

가

Last update: 2025/11/05

가	1
.....	1
.....	1

가

JsBarcode 가 .

JsBarcode xframe5.js ENGINEURL /ext/lib (:
C:/xframe5/xf5/ext/lib/JsBarcode.all.min.js).

display_mode, display_value, format가 .

on_click, on_valid가 .

API settext, setformat, getinnerhtml .

: /HTML5/COMPONENT/BARCODE/barcode_basic

- [barcode_basic.xml](#)
- [barcode_basic.js](#)
- [barcode_basic.css](#)

```
function btnSettext_on_mouseup(objInst)
{
    var barcode_value;

    //
    barcode_value = this.barcode2.gettext();

    // 0 가
    barcode_value = barcode_value + "0";

    //
    this.barcode2.settext(barcode_value);
}

function barcode2_on_valid(objInst, bValid)
{
    if (bValid) {
```

```
        console.log("                .");
    } else {
        console.log("                .");
    }

    console.log("=====");
}

function btnSetformat_on_mouseup(objInst)
{
    var format, format_txt;

    format = this.barcode3.getformat();
    console.log("                = " + format);

    format = (format + 1) % 18;
    console.log("                = " + format);

    switch (format) {
        case 0: // 0: CODE128
            barcode_value = "Example 1234";
            format_txt = "0: CODE128";
            break;
        case 1: // 1: CODE128A
            barcode_value = "EXAMPLE";
            format_txt = "1: CODE128A";
            break;
        case 2: // 2: CODE128B
            barcode_value = "Example text";
            format_txt = "2: CODE128B";
            break;
        case 3: // 3: CODE128C
            barcode_value = "12345678";
            format_txt = "3: CODE128C";
            break;
        case 4: // 4: EAN-13
            barcode_value = "1234567890128";
            format_txt = "4: EAN-13";
            break;
        case 5: // 5: EAN-8
            barcode_value = "12345670";
            format_txt = "5: EAN-8";
            break;
        case 6: // 6: EAN-5
            barcode_value = "90200";
            format_txt = "6: EAN-5";
            break;
        case 7: // 7: EAN-2
            barcode_value = "05";
            format_txt = "7: EAN-2";
            break;
    }
}
```

```
case 8: // 8: UPC
    barcode_value = "123456789999";
    format_txt = "8: UPC";
    break;
case 9: // 9: CODE39
    barcode_value = "EXAMPLE TEXT";
    format_txt = "9: CODE39";
    break;
case 10: // 10: ITF14
    barcode_value = "10012345000017";
    format_txt = "10: ITF14";
    break;
case 11: // 11: MSI
    barcode_value = "123456";
    format_txt = "11: MSI";
    break;
case 12: // 12: MSI10
    barcode_value = "123456";
    format_txt = "12: MSI10";
    break;
case 13: // 13: MSI11
    barcode_value = "123456";
    format_txt = "13: MSI11";
    break;
case 14: // 14: MSI1010
    barcode_value = "123456";
    format_txt = "14: MSI1010";
    break;
case 15: // 15: MSI1110
    barcode_value = "123456";
    format_txt = "15: MSI1110";
    break;
case 16: // 16: pharmacode
    barcode_value = "1234";
    format_txt = "16: pharmacode";
    break;
case 17: // 17: codabar
    barcode_value = "A123456A";
    format_txt = "17: codabar";
    break;
}

this.barcode3.setformat(format);
this.barcode3.settext(barcode_value);

this.txtFormat.settext("[          ] " + format_txt);
}

function barcode3_on_valid(objInst, bValid)
{
    if (bValid) {
```

```
        console.log("                .");
    } else {
        console.log("                .");
    }

    console.log("=====");
}

function btnGetinnerhtml_on_mouseup(objInst)
{
    var barcode_html;

    //      html
    barcode_html = this.barcode4.getinnerhtml()

    //      가
    if(grd.getrowcount() == 0){
        grd.addrow();
    }

    //
    this.grd.setitemtext(0, 0, barcode_html);
}
```

From:

<http://technet.softbase.co.kr/wiki/> - **xFrame5 TechNet**

Permanent link:

http://technet.softbase.co.kr/wiki/guide/component/barcode/barcode_basic



Last update: **2025/11/05 10:09**