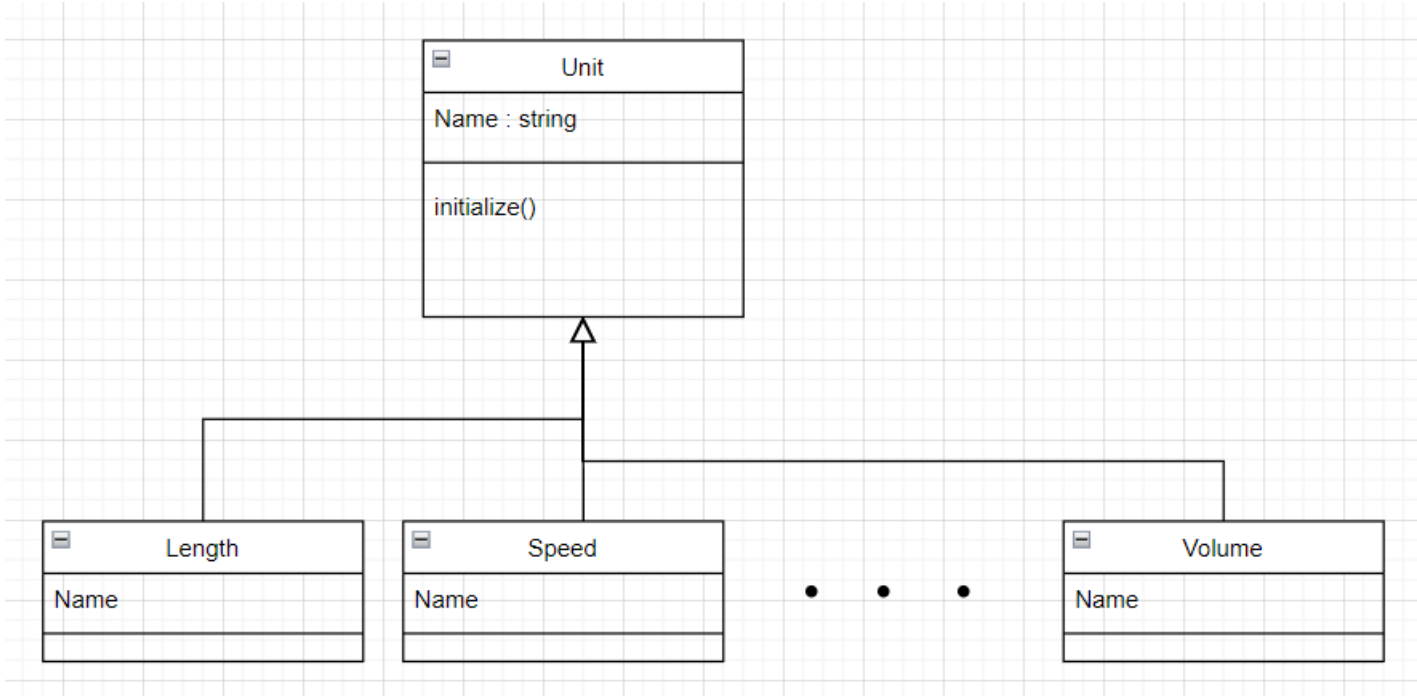
기획

3. 단위박스 객체를 사용할 때 로컬스토리지에 저장

km m mm mile inch

4. => 버튼을 눌러 계산

단위 클래스


Unit
Name : string
initialize()
Length
Name
Speed
Name
Volume
Name

```
class Unit{
    constructor(name){
        this.name = name;
        this.initialize();
    }

    initialize(){
        const unitBox = document.getElementById('unitBox');
        let div = document.createElement("span");
        div.setAttribute('class', 'unitOffcanvas');
        div.innerHTML = this.name;

        unitBox.appendChild(div);
    }
}

class Speed extends Unit{
    constructor(name){
        super(name);
    }
}
class Angle extends Unit{
    constructor(name){
        super(name);
    }
}
```

사이드바를 눌렀을 때 객체 생성

km m mm mile inch

```
const speed = document.getElementById("speed");
const angle = document.getElementById("angle");

speed.addEventListener('click', function () {
    unitBox.innerHTML = "";
    const kmh = new Speed('km/h');
    const mph = new Speed('mile/h');

    placeOnContainer('speed');
})
angle.addEventListener('click', function () {
    unitBox.innerHTML = "";
    const degree = new Angle('도');
    const radian = new Angle('라디안');

    placeOnContainer('angle');
})
```

```
function placeOnContainer(className){
    let elements = document.querySelectorAll('.unitOffcanvas');

    elements.forEach(element => {
        element.addEventListener('click', function () {
            let node = element.cloneNode(true);
            node.setAttribute('class', 'unit');
            node.classList.add(className);
            node.setAttribute('draggable', true);
            node.setAttribute('ondragstart', 'putinBox(event)');
            container.appendChild(node);

            //locatlStorage에 저장
            saveData(node);

        })
    })
}
```

로컬스토리지 사용

```
var offcanvas = document.getElementById('offcanvasWithBothOptions');
offcanvas.addEventListener('show.bs.offcanvas', function(){
    savedData.innerHTML ='';
    let div = document.createElement('div');

    for(var i = 0; i < 3; i++){
        let span = document.createElement("span");
        let data = localStorage.getItem(i);

        if(data === ''){
        }
        else{
            span.setAttribute('class', 'unitOffcanvas');
            span.innerHTML = data;
            div.appendChild(span);
        }
    }
    savedData.appendChild(div);
})
```


```
function makeComponentBox(){
    let componentBox = document.createElement('span');
    let num1 = document.createElement('input');
    let unit1 = document.createElement('div');
    let btnConvert = document.createElement('button');
    let num2 = document.createElement('input');
    let unit2 = document.createElement('div');

    componentBox.setAttribute('class', 'componentBox');
    componentBox.setAttribute('draggable', 'true');
    componentBox.setAttribute('ondragstart', 'dragNdrop(event)');
    num1.setAttribute('class', 'num1');
    unit1.setAttribute('class', 'unit1');
    btnConvert.setAttribute('class', 'btnConvert');
    btnConvert.setAttribute('onclick', 'btnConvert(event)');
    btnConvert.innerHTML = '=>';
    num2.setAttribute('class', 'num2');
    unit2.setAttribute('class', 'unit2');

    componentBox.appendChild(num1);
    componentBox.appendChild(unit1);
    componentBox.appendChild(btnConvert);
    componentBox.appendChild(num2);
    componentBox.appendChild(unit2);

    document.getElementById('container').innerHTML += componentBox.outerHTML;
}
```

드래그 앤 드롭 구현

```
//dragstart
function putinBox(e){
    e.target.classList.add('dragging');

    //dragend
    e.currentTarget.addEventListener('dragend', () => {
        e.target.classList.remove('dragging');
    })

    let unitContainers1 = Array.from(document.querySelectorAll('.unit1'));
    let unitContainers2 = Array.from(document.querySelectorAll('.unit2'));
    let unitContainers = unitContainers1.concat(unitContainers2);

    //dragover
    unitContainers.forEach(unitContainer => {
        unitContainer.addEventListener('dragover', e => {
            e.preventDefault();
            const draggable = document.querySelector('.dragging');
            draggable.style.position = 'relative';

            if(unitContainer.innerHTML.length === 0)
            {
                unitContainer.style.width = draggable.offsetWidth + 'px';
                unitContainer.appendChild(draggable);
            }
        });
    });
}
```

계산 로직

```
function btnConvert(e) {
    let parentComponent = e.currentTarget.parentElement;
    let unit1 = parentComponent.children[1].firstChild;
    let unit2 = parentComponent.children[4].firstChild;
    let num1 = e.currentTarget.previousElementSibling.previousElementSibling.value;
    let num2 = e.currentTarget.nextElementSibling.value;

    if (unit1.className === unit2.className) {

        //speed, angle, length, data, weight, area, energy, force, pressure, time, torque, volume
        switch (unit1.className) {
            case 'unit length':
                num2 = length2(unit2.innerHTML, length1(unit1.innerHTML, num1));
                e.currentTarget.nextElementSibling.value = num2;
                break;
            case 'unit speed':
                num2 = speed2(unit2.innerHTML, speed1(unit1.innerHTML, num1));
                e.currentTarget.nextElementSibling.value = num2;
                break;
```

```
function length1(unit, num) {
        let temp;
        switch (unit) {
            case 'm':
                temp = num;
                break;
            case 'km':
                temp = num * 1000;
                break;
            case 'mm':
                temp = num * 0.001;
                break;
            case 'mile':
                temp = num * 1609.34;
                break;
            case 'inch':
                temp = num * 0.0254;
                break;
            case 'ft':
                temp = num * 3.2808;
                break;
        }
        return temp;
    }
```

아쉬운 점

- 드래그 앤 드롭을 해서 계산 박스에 넣는 기능과, 특정 위치에 배치하는 기능을 같이 넣으려고 했는데 로직이 충돌해서 그런지 같이 적용이 되지 않았다.

이상 발표를 마치겠습니다.