

date picker

کی دگ ہنومن date picker :

```
<page onshown="dialog/datePickerDialog">
  <reactive-data id="defaultValue" valueBind="component/defaultValue"
elementBind="year: text" />
  <button onclick="dialog/datePickerDialog" text="Ss" />
</page>

<bottomsheetdialog id="datePickerDialog" onshown="script/datePickerScript: initCalendar()">
  <text margin_bottom="16">دینک ب ایختن ا ار رطن دروم دنسسم خیر ات</text>

  <frame layout="frame">
    <imagebutton onclick="script/datePickerScript: nextMonth()" height="42"
src="images/previous.png" gravity="left" />
    <frame layout="linear_horizontal">
      <text id="year">1401</text>
      <text id="month">ری</text>
    </frame>
    <imagebutton onclick="script/datePickerScript: lastMonth()" height="42"
src="images/next.png" gravity="right" />
  </frame>

  <frame id="calendar" layout="linear_vertical">

</frame>

  <frame layout="linear_horizontal" margin_top="10" borderCornerRadius="5"
background="">
    <button background="#55BECC" textsize="14" textcolor="white">ب ایختن ا</button>
  </frame>

  <include layout="comp/date-picker.js.appsan" />
</bottomsheetdialog>
```

```
<script id="datePickerScript">
```

```
class PersianDate extends Date {
  constructor(...args) {
    super(...args);
  }

  toLocaleDateString = () => super.toLocaleDateString('fa-IR-u-nu-latn');
  getParts = () => this.toLocaleDateString().split("/")
  getDay = () => super.getDay() === 6 ? 0 : super.getDay() + 1
  getDate = () => this.getParts()[2];
  getMonth = () => this.getParts()[1];
  getYear = () => this.getParts()[0];
  getMonthName = () => this.toLocaleDateString("fa-IR", {month: 'long'});
  getDayName = () => this.toLocaleDateString("fa-IR", {weekday: 'long'});
}
```

```
const jCal = {
  monthsDays: [
    31, 31, 31,
    31, 31, 31,
    30, 30, 30,
    30, 30, 29
  ],
  monthNames: [
    'نی‌درورف',
    'تش‌ه‌بی‌در ا',
    'د‌د‌رخ',
    'ریت',
    'د‌د‌رم',
    'روی‌رهش',
    'رهم',
    'ن‌اب‌آ',
    'ر‌د‌آ',
    'ی‌د',
    'ن‌م‌ه‌ب',
    'د‌ن‌فس‌ا',
  ],
  getMonthDays: (month) => {
    return jCal.monthsDays[month - 1];
  }
}
```

```

},
gregorian_to_jalali: (gy, gm, gd) => {
    var g_d_m = [0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334];
    var jy = (gy <= 1600) ? 0 : 979;
    gy -= (gy <= 1600) ? 621 : 1600;
    var gy2 = (gm > 2) ? (gy + 1) : gy;
    var days = (365 * gy) + (parseInt((gy2 + 3) / 4)) - (parseInt((gy2 + 99) /
100))
        + (parseInt((gy2 + 399) / 400)) - 80 + gd + g_d_m[gm - 1];
    jy += 33 * (parseInt(days / 12053));
    days %= 12053;
    jy += 4 * (parseInt(days / 1461));
    days %= 1461;
    jy += parseInt((days - 1) / 365);
    if (days > 365) days = (days - 1) % 365;
    var jm = (days < 186) ? 1 + parseInt(days / 31) : 7 + parseInt((days - 186) /
30);

    var jd = 1 + ((days < 186) ? (days % 31) : ((days - 186) % 30));
    return [jy, jm, jd];
},
jalali_to_gregorian: (jy, jm, jd) => {
    var gy = (jy <= 979) ? 621 : 1600;
    jy -= (jy <= 979) ? 0 : 979;
    var days = (365 * jy) + ((parseInt(jy / 33)) * 8) + (parseInt(((jy % 33) + 3) /
4))
        + 78 + jd + ((jm < 7) ? (jm - 1) * 31 : ((jm - 7) * 30) + 186);
    gy += 400 * (parseInt(days / 146097));
    days %= 146097;
    if (days > 36524) {
        gy += 100 * (parseInt(--days / 36524));
        days %= 36524;
        if (days >= 365) days++;
    }
    gy += 4 * (parseInt((days) / 1461));
    days %= 1461;
    gy += parseInt((days - 1) / 365);
    if (days > 365) days = (days - 1) % 365;
    var gd = days + 1;
    var sal_a = [0, 31, ((gy % 4 == 0 && gy % 100 != 0) || (gy % 400 == 0)) ? 29 :
28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

```

```

        var gm
        for (gm = 0; gm < 13; gm++) {
            var v = sal_a[gm];
            if (gd <= v) break;
            gd -= v;
        }
        return [gy, gm, gd];
    },
    getMonthName: (month) => {
        return jCal.monthNames[month - 1];
    }
}

```

```

let pDate = new PersianDate();

```

```

function nextMonth() {
    let month = +pDate.getMonth() + 1;
    if (month === 13) month = 1;
    const firstOfMonth = jCal.jalali_to_gregorian(pDate.getYear(), month, 1);
    pDate = new PersianDate(firstOfMonth);
    initCalendar();
}

```

```

function lastMonth() {
    let month = pDate.getMonth() - 1;
    if (month === 0) month = 12;
    const firstOfMonth = jCal.jalali_to_gregorian(pDate.getYear(), month, 1);
    pDate = new PersianDate(firstOfMonth);
    initCalendar();
}

```

```

function initCalendar() {
    let rows = '<frame layout="linear_vertical">';

    const firstOfMonth = jCal.jalali_to_gregorian(pDate.getYear(), pDate.getMonth(),
1);

    let monthOffset = new Date(firstOfMonth[0], firstOfMonth[1] - 1,
firstOfMonth[2]).getDay() + 2;
    if(monthOffset > 6) monthOffset = monthOffset - 7;

```

```

// rows += `${new Date(firstOfMonth[0], firstOfMonth[1],
firstOfMonth[2])}</text>`

// rows += `${new Date(firstOfMonth[0], firstOfMonth[1] - 1,
firstOfMonth[2]).getDay()}</text>`

// rows += `${firstOfMonth}</text>`
// rows += `${monthOffset}</text>`
// rows += `${(1 - monthOffset)}</text>`

rows += `${(i > 0 && i <= jCal.getMonthDays(pDate.getMonth())) ? i :
''}</text>\n`;
}
rows += `</frame>`
rows += `</frame>`

Appsan.setProperty("calendar", "innerElements", rows)
Appsan.setProperty("year", "text", pDate.getYear())
Appsan.setProperty("month", "text", jCal.getMonthName(pDate.getMonth()))
}
</script>

```