

## 1.按钮控件

```
"ui";

ui.layout(
    <vertical padding="16">
        <button text="普通按钮" w="auto"/>
        <button text="带颜色按钮" style="widget.AppCompat.Button.Colored"
w="auto"/>
        <button text="无边框按钮" style="widget.AppCompat.Button.Borderless"
w="auto"/>
        <button text="无边框有颜色按钮"
style="widget.AppCompat.Button.Borderless.Colored" w="auto"/>
        <button text="长长的按钮" w="*" />
        <button id="click_me" text="点我" w="auto"/>
    </vertical>
);

ui.click_me.on("click", ()=>{
    toast("我被点啦");
});

ui.click_me.on("long_click", ()=>{
    toast("我被长按啦");
});
```

## 2.复选框与单选框控件

```
"ui";

ui.layout(
    <vertical padding="16">
        <checkbox id="cb1" text="复选框"/>
        <checkbox id="cb2" checked="true" text="勾选的复选框"/>
        <radiogroup>
            <radio text="单选框1"/>
            <radio text="单选框2"/>
            <radio text="单选框3"/>
        </radiogroup>
        <radiogroup marginTop="16">
            <radio text="单选框1"/>
            <radio text="单选框2"/>
            <radio text="勾选的单选框3" checked="true"/>
        </radiogroup>
    </vertical>
);

ui.cb1.on("check", (checked)=>{
    if(checked){
        toast("第一个框被勾选了");
    }else{

```

```

        toast("第一个框被取消勾选了");
    }
});

```

### 3.进度条控件

```

"ui";

ui.layout(
    <vertical padding="16">
        <text text="处理中..." textColor="black" textSize="16sp"/>
        <progressbar />

        <text text="直线无限进度条" textColor="black" textSize="16sp"
marginTop="24"/>
        <progressbar indeterminate="true"
style="@style/Base.Widget.AppCompat.ProgressBar.Horizontal"/>

        <text text="直线进度条" textColor="black" textSize="16sp" marginTop="24"/>
        <progressbar progress="30"
style="@style/Base.Widget.AppCompat.ProgressBar.Horizontal"/>

        <text text="可调节进度条" textColor="black" textSize="16sp"
marginTop="24"/>
        <seekbar progress="20"/>

        <horizontal gravity="center" marginTop="24">
            <text id="progress_value" textColor="black" textSize="16sp"
margin="8" text="0"/>
            <progressbar id="progress" w="*"
style="@style/Base.Widget.AppCompat.ProgressBar.Horizontal"/>
        </horizontal>
        <button id="download">开始下载</button>
    </vertical>
);

var downloadId = null;

ui.download.click(()=>{
    if(downloadId != null){
        stopDownload();
    }else{
        startDownload();
    }
});

function stopDownload(){
    ui.download.text("开始下载");
    clearInterval(downloadId);
    downloadId = null;
}

function startDownload(){
    if(ui.progress.getProgress() == 100){
        ui.progress.setProgress(0);
    }
}

```

```

}
ui.download.text("停止下载");
downloadId = setInterval(()=>{
    var p = ui.progress.getProgress();
    p++;
    if(p > 100){
        stopDownload();
        return;
    }
    ui.progress.setProgress(p);
    ui.progress_value.setText(p.toString());
}, 200);
}

```

## 4.卡片布局

```

"ui";

ui.layout(
    <vertical>
        <appbar>
            <toolbar id="toolbar" title="卡片布局"/>
        </appbar>
        <card w="*" h="70" margin="10 5" cardCornerRadius="2dp"
            cardElevation="1dp" gravity="center_vertical">
            <vertical padding="18 8" h="auto">
                <text text="写操作系统作业" textColor="#222222" textSize="16sp"/>
                <text text="明天第1~2节" textColor="#999999" textSize="14sp"/>
            </vertical>
            <View bg="#f44336" h="*" w="10"/>
        </card>
        <card w="*" h="70" margin="10 5" cardCornerRadius="2dp"
            cardElevation="1dp" gravity="center_vertical">
            <vertical padding="18 8" h="auto">
                <text text="修复ui模式的Bug" textColor="#222222" textSize="16sp"/>
                <text text="无限期" textColor="#999999" textSize="14sp"/>
            </vertical>
            <View bg="#ff5722" h="*" w="10"/>
        </card>
        <card w="*" h="70" margin="10 5" cardCornerRadius="2dp"
            cardElevation="1dp" gravity="center_vertical">
            <vertical padding="18 8" h="auto">
                <text text="发布Auto.js 5.0.0正式版" textColor="#222222"
textSize="16sp"/>
                <text text="2019年1月" textColor="#999999" textSize="14sp"/>
            </vertical>
            <View bg="#4caf50" h="*" w="10"/>
        </card>
        <card w="*" h="70" margin="10 5" cardCornerRadius="2dp"
            cardElevation="1dp" gravity="center_vertical">
            <vertical padding="18 8" h="auto">
                <text text="完成毕业设计和论文" textColor="#222222"
textSize="16sp"/>
                <text text="2019年4月" textColor="#999999" textSize="14sp"/>
            </vertical>

```

```

        <view bg="#2196f3" h="*" w="10"/>
      </card>
    </vertical>
  );

```

## 5.列表控件

```

"ui";

ui.layout(
  <frame>
    <list id="list">
      <vertical>
        <text id="name" textSize="16sp" textcolor="#000000" text="姓名:
{{name}}"/>
        <text id="age" textSize="16sp" textcolor="#000000" text="年龄:
{{age}}岁"/>
        <button id="deleteItem" text="删除"/>
      </vertical>
    </list>
  </frame>
);

var items = [
  {name: "小明", age: 18}, {name: "小红", age: 30},
  {name: "小东", age: 19}, {name: "小强", age: 31},
  {name: "小满", age: 20}, {name: "小一", age: 32},
  {name: "小和", age: 21}, {name: "小二", age: 1},
  {name: "小贤", age: 22}, {name: "小三", age: 2},
  {name: "小伟", age: 23}, {name: "小四", age: 3},
  {name: "小黄", age: 24}, {name: "小五", age: 4},
  {name: "小健", age: 25}, {name: "小六", age: 5},
  {name: "小啦", age: 26}, {name: "小七", age: 6},
  {name: "小哈", age: 27}, {name: "小八", age: 7},
  {name: "小啊", age: 28}, {name: "小九", age: 8},
  {name: "小啪", age: 29}, {name: "小十", age: 9}
];

ui.list.setDataSource(items);

ui.list.on("item_click", function(item, i, itemView, listView){
  toast("被点击的人名字为: " + item.name + ", 年龄为: " + item.age);
});

ui.list.on("item_bind", function(itemView, itemHolder){
  itemView.deleteItem.on("click", function(){
    let item = itemHolder.item;
    toast("被删除的人名字为: " + item.name + ", 年龄为: " + item.age);
    items.splice(itemHolder.position, 1);
  });
});

```

## 6. 图片控件

```
"ui";

ui.layout(
<scroll>
    <vertical bg="#707070" padding="16">
        <text text="网络图片" textColor="black" textSize="16sp" marginTop="16"/>
        

        <text text="带边框的图片" textColor="black" textSize="16sp"
            marginTop="16"/>
        

        <text text="圆形图片" textColor="black" textSize="16sp" marginTop="16"/>
        

        <text text="带边框的圆形图片" textColor="black" textSize="16sp"
            marginTop="16"/>
        

        <text text="圆角图片" textColor="black" textSize="16sp" marginTop="16"/>
        
        <button id="change_img" text="更改图片"/>
    </vertical>
</scroll>
);

ui.change_img.on("click", ()=>{
    ui.rounded_img.setSource("http://www.autojs.org/assets/uploads/profile/1-profilecover.jpeg");
});
```

## 7. 文本控件

```
"ui";

ui.layout(
    <vertical padding="16">
        <text textSize="40sp">大字</text>
        <text textSize="12sp">小字</text>
        <text textStyle="bold" textColor="black">加粗</text>
        <text textStyle="italic">斜体</text>
```

```

        <text textColor="#00ff00">原谅色</text>
        <text margin="8">Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。</text>
        <text maxLines="1" ellipsize="end" margin="8">Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。</text>
        <text maxLines="2" ellipsize="end" margin="8">Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。</text>
        <text w="*" gravity="center" textSize="20sp">居中</text>
        <text autoLink="all">自动超链接网址www.baidu.com，邮箱 123@qq.com等</text>
    </vertical>
);

```

## 8. 下拉菜单

```

"ui";

ui.layout(
    <vertical padding="16">
        <horizontal>
            <text textSize="16sp">下拉菜单</text>
            <spinner id="sp1" entries="选项1|选项2|选项3"/>
        </horizontal>
        <horizontal>
            <text textSize="16sp">对话框菜单</text>
            <spinner id="sp2" entries="选项4|选项5|选项6" spinnerMode="dialog"/>
        </horizontal>
        <button id="ok">确定</button>
        <button id="select3">选择选项3</button>
    </vertical>
);

ui.ok.on("click", ()=>{
    var i = ui.sp1.getSelectedPosition();
    var j = ui.sp2.getSelectedPosition();
    toast("您的选择是选项" + (i + 1) + "和选项" + (j + 4));
});

ui.select3.on("click", ()=>{
    ui.sp1.setSelection(2);
});

```

## 9. 自定义控件-带颜色按钮

```

"ui";

var ColoredButton = (function() {
    //继承ui.Widget
    util.extend(ColoredButton, ui.Widget);

    function ColoredButton() {
        //调用父类构造函数
        ui.Widget.call(this);
        //自定义属性color, 定义按钮颜色
        this.defineAttr("color", (view, name, defaultGetter) => {
            return this._color;
        }, (view, name, value, defaultSetter) => {
            this._color = value;
            view.attr("backgroundTint", value);
        });
        //自定义属性onClick, 定义被点击时执行的代码
        this.defineAttr("onClick", (view, name, defaultGetter) => {
            return this._onClick;
        }, (view, name, value, defaultSetter) => {
            this._onClick = value;
        });
    }
    ColoredButton.prototype.render = function() {
        return (
            <button textSize="16sp" style="widget.AppCompat.Button.Colored"
w="auto"/>
        );
    }
    ColoredButton.prototype.onViewCreated = function(view) {
        view.on("click", () => {
            if (this._onClick) {
                eval(this._onClick);
            }
        });
    }
    ui.registerWidget("colored-button", ColoredButton);
    return ColoredButton;
})();

ui.layout(
    <vertical>
        <colored-button text="第一个按钮" color="#ff5722"/>
        <colored-button text="第二个按钮" onClick="hello()"/>
    </vertical>
);

function hello() {
    alert("Hello ~");
}

```