휴먼컴퓨터인터페이스

과제#2 WebUI 계산기 구현

2021-05-15

소프트웨어학부

2016726072

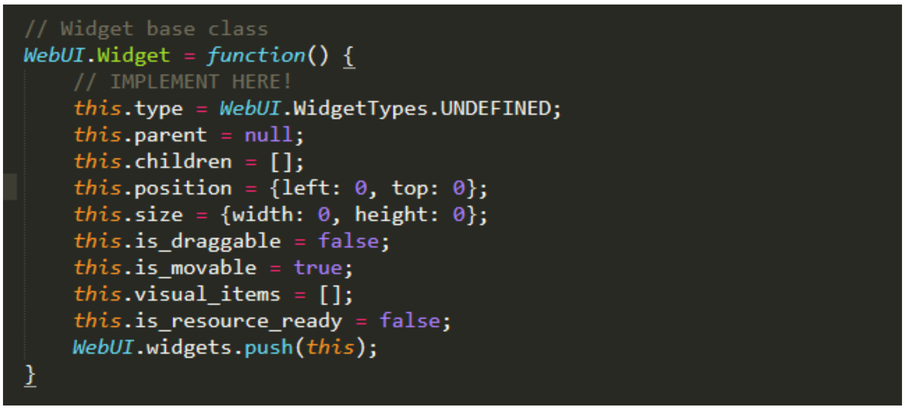
조현민

요약표

0단계	1단계	2단계	3단계
위젯 기능 구현o	레이아웃 기능 구현o	CalcButton 클래스o	GridView생성o
로그인 페이지 구성o	로그인 페이지 재구성o	initWidgets() 함수 재정의o	calcSave생성o

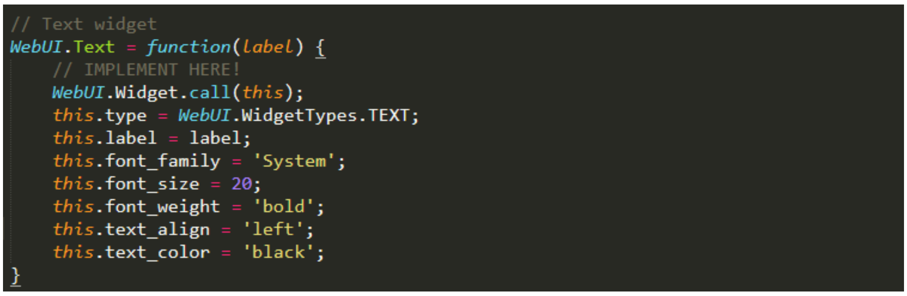
1. 0단계

Widget 구현(강의자료)



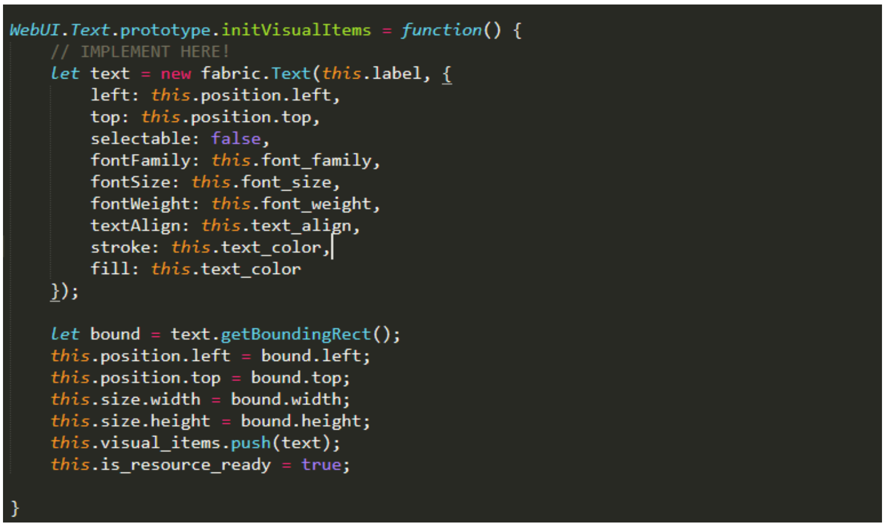
```
// Widget base class
WebUI.Widget = function() {
    // IMPLEMENT HERE!
    this.type = WebUI.WidgetTypes.UNDEFINED;
    this.parent = null;
    this.children = [];
    this.position = {left: 0, top: 0};
    this.size = {width: 0, height: 0};
    this.is_draggable = false;
    this.is_movable = true;
    this.visual_items = [];
    this.is_resource_ready = false;
    WebUI.widgets.push(this);
}
```

Text 구현(강의자료)



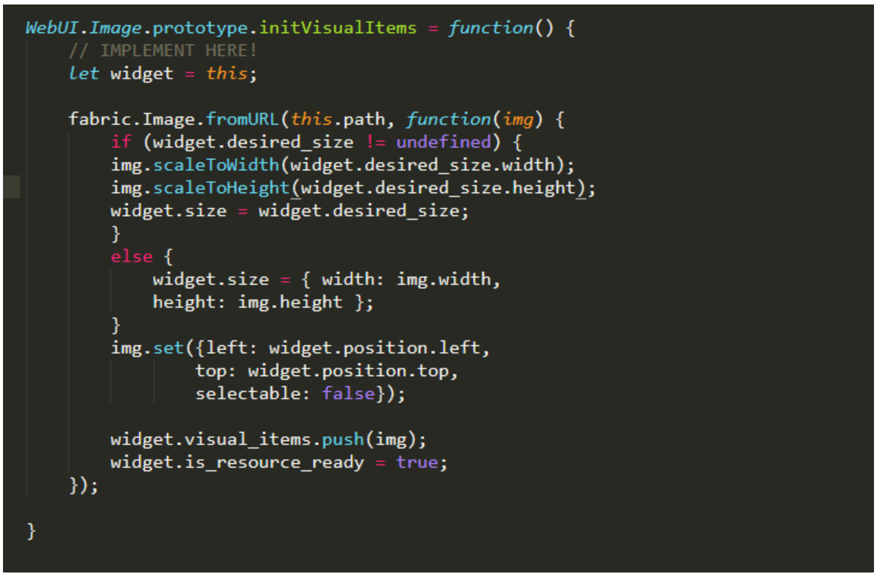
```
// Text widget
WebUI.Text = function(label) {
    // IMPLEMENT HERE!
    WebUI.Widget.call(this);
    this.type = WebUI.WidgetTypes.TEXT;
    this.label = label;
    this.font_family = 'System';
    this.font_size = 20;
    this.font_weight = 'bold';
    this.text_align = 'left';
    this.text_color = 'black';
}
```

Text initVisualItems 구현(강의자료)



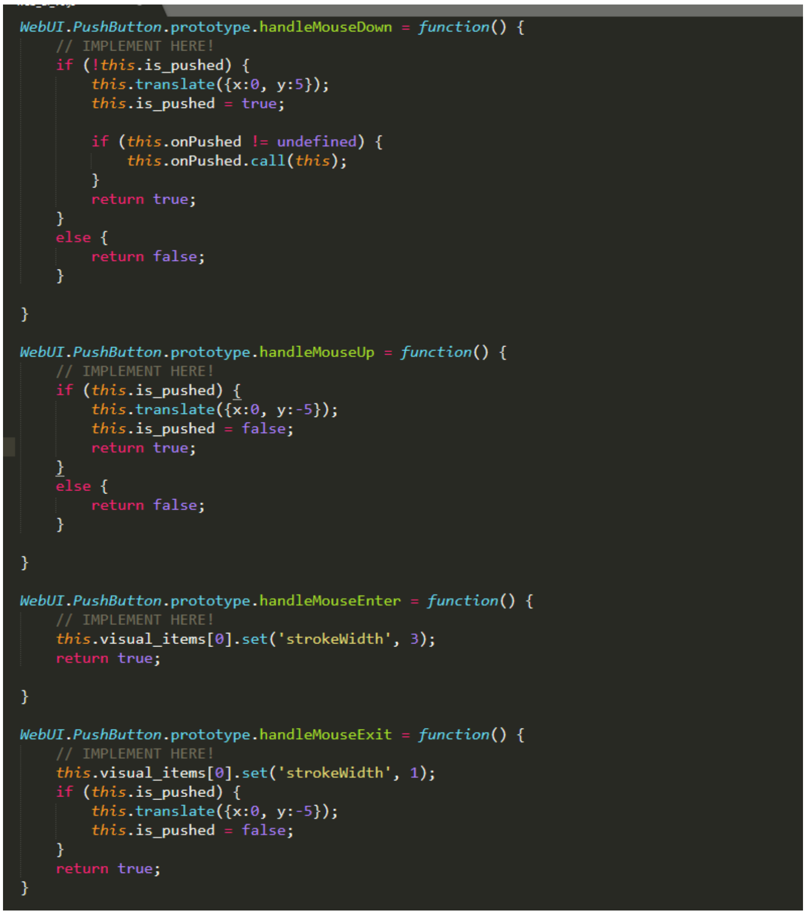
```
WebUI.Text.prototype.initVisualItems = function() {
    // IMPLEMENT HERE!
    let text = new fabric.Text(this.label, {
        left: this.position.left,
        top: this.position.top,
        selectable: false,
        fontFamily: this.font_family,
        fontSize: this.font_size,
        fontWeight: this.font_weight,
        textAlign: this.text_align,
        stroke: this.text_color,
        fill: this.text_color
    });

    let bound = text.getBoundingRect();
    this.position.left = bound.left;
    this.position.top = bound.top;
    this.size.width = bound.width;
    this.size.height = bound.height;
    this.visual_items.push(text);
    this.is_resource_ready = true;

}
```

Image initVisualItems 구현(강의자료)



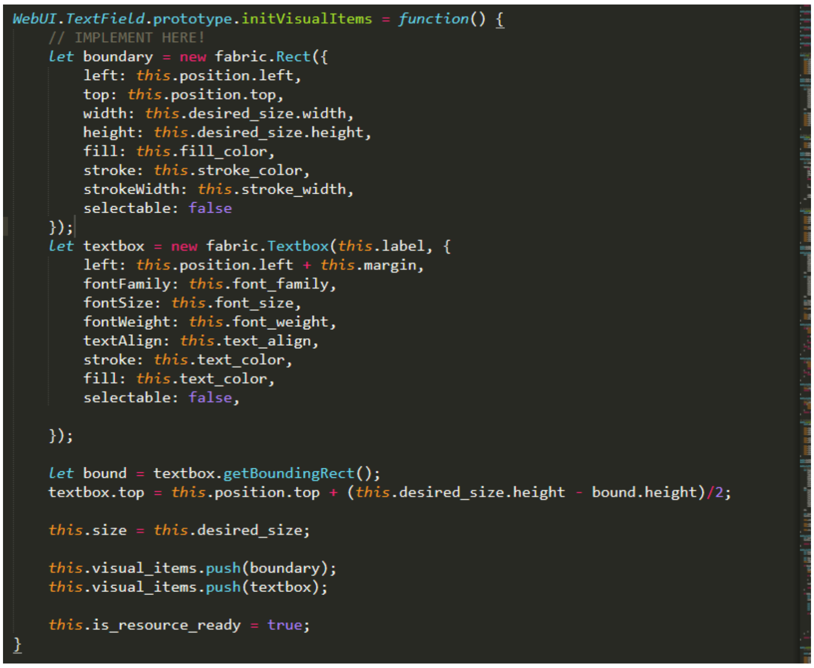
```
WebUI.Image.prototype.initVisualItems = function() {
    // IMPLEMENT HERE!
    let widget = this;

    fabric.Image.fromURL(this.path, function(img) {
        if (widget.desired_size != undefined) {
        img.scaleToWidth(widget.desired_size.width);
        img.scaleToHeight(widget.desired_size.height);
        widget.size = widget.desired_size;
        }
        else {
            widget.size = { width: img.width,
            height: img.height };
        }
        img.set({left: widget.position.left,
                top: widget.position.top,
                selectable: false});

        widget.visual_items.push(img);
        widget.is_resource_ready = true;
    });

}
```

PushButton 마우스조작 구현(강의자료)



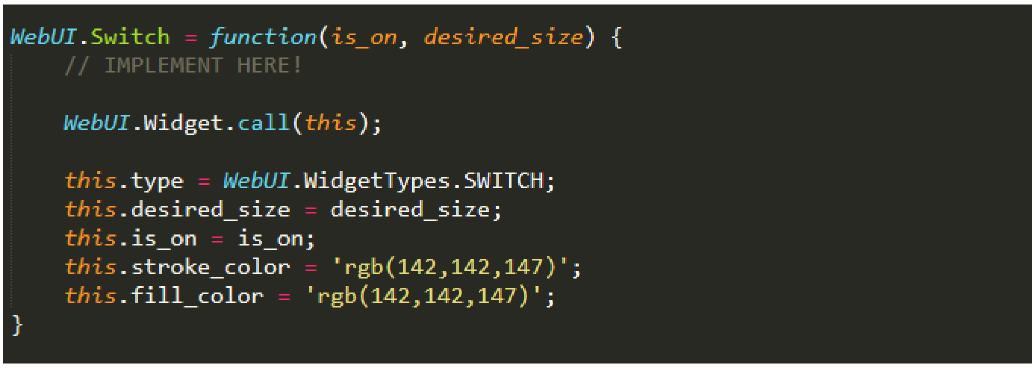
```
WebUI.PushButton.prototype.handleMouseDown = function() {
    // IMPLEMENT HERE!
    if (!this.is_pushed) {
        this.translate({x:0, y:5});
        this.is_pushed = true;

        if (this.onPushed != undefined) {
            this.onPushed.call(this);
        }
        return true;
    }
    else {
        return false;
    }

}

WebUI.PushButton.prototype.handleMouseUp = function() {
    // IMPLEMENT HERE!
    if (this.is_pushed) {
        this.translate({x:0, y:-5});
        this.is_pushed = false;
        return true;
    }
    else {
        return false;
    }

}

WebUI.PushButton.prototype.handleMouseEnter = function() {
    // IMPLEMENT HERE!
    this.visual_items[0].set('strokeWidth', 3);
    return true;

}

WebUI.PushButton.prototype.handleMouseExit = function() {
    // IMPLEMENT HERE!
    this.visual_items[0].set('strokeWidth', 1);
    if (this.is_pushed) {
        this.translate({x:0, y:-5});
        this.is_pushed = false;
    }
    return true;
}
```

TextField initVisualItems 구현(강의자료)



```
WebUI.TextField.prototype.initVisualItems = function() {
    // IMPLEMENT HERE!
    let boundary = new fabric.Rect({
        left: this.position.left,
        top: this.position.top,
        width: this.desired_size.width,
        height: this.desired_size.height,
        fill: this.fill_color,
        stroke: this.stroke_color,
        strokeWidth: this.stroke_width,
        selectable: false
    });
    let textbox = new fabric.Textbox(this.label, {
        left: this.position.left + this.margin,
        fontFamily: this.font_family,
        fontSize: this.font_size,
        fontWeight: this.font_weight,
        textAlign: this.text_align,
        stroke: this.text_color,
        fill: this.text_color,
        selectable: false,

    });

    let bound = textbox.getBoundingRect();
    textbox.top = this.position.top + (this.desired_size.height - bound.height)/2;

    this.size = this.desired_size;

    this.visual_items.push(boundary);
    this.visual_items.push(textbox);

    this.is_resource_ready = true;
}
```

Switch 정의

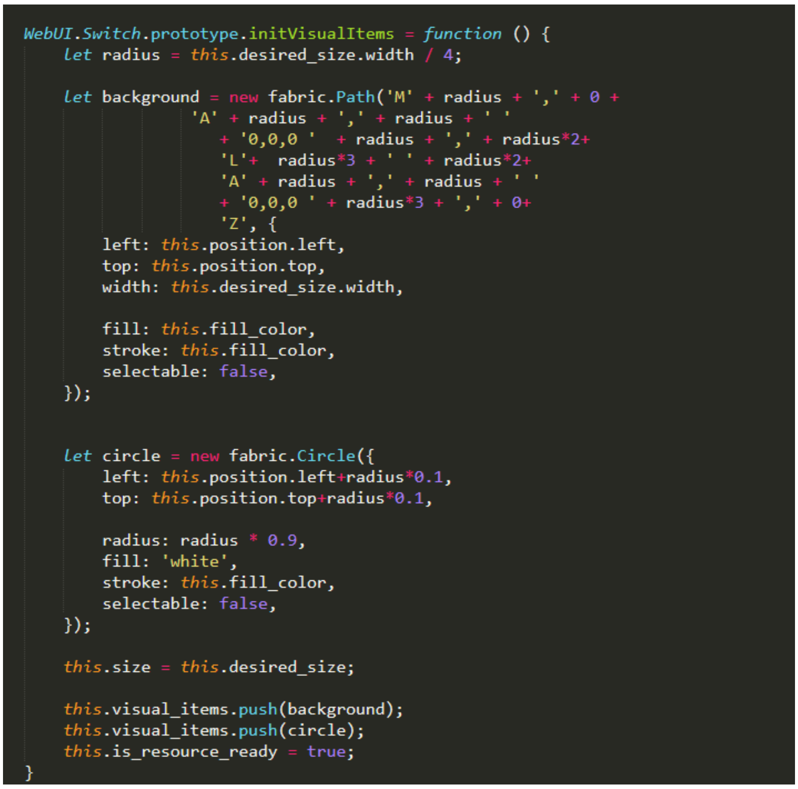


```
WebUI.Switch = function(is_on, desired_size) {
    // IMPLEMENT HERE!

    WebUI.Widget.call(this);

    this.type = WebUI.WidgetTypes.SWITCH;
    this.desired_size = desired_size;
    this.is_on = is_on;
    this.stroke_color = 'rgb(142,142,147)';
    this.fill_color = 'rgb(142,142,147)';
}
```

Switch initVisualItems 구현

스위치의 배경을 그리기 위해 fabric.Path를 이용하였다(사이트 참고).

스위치의 원은 배경보다 작게 제작하였다.

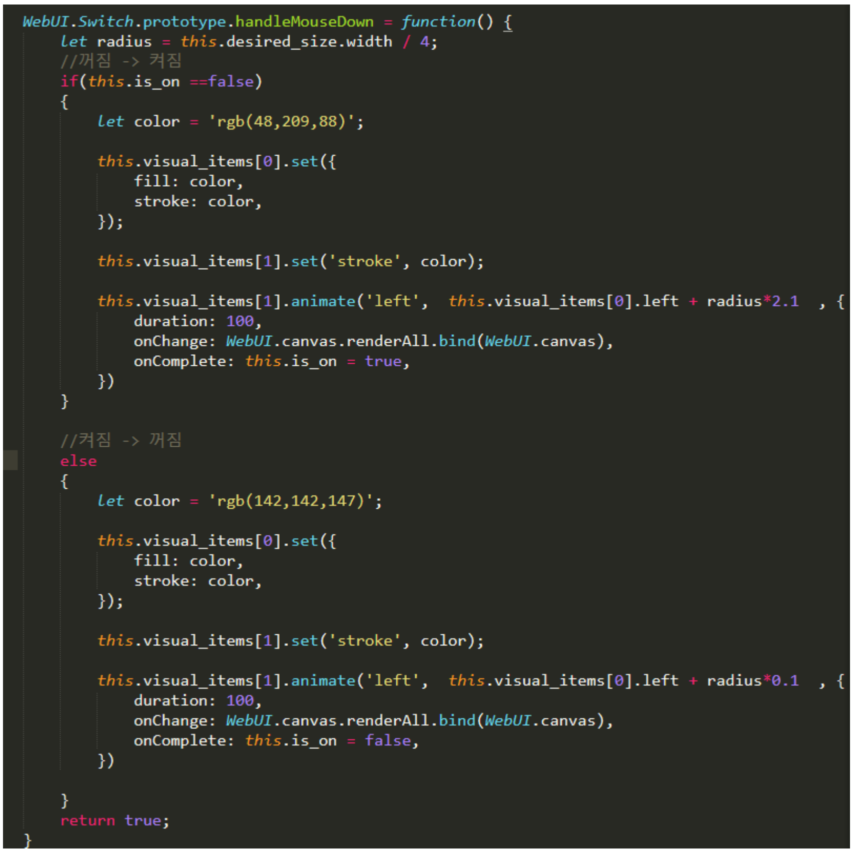


```
WebUI.Switch.prototype.initVisualItems = function () {
    let radius = this.desired_size.width / 4;

    let background = new fabric.Path('M' + radius + ',' + 0 +
                 'A' + radius + ',' + radius + ' '
                    + '0,0,0 '  + radius + ',' + radius*2+
                    'L'+  radius*3 + ' ' + radius*2+
                    'A' + radius + ',' + radius + ' '
                    + '0,0,0 ' + radius*3 + ',' + 0+
                    'Z', {
        left: this.position.left,
        top: this.position.top,
        width: this.desired_size.width,

        fill: this.fill_color,
        stroke: this.fill_color,
        selectable: false,
    });


    let circle = new fabric.Circle({
        left: this.position.left+radius*0.1,
        top: this.position.top+radius*0.1,

        radius: radius * 0.9,
        fill: 'white',
        stroke: this.fill_color,
        selectable: false,
    });

    this.size = this.desired_size;

    this.visual_items.push(background);
    this.visual_items.push(circle);
    this.is_resource_ready = true;
}
```

Switch 버튼 클릭 구현

켜져 있을 때와 꺼져있을 때, 각각에 맞는 작동을 구현하였다.

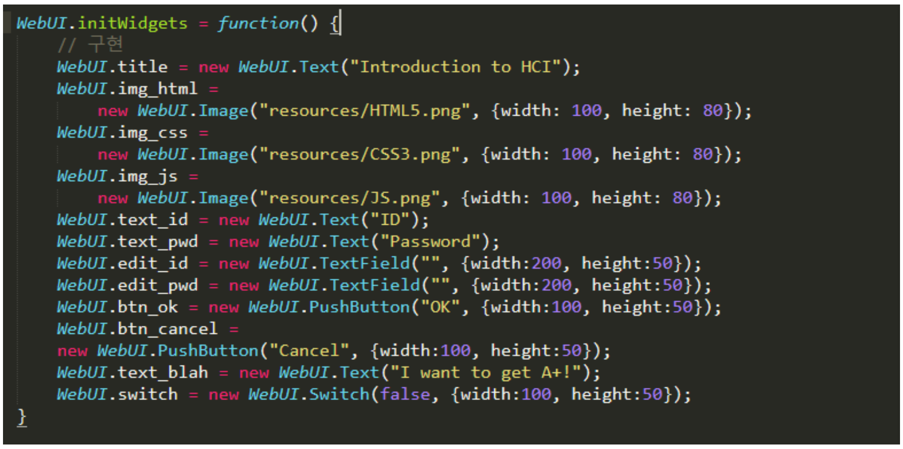
animate를 사용하여 부드럽게 움직이도록 구현하였다.



```
WebUI.Switch.prototype.handleMouseDown = function() {
    let radius = this.desired_size.width / 4;
    //꺼짐 -> 켜짐
    if(this.is_on ==false)
    {
        let color = 'rgb(48,209,88)';

        this.visual_items[0].set({
            fill: color,
            stroke: color,
        });

        this.visual_items[1].set('stroke', color);

        this.visual_items[1].animate('left',  this.visual_items[0].left + radius*2.1   , {
            duration: 100,
            onChange: WebUI.canvas.renderAll.bind(WebUI.canvas),
            onComplete: this.is_on = true,
        })
    }

    //켜짐 -> 꺼짐
    else
    {
        let color = 'rgb(142,142,147)';

        this.visual_items[0].set({
            fill: color,
            stroke: color,
        });

        this.visual_items[1].set('stroke', color);

        this.visual_items[1].animate('left',  this.visual_items[0].left + radius*0.1   , {
            duration: 100,
            onChange: WebUI.canvas.renderAll.bind(WebUI.canvas),
            onComplete: this.is_on = false,
        })

    }
    return true;
}
```

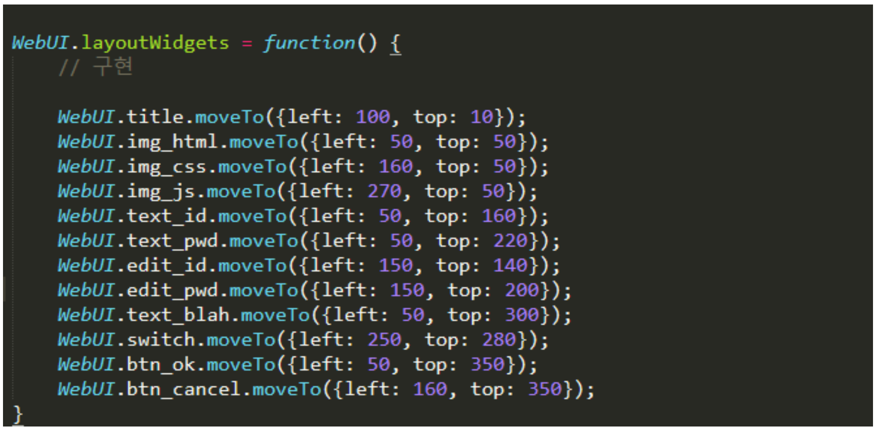
로그인 페이지 구성

initWidgets 구현(강의자료)



```
WebUI.initWidgets = function() {
    // 구현
    WebUI.title = new WebUI.Text("Introduction to HCI");
    WebUI.img_html =
        new WebUI.Image("resources/HTML5.png", {width: 100, height: 80});
    WebUI.img_css =
        new WebUI.Image("resources/CSS3.png", {width: 100, height: 80});
    WebUI.img_js =
        new WebUI.Image("resources/JS.png", {width: 100, height: 80});
    WebUI.text_id = new WebUI.Text("ID");
    WebUI.text_pwd = new WebUI.Text("Password");
    WebUI.edit_id = new WebUI.TextField("", {width:200, height:50});
    WebUI.edit_pwd = new WebUI.TextField("", {width:200, height:50});
    WebUI.btn_ok = new WebUI.PushButton("OK", {width:100, height:50});
    WebUI.btn_cancel =
    new WebUI.PushButton("Cancel", {width:100, height:50});
    WebUI.text_blah = new WebUI.Text("I want to get A+!");
    WebUI.switch = new WebUI.Switch(false, {width:100, height:50});
}
```

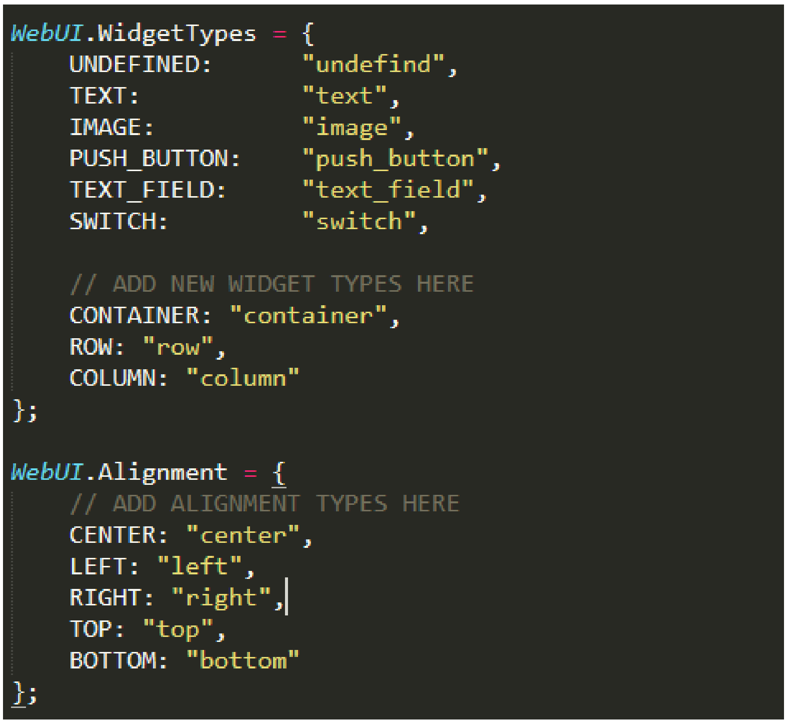
layoutWidgets 구현(강의자료)



```
WebUI.layoutWidgets = function() {
    // 구현

    WebUI.title.moveTo({left: 100, top: 10});
    WebUI.img_html.moveTo({left: 50, top: 50});
    WebUI.img_css.moveTo({left: 160, top: 50});
    WebUI.img_js.moveTo({left: 270, top: 50});
    WebUI.text_id.moveTo({left: 50, top: 160});
    WebUI.text_pwd.moveTo({left: 50, top: 220});
    WebUI.edit_id.moveTo({left: 150, top: 140});
    WebUI.edit_pwd.moveTo({left: 150, top: 200});
    WebUI.text_blah.moveTo({left: 50, top: 300});
    WebUI.switch.moveTo({left: 250, top: 280});
    WebUI.btn_ok.moveTo({left: 50, top: 350});
    WebUI.btn_cancel.moveTo({left: 160, top: 350});
}
```

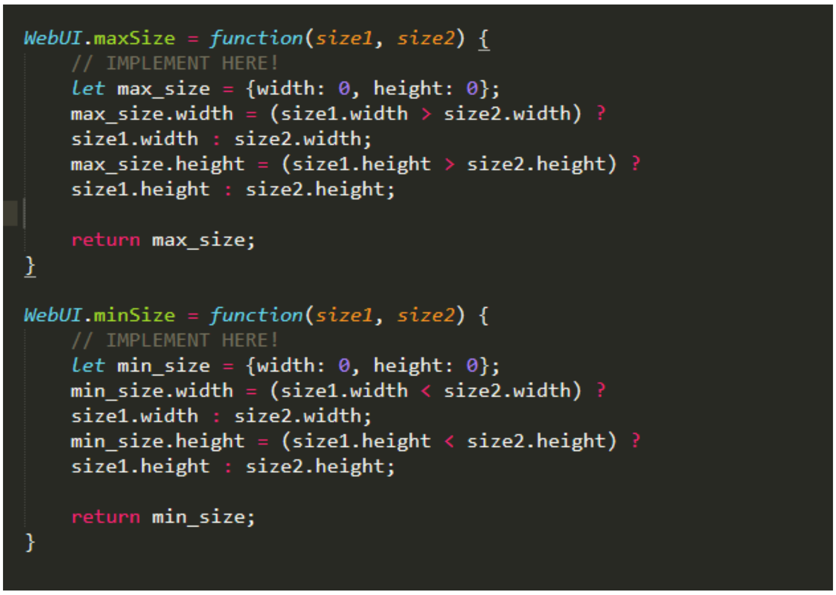
2. 1단계

WidgetTypes/Alignment 구현(강의자료)



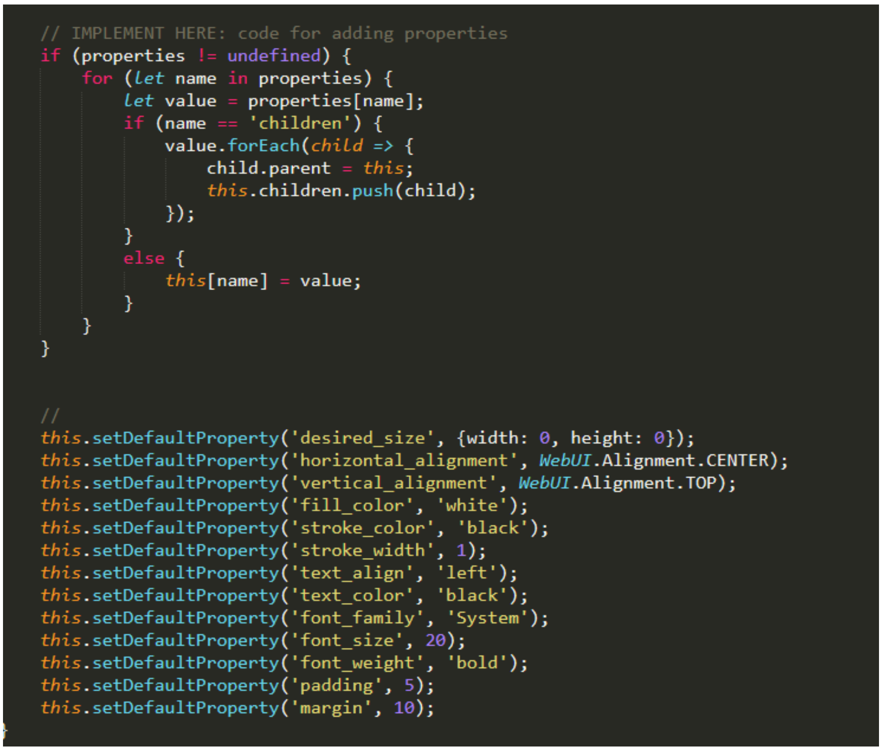
```
WebUI.WidgetTypes = {
    UNDEFINED:      "undefind",
    TEXT:           "text",
    IMAGE:          "image",
    PUSH_BUTTON:    "push_button",
    TEXT_FIELD:     "text_field",
    SWITCH:         "switch",

    // ADD NEW WIDGET TYPES HERE
    CONTAINER: "container",
    ROW: "row",
    COLUMN: "column"
};

WebUI.Alignment = {
    // ADD ALIGNMENT TYPES HERE
    CENTER: "center",
    LEFT: "left",
    RIGHT: "right",
    TOP: "top",
    BOTTOM: "bottom"
};
```

maxSize/minSize 구현(강의자료)



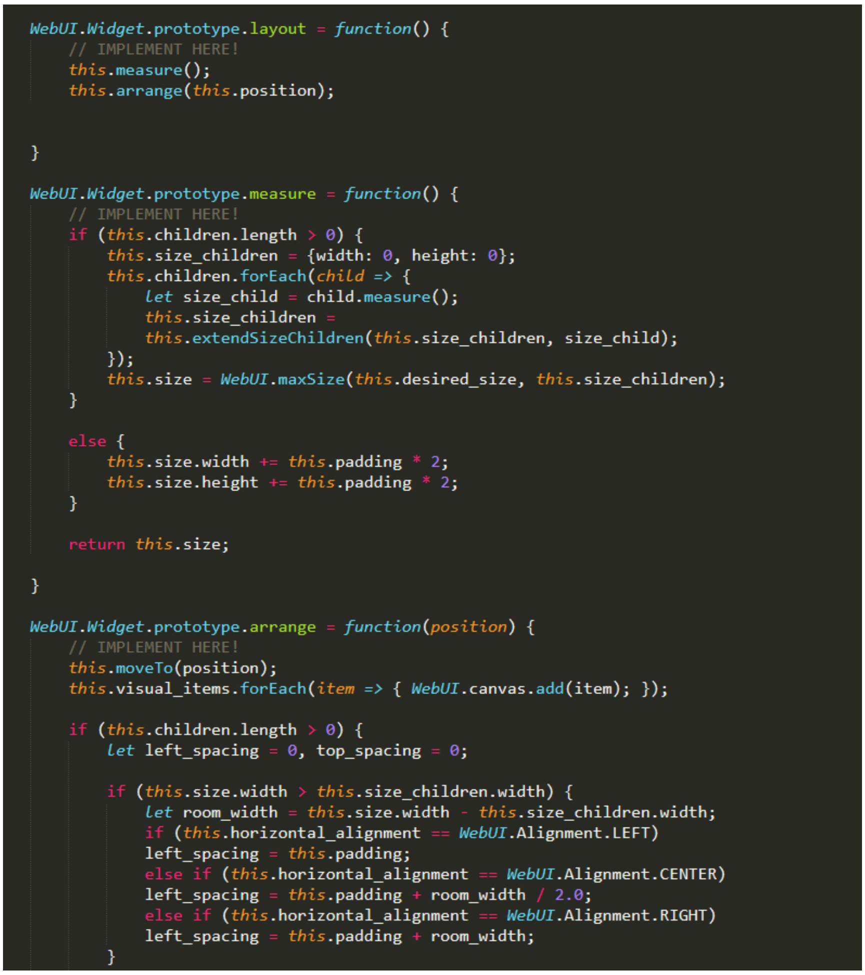
```
WebUI.maxSize = function(size1, size2) {
    // IMPLEMENT HERE!
    let max_size = {width: 0, height: 0};
    max_size.width = (size1.width > size2.width) ?
    size1.width : size2.width;
    max_size.height = (size1.height > size2.height) ?
    size1.height : size2.height;

    return max_size;
}

WebUI.minSize = function(size1, size2) {
    // IMPLEMENT HERE!
    let min_size = {width: 0, height: 0};
    min_size.width = (size1.width < size2.width) ?
    size1.width : size2.width;
    min_size.height = (size1.height < size2.height) ?
    size1.height : size2.height;

    return min_size;
}
```

Widget 구현(강의자료)



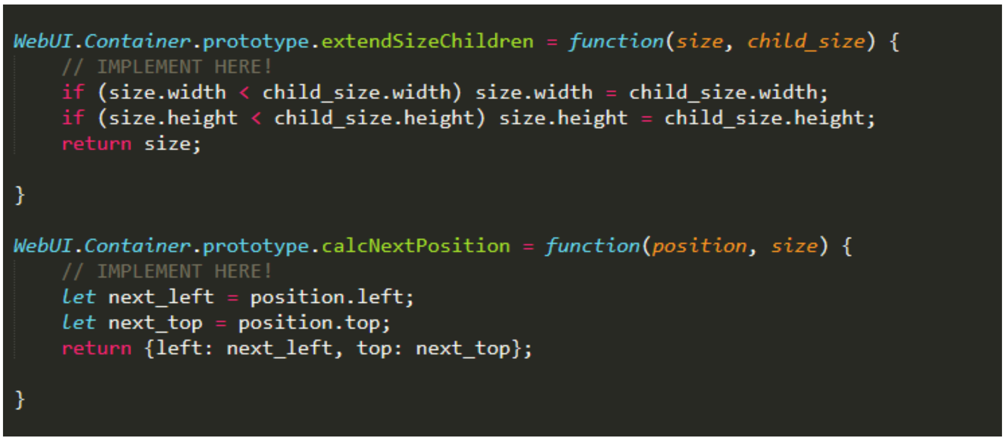
```
    // IMPLEMENT HERE: code for adding properties
    if (properties != undefined) {
        for (let name in properties) {
            let value = properties[name];
            if (name == 'children') {
                value.forEach(child => {
                    child.parent = this;
                    this.children.push(child);
                });
            }
            else {
                this[name] = value;
            }
        }
    }


    //
    this.setDefaultProperty('desired_size', {width: 0, height: 0});
    this.setDefaultProperty('horizontal_alignment', WebUI.Alignment.CENTER);
    this.setDefaultProperty('vertical_alignment', WebUI.Alignment.TOP);
    this.setDefaultProperty('fill_color', 'white');
    this.setDefaultProperty('stroke_color', 'black');
    this.setDefaultProperty('stroke_width', 1);
    this.setDefaultProperty('text_align', 'left');
    this.setDefaultProperty('text_color', 'black');
    this.setDefaultProperty('font_family', 'System');
    this.setDefaultProperty('font_size', 20);
    this.setDefaultProperty('font_weight', 'bold');
    this.setDefaultProperty('padding', 5);
    this.setDefaultProperty('margin', 10);
}
```

Layout/measure/arrange 구현(강의자료)



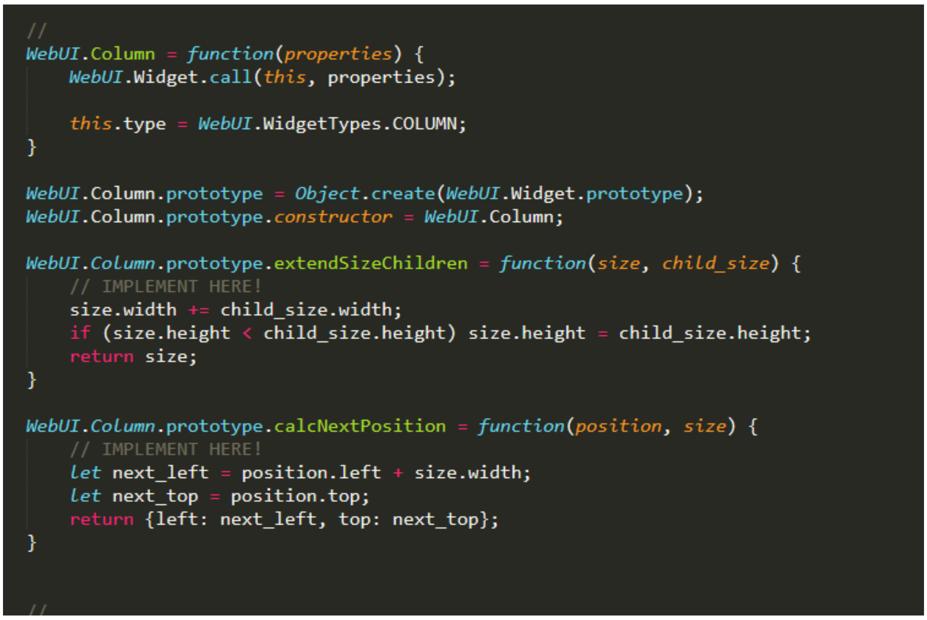
```
WebUI.Widget.prototype.layout = function() {
    // IMPLEMENT HERE!
    this.measure();
    this.arrange(this.position);


}

WebUI.Widget.prototype.measure = function() {
    // IMPLEMENT HERE!
    if (this.children.length > 0) {
        this.size_children = {width: 0, height: 0};
        this.children.forEach(child => {
            let size_child = child.measure();
            this.size_children =
            this.extendSizeChildren(this.size_children, size_child);
        });
        this.size = WebUI.maxSize(this.desired_size, this.size_children);
    }

    else {
        this.size.width += this.padding * 2;
        this.size.height += this.padding * 2;
    }

    return this.size;

}

WebUI.Widget.prototype.arrange = function(position) {
    // IMPLEMENT HERE!
    this.moveTo(position);
    this.visual_items.forEach(item => { WebUI.canvas.add(item); });

    if (this.children.length > 0) {
        let left_spacing = 0, top_spacing = 0;

        if (this.size.width > this.size_children.width) {
            let room_width = this.size.width - this.size_children.width;
            if (this.horizontal_alignment == WebUI.Alignment.LEFT)
            left_spacing = this.padding;
            else if (this.horizontal_alignment == WebUI.Alignment.CENTER)
            left_spacing = this.padding + room_width / 2.0;
            else if (this.horizontal_alignment == WebUI.Alignment.RIGHT)
            left_spacing = this.padding + room_width;
        }
```

Container 속성 구현(강의자료)



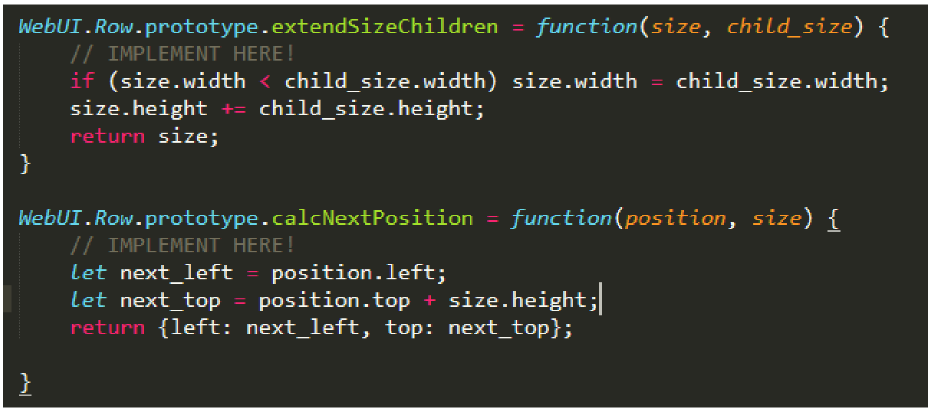
```
WebUI.Container.prototype.extendSizeChildren = function(size, child_size) {
    // IMPLEMENT HERE!
    if (size.width < child_size.width) size.width = child_size.width;
    if (size.height < child_size.height) size.height = child_size.height;
    return size;

}

WebUI.Container.prototype.calcNextPosition = function(position, size) {
    // IMPLEMENT HERE!
    let next_left = position.left;
    let next_top = position.top;
    return {left: next_left, top: next_top};

}
```

Column 속성 구현(강의자료)



```
//
WebUI.Column = function(properties) {
    WebUI.Widget.call(this, properties);

    this.type = WebUI.WidgetTypes.COLUMN;
}

WebUI.Column.prototype = Object.create(WebUI.Widget.prototype);
WebUI.Column.prototype.constructor = WebUI.Column;

WebUI.Column.prototype.extendSizeChildren = function(size, child_size) {
    // IMPLEMENT HERE!
    size.width += child_size.width;
    if (size.height < child_size.height) size.height = child_size.height;
    return size;
}

WebUI.Column.prototype.calcNextPosition = function(position, size) {
    // IMPLEMENT HERE!
    let next_left = position.left + size.width;
    let next_top = position.top;
    return {left: next_left, top: next_top};
}


//
```

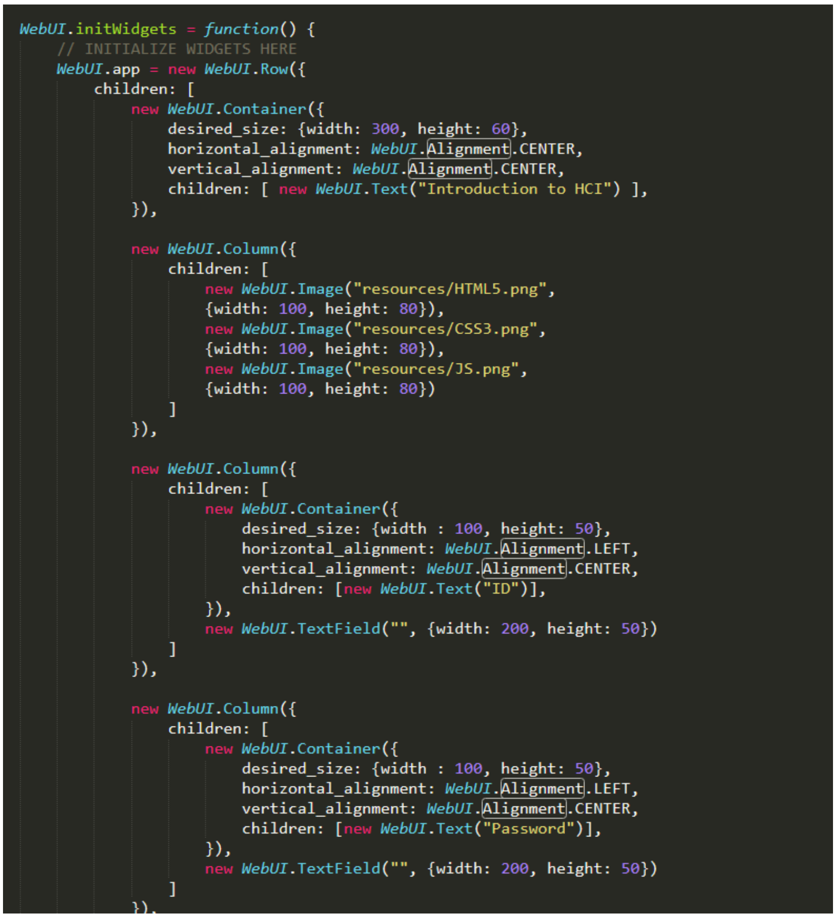
Row 속성 구현(강의자료)



```
WebUI.Row.prototype.extendSizeChildren = function(size, child_size) {
    // IMPLEMENT HERE!
    if (size.width < child_size.width) size.width = child_size.width;
    size.height += child_size.height;
    return size;
}

WebUI.Row.prototype.calcNextPosition = function(position, size) {
    // IMPLEMENT HERE!
    let next_left = position.left;
    let next_top = position.top + size.height;
    return {left: next_left, top: next_top};

}
```

initWidgets 구현

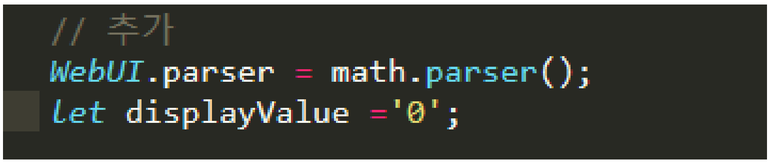
Row/Column/Container를 이용하여 initWidgets을 구현하였다.



```
WebUI.initWidgets = function() {
    // INITIALIZE WIDGETS HERE
    WebUI.app = new WebUI.Row({
        children: [
            new WebUI.Container({
                desired_size: {width: 300, height: 60},
                horizontal_alignment: WebUI.Alignment.CENTER,
                vertical_alignment: WebUI.Alignment.CENTER,
                children: [ new WebUI.Text("Introduction to HCI") ],
            }),

            new WebUI.Column({
                children: [
                    new WebUI.Image("resources/HTML5.png",
                    {width: 100, height: 80}),
                    new WebUI.Image("resources/CSS3.png",
                    {width: 100, height: 80}),
                    new WebUI.Image("resources/JS.png",
                    {width: 100, height: 80})
                ]
            }),

            new WebUI.Column({
                children: [
                    new WebUI.Container({
                        desired_size: {width : 100, height: 50},
                        horizontal_alignment: WebUI.Alignment.LEFT,
                        vertical_alignment: WebUI.Alignment.CENTER,
                        children: [new WebUI.Text("ID")],
                    }),
                    new WebUI.TextField("", {width: 200, height: 50})
                ]
            }),

            new WebUI.Column({
                children: [
                    new WebUI.Container({
                        desired_size: {width : 100, height: 50},
                        horizontal_alignment: WebUI.Alignment.LEFT,
                        vertical_alignment: WebUI.Alignment.CENTER,
                        children: [new WebUI.Text("Password")],
                    }),
                    new WebUI.TextField("", {width: 200, height: 50})
                ]
            }),
```

3. 2단계

수식 계산을 위한 객체 정의

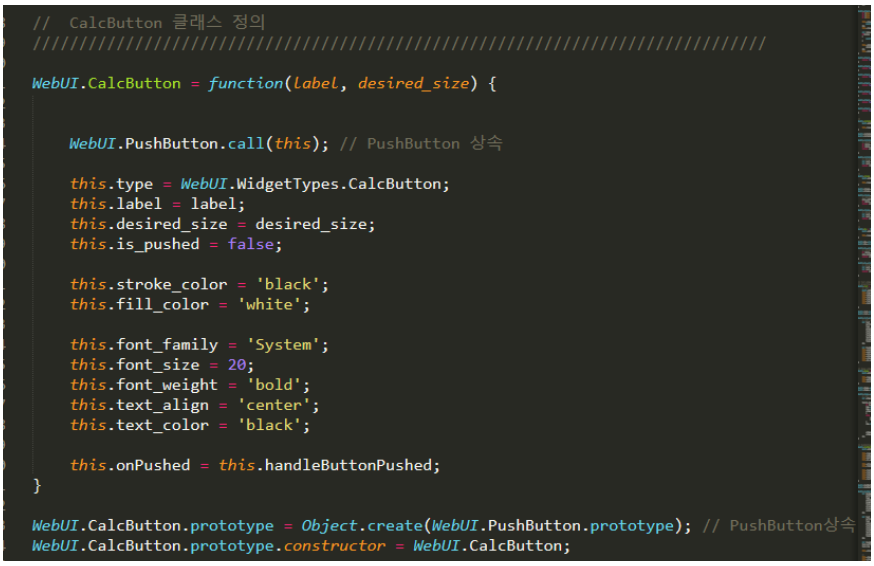


```
// 추가
WebUI.parser = math.parser();
let displayValue ='0';
```

CalcButton 클래스 정의

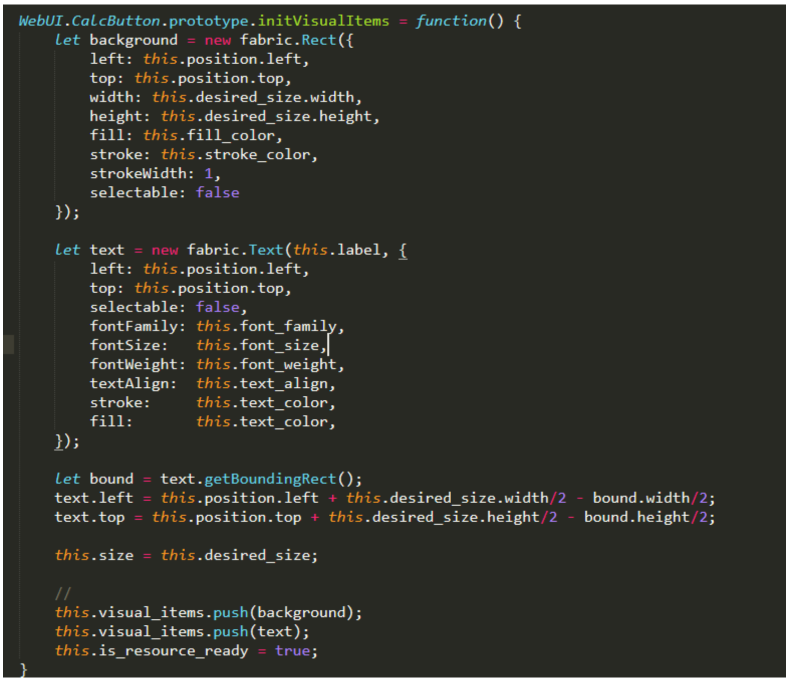
this.onPushed = this.handleButtonPushed 추가

PushButton 상속



```
//  CalcButton 클래스 정의
///////////////////////////////////////////////////////////////////////////////

WebUI.CalcButton = function(label, desired_size) {


    WebUI.PushButton.call(this); // PushButton 상속

    this.type = WebUI.WidgetTypes.CalcButton;
    this.label = label;
    this.desired_size = desired_size;
    this.is_pushed = false;

    this.stroke_color = 'black';
    this.fill_color = 'white';

    this.font_family = 'System';
    this.font_size = 20;
    this.font_weight = 'bold';
    this.text_align = 'center';
    this.text_color = 'black';

    this.onPushed = this.handleButtonPushed;
}

WebUI.CalcButton.prototype = Object.create(WebUI.PushButton.prototype); // PushButton상속
WebUI.CalcButton.prototype.constructor = WebUI.CalcButton;
```

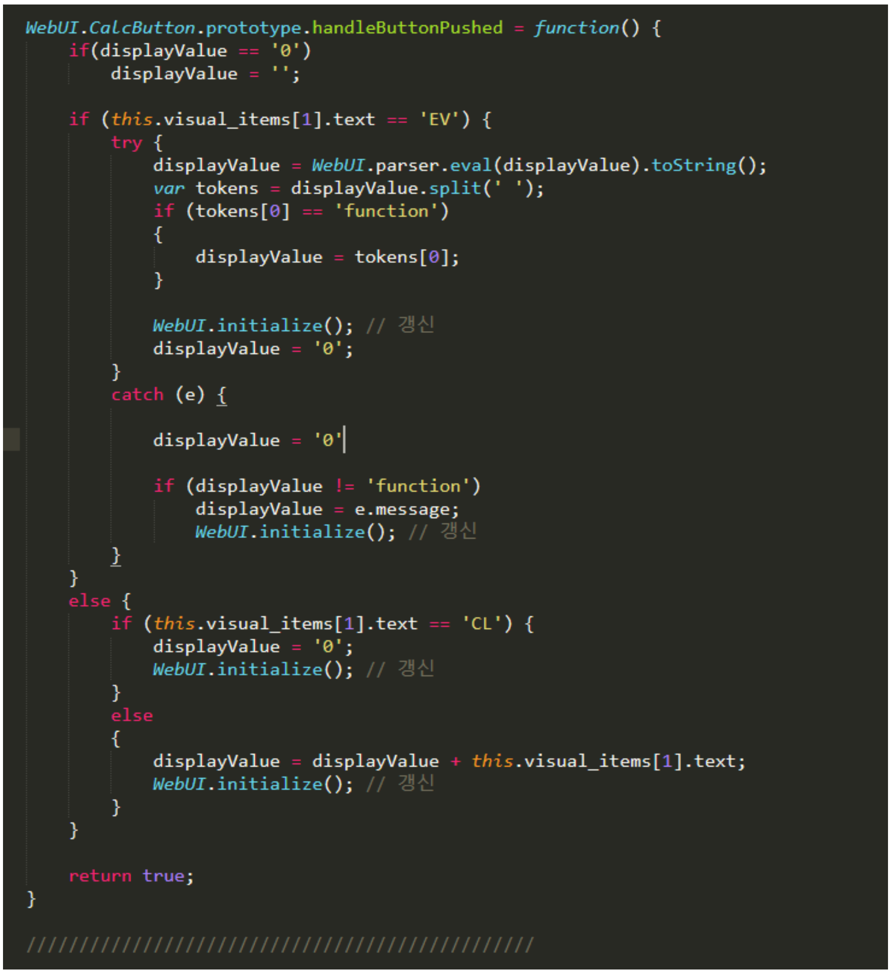
CalcButton initVisualItems 구현



```
WebUI.CalcButton.prototype.initVisualItems = function() {
    let background = new fabric.Rect({
        left: this.position.left,
        top: this.position.top,
        width: this.desired_size.width,
        height: this.desired_size.height,
        fill: this.fill_color,
        stroke: this.stroke_color,
        strokeWidth: 1,
        selectable: false
    });

    let text = new fabric.Text(this.label, {
        left: this.position.left,
        top: this.position.top,
        selectable: false,
        fontFamily: this.font_family,
        fontSize:   this.font_size,
        fontWeight: this.font_weight,
        textAlign:  this.text_align,
        stroke:     this.text_color,
        fill:       this.text_color,
    });

    let bound = text.getBoundingRect();
    text.left = this.position.left + this.desired_size.width/2 - bound.width/2;
    text.top = this.position.top + this.desired_size.height/2 - bound.height/2;

    this.size = this.desired_size;

    //
    this.visual_items.push(background);
    this.visual_items.push(text);
    this.is_resource_ready = true;
}
```

CalcButton handleButtonPushed 구현(강의자료)

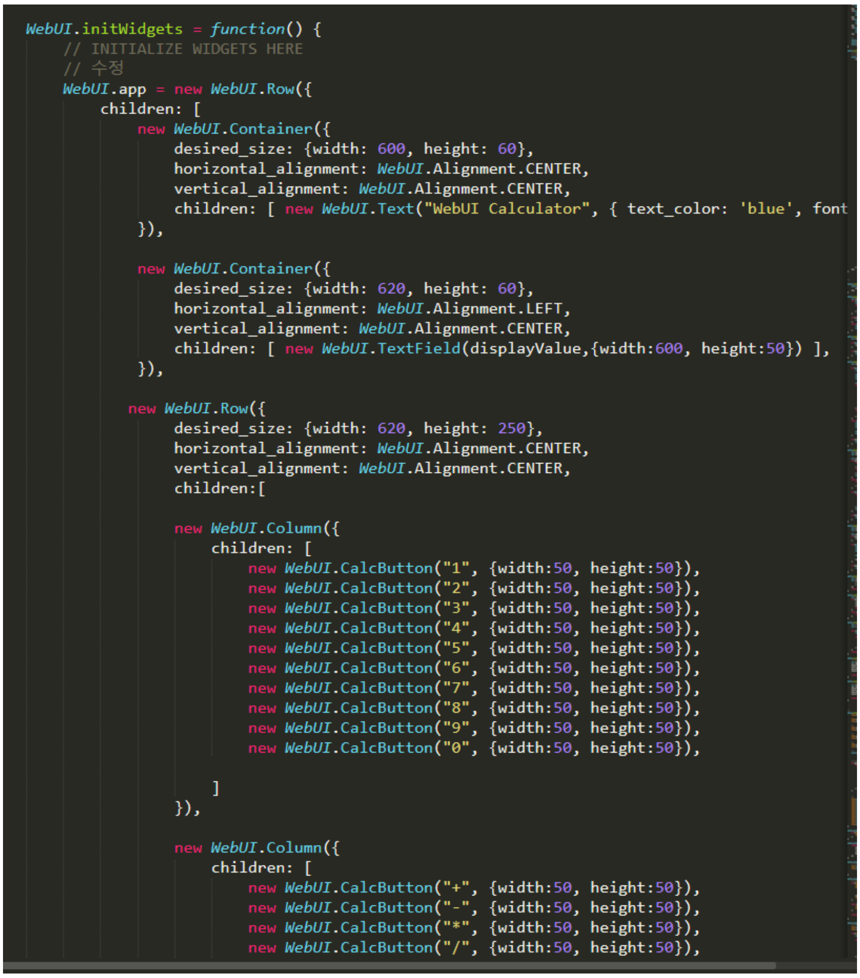
갱신을 위해 WebUI.initialize() 구문을 추가하였다.



```
WebUI.CalcButton.prototype.handleButtonPushed = function() {
    if(displayValue == '0')
        displayValue = '';

    if (this.visual_items[1].text == 'EV') {
        try {
            displayValue = WebUI.parser.eval(displayValue).toString();
            var tokens = displayValue.split(' ');
            if (tokens[0] == 'function')
            {
                displayValue = tokens[0];
            }

            WebUI.initialize(); // 갱신
            displayValue = '0';
        }
        catch (e) {

            displayValue = '0'

            if (displayValue != 'function')
                displayValue = e.message;
                WebUI.initialize(); // 갱신
        }
    }
    else {
        if (this.visual_items[1].text == 'CL') {
            displayValue = '0';
            WebUI.initialize(); // 갱신
        }
        else
        {
            displayValue = displayValue + this.visual_items[1].text;
            WebUI.initialize(); // 갱신
        }
    }

    return true;
}

////////////////////////////////////////////////
```

initWidgets 재정의

레이아웃에 맞춰 initWidgets을 수정하였다.

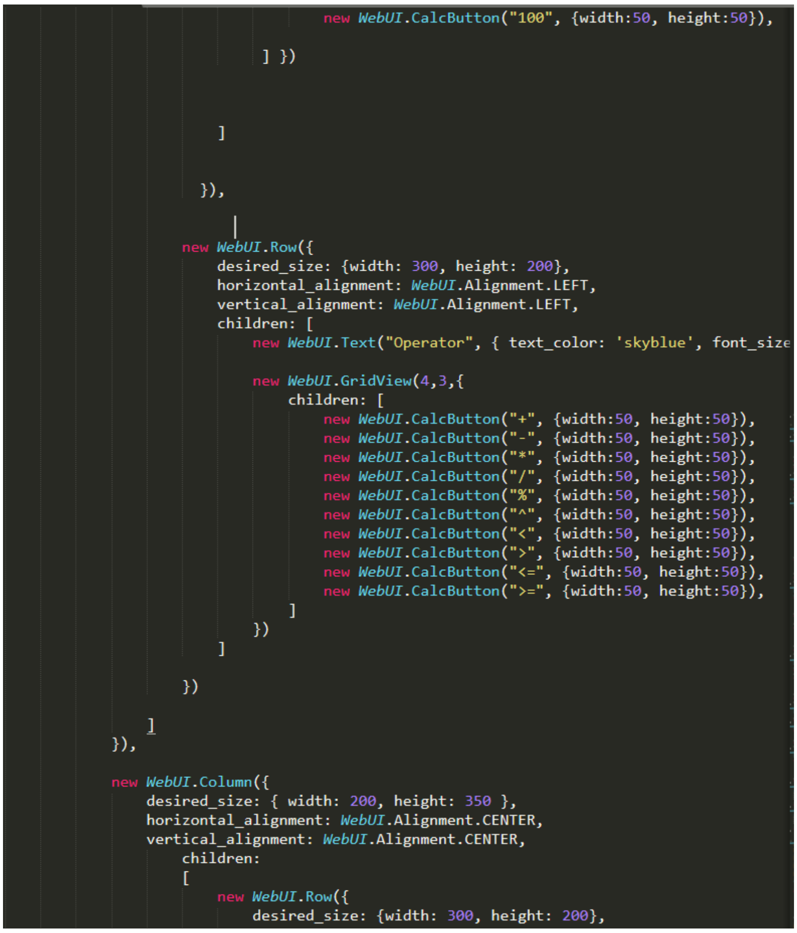


```
WebUI.initWidgets = function() {
    // INITIALIZE WIDGETS HERE
    // 수정
    WebUI.app = new WebUI.Row({
        children: [
            new WebUI.Container({
                desired_size: {width: 600, height: 60},
                horizontal_alignment: WebUI.Alignment.CENTER,
                vertical_alignment: WebUI.Alignment.CENTER,
                children: [ new WebUI.Text("WebUI Calculator", { text_color: 'blue', font
            }),

            new WebUI.Container({
                desired_size: {width: 620, height: 60},
                horizontal_alignment: WebUI.Alignment.LEFT,
                vertical_alignment: WebUI.Alignment.CENTER,
                children: [ new WebUI.TextField(displayValue,{width:600, height:50}) ],
            }),

           new WebUI.Row({
                desired_size: {width: 620, height: 250},
                horizontal_alignment: WebUI.Alignment.CENTER,
                vertical_alignment: WebUI.Alignment.CENTER,
                children:[

                new WebUI.Column({
                    children: [
                        new WebUI.CalcButton("1", {width:50, height:50}),
                        new WebUI.CalcButton("2", {width:50, height:50}),
                        new WebUI.CalcButton("3", {width:50, height:50}),
                        new WebUI.CalcButton("4", {width:50, height:50}),
                        new WebUI.CalcButton("5", {width:50, height:50}),
                        new WebUI.CalcButton("6", {width:50, height:50}),
                        new WebUI.CalcButton("7", {width:50, height:50}),
                        new WebUI.CalcButton("8", {width:50, height:50}),
                        new WebUI.CalcButton("9", {width:50, height:50}),
                        new WebUI.CalcButton("0", {width:50, height:50}),

                    ]
                }),

                new WebUI.Column({
                    children: [
                        new WebUI.CalcButton("+", {width:50, height:50}),
                        new WebUI.CalcButton("-", {width:50, height:50}),
                        new WebUI.CalcButton("*", {width:50, height:50}),
                        new WebUI.CalcButton("/", {width:50, height:50}),
```

4. 3단계

initWidgets 재정의

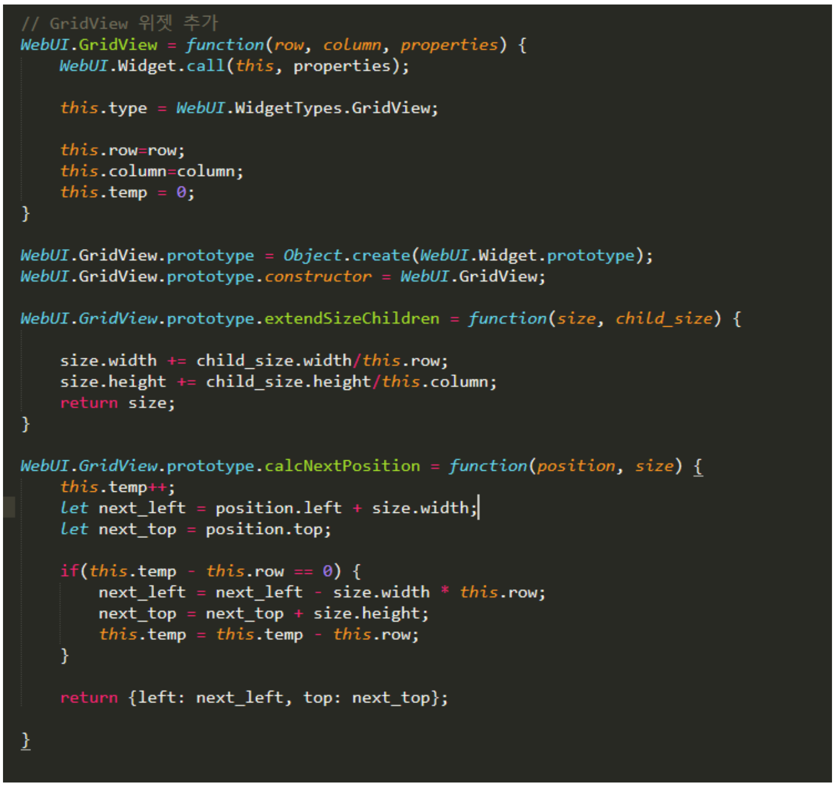
GridView를 사용하고 편의성을 위해서 initWidgets을 수정하였다.



```
                        new WebUI.CalcButton("100", {width:50, height:50}),

                 ] })



            ]


          }),

        new WebUI.Row({
            desired_size: {width: 300, height: 200},
            horizontal_alignment: WebUI.Alignment.LEFT,
            vertical_alignment: WebUI.Alignment.LEFT,
            children: [
                new WebUI.Text("Operator", { text_color: 'skyblue', font_size

                new WebUI.GridView(4,3,{
                    children: [
                        new WebUI.CalcButton("+", {width:50, height:50}),
                        new WebUI.CalcButton("-", {width:50, height:50}),
                        new WebUI.CalcButton("*", {width:50, height:50}),
                        new WebUI.CalcButton("/", {width:50, height:50}),
                        new WebUI.CalcButton("%", {width:50, height:50}),
                        new WebUI.CalcButton("^", {width:50, height:50}),
                        new WebUI.CalcButton("<", {width:50, height:50}),
                        new WebUI.CalcButton(">", {width:50, height:50}),
                        new WebUI.CalcButton("<=", {width:50, height:50}),
                        new WebUI.CalcButton(">=", {width:50, height:50}),
                    ]
                })
            ]

        })

    ]
}),

new WebUI.Column({
    desired_size: { width: 200, height: 350 },
    horizontal_alignment: WebUI.Alignment.CENTER,
    vertical_alignment: WebUI.Alignment.CENTER,
        children:
        [
            new WebUI.Row({
                desired_size: {width: 300, height: 200},
```

GridView 위젯 추가

열과 행에 맞춰서 위젯이 추가된다.

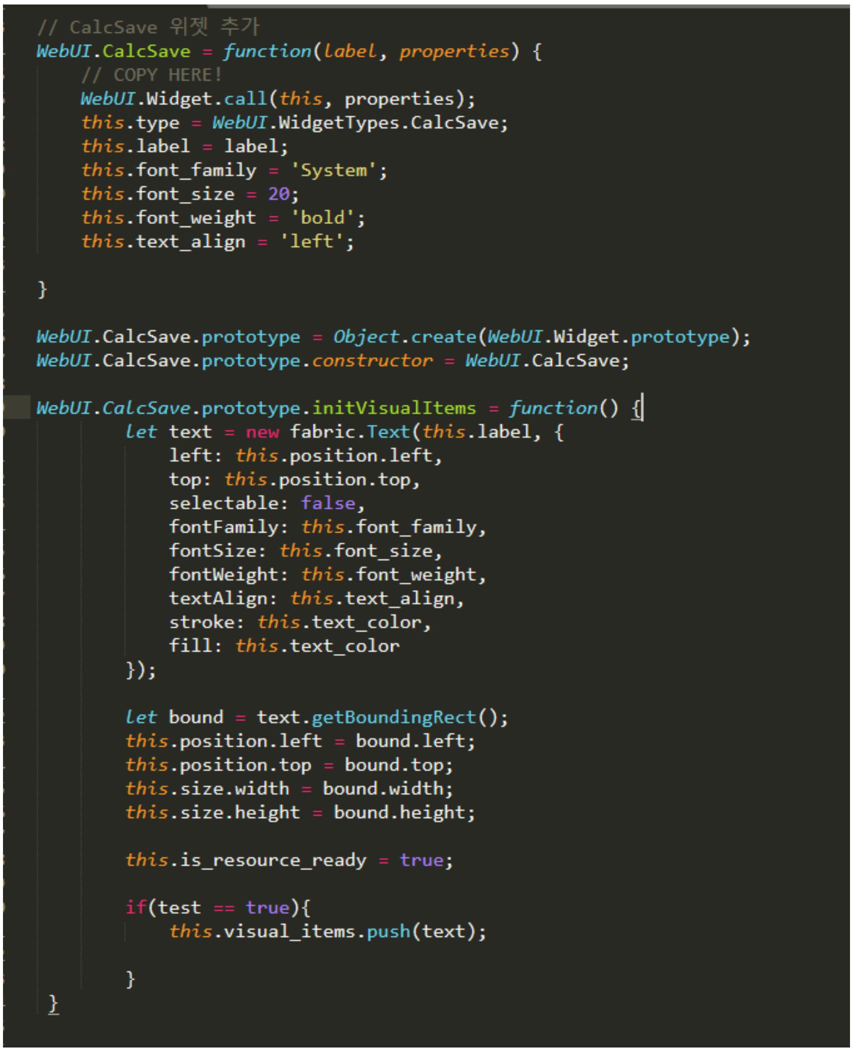


```
// GridView 위젯 추가
WebUI.GridView = function(row, column, properties) {
    WebUI.Widget.call(this, properties);

    this.type = WebUI.WidgetTypes.GridView;

    this.row=row;
    this.column=column;
    this.temp = 0;
}

WebUI.GridView.prototype = Object.create(WebUI.Widget.prototype);
WebUI.GridView.prototype.constructor = WebUI.GridView;

WebUI.GridView.prototype.extendSizeChildren = function(size, child_size) {

    size.width += child_size.width/this.row;
    size.height += child_size.height/this.column;
    return size;
}

WebUI.GridView.prototype.calcNextPosition = function(position, size) {
    this.temp++;
    let next_left = position.left + size.width;
    let next_top = position.top;

    if(this.temp - this.row == 0) {
        next_left = next_left - size.width * this.row;
        next_top = next_top + size.height;
        this.temp = this.temp - this.row;
    }

    return {left: next_left, top: next_top};

}
```

CalcSave 위젯 추가

EV 버튼을 누르면 Result에 계산된 값이 저장된다(CL버튼 누르기 전까지 값 저장).

CL 버튼을 누르면 Result에 저장된 값이 사라진다.



```
// CalcSave 위젯 추가
WebUI.CalcSave = function(label, properties) {
    // COPY HERE!
    WebUI.Widget.call(this, properties);
    this.type = WebUI.WidgetTypes.CalcSave;
    this.label = label;
    this.font_family = 'System';
    this.font_size = 20;
    this.font_weight = 'bold';
    this.text_align = 'left';

}

WebUI.CalcSave.prototype = Object.create(WebUI.Widget.prototype);
WebUI.CalcSave.prototype.constructor = WebUI.CalcSave;

WebUI.CalcSave.prototype.initVisualItems = function() {
        let text = new fabric.Text(this.label, {
            left: this.position.left,
            top: this.position.top,
            selectable: false,
            fontFamily: this.font_family,
            fontSize: this.font_size,
            fontWeight: this.font_weight,
            textAlign: this.text_align,
            stroke: this.text_color,
            fill: this.text_color
        });

        let bound = text.getBoundingRect();
        this.position.left = bound.left;
        this.position.top = bound.top;
        this.size.width = bound.width;
        this.size.height = bound.height;

        this.is_resource_ready = true;

        if(test == true){
            this.visual_items.push(text);

        }
 }
```

5. 논의

과제에서 요구하는 조건은 구현하였지만 3단계에서 GraphPlot을 이용하여 사용자 정의 함수를 시각화하지 못하였다.

WebUI를 이용하여 작동원리를 이해하는데 도움이 많이 되었고 나중에 이 코드를 다시 수정하고 개선하게 된다면 그래프 시각화를 꼭 추가해보고 싶다.