



Lists

(Information and examples taken from the DroidScript documentation)

Description

Create List objects using the **CreateList** method of the **app** object:

```
lst = app.CreateList( list, width, height, options );
```

Use the **SetOnTouch** and **SetOnLongTouch** methods of the List object to set the name of a function you want to be called when a list item is selected. The selected item is passed into your OnTouch callback function as a parameter every time an item is selected or long touched.

Example - Simple

```
function OnStart()
{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    lst = app.CreateList( "Fred,Bill,Jim", 0.8, 0.4 );
    lst.SetOnTouch( lst_OnTouch );
    lst.SetOnLongTouch( lst_OnLongTouch );
    lay.AddChild( lst );

    app.AddLayout( lay );
}

function lst_OnTouch( item )
{
    app.ShowPopup( "Item = " + item, "Short" );
}
function lst_OnLongTouch( item )
{
    app.ShowPopup( "Long Touch = " + item, "Short" );
}
```

You can change the look of a List using the **SetBackColor** and **SetTextColor** functions on the list object. You can also set a background image/pattern or background gradient for the List using the **SetBackground** and **SetBackGradient** functions.

Example - Gray on white

```
function OnStart()
{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    lst = app.CreateList( "Fred,Bill,Jim", 0.8, 0.4 );
    lst.SetTextColor( "#ff666666" );
    lst.SetBackColor( "#ffffffff" );
    lst.SetOnTouch( lst_OnTouch );
    lay.AddChild( lst );

    app.AddLayout( lay );
}

function lst_OnTouch( item )
{
    app.ShowPopup( "Touched Item = " + item );
}
```

The List object also supports **multi-line** list items and can show certain types of **icon**. Multi-line items are created by dividing each list item up using the ':' (colon) character. If you need to use a colon character in your item text then use this character sequence: ^c^.

You can have one icon and up to three lines of text using the following formats:

- title : icon
- title : body : icon
- title : body : extra : icon

The available icons types are displayed using the following key words:

- "folder"
- "audiofolder"
- "photofolder"
- "videofolder"
- "audio"
- "photo"
- "video"
- "playlist"

Example - Title + Icon

```
function OnStart()
{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    var data = "Folder:folder,Audio:audio,Photo:photo,Video:video";
    lst = app.CreateList( data, 0.8, 0.4 );
    lst.SetOnTouch( lst_OnTouch );
    lay.AddChild( lst );

    app.AddLayout( lay );
}

function lst_OnTouch( item )
{
}
```

```
    app.ShowPopup( "Touched Item = " + item );  
}
```

Example - Title + Icon

```
function OnStart()  
{  
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );  
  
    var data = "The Hobbit:Author^c^ J.R.R. Tolkien:null";  
    data += ",Watership Down:Author^c^ Richard Adams:null";  
    lst = app.CreateList( data, 0.8, 0.4 );  
    lst.SetOnTouch( lst_OnTouch );  
    lay.AddChild( lst );  
  
    app.AddLayout( lay );  
}  
  
function lst_OnTouch( item )  
{  
    app.ShowPopup( "Touched Item = " + item );  
}
```

You can also create lists items that look like **buttons** by using one of the following options:

- "AlumButton"
- "GreenButton"
- "OrangeButton"

Example - Orange Buttons

```
function OnStart()  
{  
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );  
  
    var data = "Button 1,Button 2,Button 3,Button 4";  
    lst = app.CreateList( data, 0.8, 0.8, "OrangeButton" );  
    lst.SetBackColor( "#ffffff" );  
    lst.SetPadding( 0.1, 0.1, 0.1, 0.1 );  
    lst.SetOnTouch( lst_OnTouch );  
    lay.AddChild( lst );  
  
    app.AddLayout( lay );  
}  
  
function lst_OnTouch( item )  
{  
    app.ShowPopup( "Touched Item = " + item );  
}
```

Or create lists with **Gradient** backgrounds like this:

Example - Gradient Background

```
function OnStart()
```

```

{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    var data = "";
    for( var i=1; i<=30; i++)
    {
        if( i>1 ) data += ",";
        data += "Item "+i+":Details for item "+i+":null";
    }
    lst = app.CreateList( data, 1, 1, "WhiteGrad" );
    lst.SetTextColor1( "#ff555558");
    lst.SetTextColor2( "#ff555558" );
    lst.SetTextMargins( 0.04, 0, 0, 0 );
    lst.SetOnTouch( lst_OnTouch );
    lay.AddChild( lst );

    app.AddLayout( lay );
}

function lst_OnTouch( item )
{
    app.ShowPopup( "Touched Item = " + item );
}

```

Methods

Some controls use the same methods.
For examples of the **same methods** look here.

| Method | Description |
|--|---|
| ListView.AddItem(title,body,image) | |
| ListView.Destroy() | |
| ListView.GetAbsHeight() | |
| ListView.GetAbsWidth() | |
| ListView.GetHeight() | |
| ListView.GetList(delimiter) | string = list.GetList(","); |
| ListView.GetPosition() | |
| ListView.GetTextSize(mode) | |
| ListView.GetType() | |
| ListView.GetVisibility() | |
| ListView.GetWidth() | |
| ListView.Release() | |
| ListView.RemoveItem(title) | |
| ListView.ScrollToItem(name,body) | |
| ListView.SelectItem(title,body,scrollTo) | |
| ListView.SelectItemByIndex(index,scroll) | index is a number, scroll could be true or false |
| ListView.SetBackColor(colorCode) | |

| Method | Description |
|---|--|
| ListView.SetBackGradient(color1,color2,color3,p4,p5,p6,p7) | |
| ListView.SetBackGradientRadial(x,y,r,color1,color2,color3,p7) | |
| ListView.SetBackground(imagefile,options) | |
| ListView.SetDivider(height,color) | |
| ListView.SetEllipsize(mode) | |
| ListView.SetEllipsize1(mode) | |
| ListView.SetEllipsize2(mode) | |
| ListView.SetFontFile(file) | |
| ListView.SetHiTextColor1(colorCode) | |
| ListView.SetHiTextColor2(colorCode) | |
| ListView.SetItem(title,newTitle,newBody,newImage) | |
| ListView.SetList(list,delimiter) | |
| ListView.SetMargins(left,top,right,bottom) | |
| ListView.SetOnLongTouch(callback) | sets the function called when list is long-touched |
| ListView.SetOnTouch(callback) | sets the function called when list is touched |
| ListView.SetPadding(left,top,right,bottom) | |
| ListView.SetPosition(left,top,width,height) | |
| ListView.SetScale(x,y) | |
| ListView.SetSize(width,height) | |
| ListView.SetTextColor(colorCode) | |
| ListView.SetTextColor1(colorCode) | |
| ListView.SetTextColor2(colorCode) | |
| ListView.SetTextMargins(left,top,right,bottom) | |
| ListView.SetTextShadow(radius,dx,dy,color) | |
| ListView.SetTextShadow1(radius,dx,dy,color) | |
| ListView.SetTextShadow2(radius,dx,dy,color) | |
| ListView.SetTextSize(size,mode) | |
| ListView.SetVisibility(ShowHide) | |

Available Options

| Option | Description |
|--------------|--|
| AlumButton | Use this to display the list as buttons |
| FontAwesome | Use this to display Icons from this inbuilt font |
| GreenButton | Use this to display the list as buttons |
| OrangeButton | Use this to display the list as buttons |

| Option | Description |
|-----------|---|
| WhiteGrad | Use this to display the list with a gradient background |

Sample callbacks

```
function lst_OnTouch( title, body, type, index )
{
    app.ShowPopup( "id: " + title + " nick: " + body );
}
```

```
function lst_OnLongTouch( title, body, type, index )
{
    app.ShowPopup( "id: " + title + " nick: " + body );
}
```

```
function lst_OnTouch( title, body, type, index )
{
    //Shows the index of the touched item
    app.ShowPopup(index)
}
```

```
function lst_OnTouch( item )
{
    //Shows the touched item
    app.ShowPopup(item)
}
```