



DroidScript v1.8 —

Create Google-Android Apps easily

Absolute Beginners' Tutorial

EDITED-BY MOHEE JARADA (BUDAPEST JULY 2020) - Doc v1.1

✉ JARADA.MOHEE@GMAIL.COM

[TOOL ORIGINAL WEB-SITE: HTTP://DROIDSCRIPT.ORG](http://DROIDSCRIPT.ORG)

COPYRIGHT NOTICE: SOME OF CODE EXAMPLES, ARE EXTRACTED AND COPIED FROM DROIDSCRIPT TOOL'S HELP DOCUMENTATION AS IS.

“It’s one small step for man, one giant leap for mankind.”

- NEIL ARMSTRONG

Introduction

- **Compatible with Google Android OS versions v6, v7, v8, v9 and v10**
- **All Examples used here are tested under Google Android OS versions: v9 and v10.**

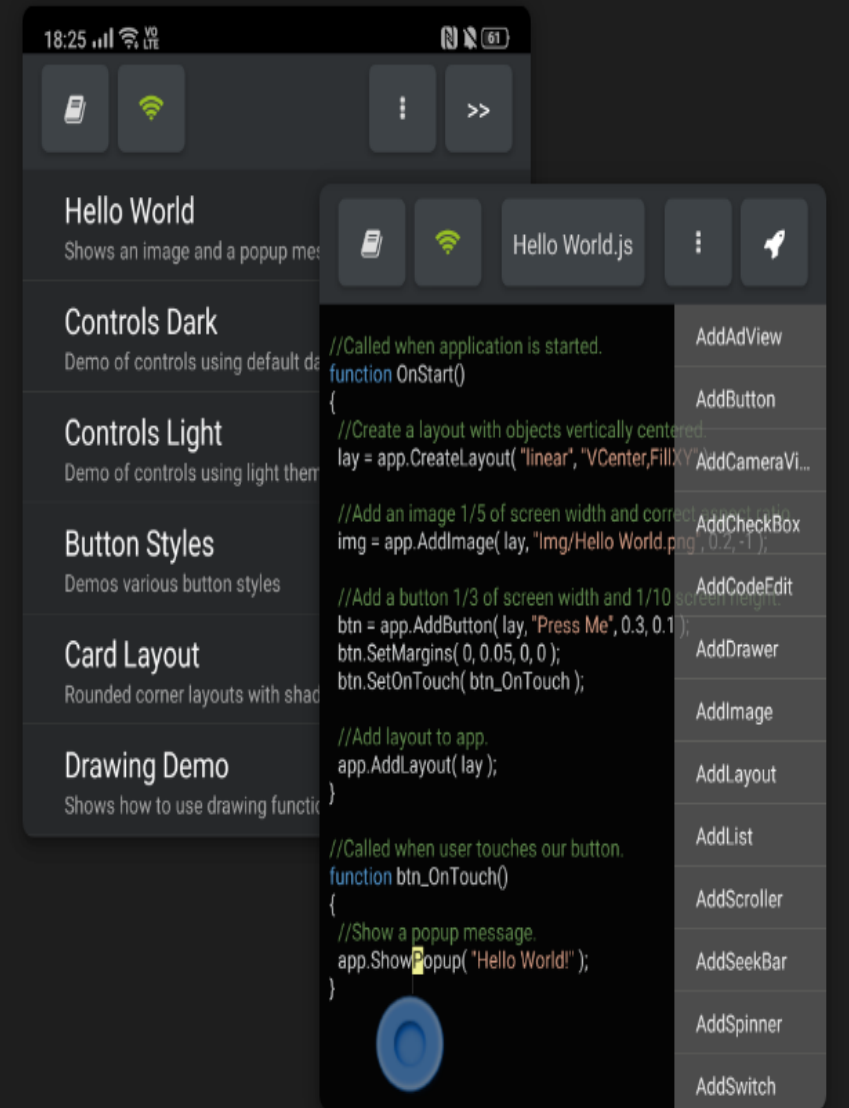
Mobile app development – Simplified.

DroidScript is an easy to use, portable coding tool which simplifies mobile app development. It dramatically improves productivity by speeding up development by as much as 10x compared with using the standard development tools.

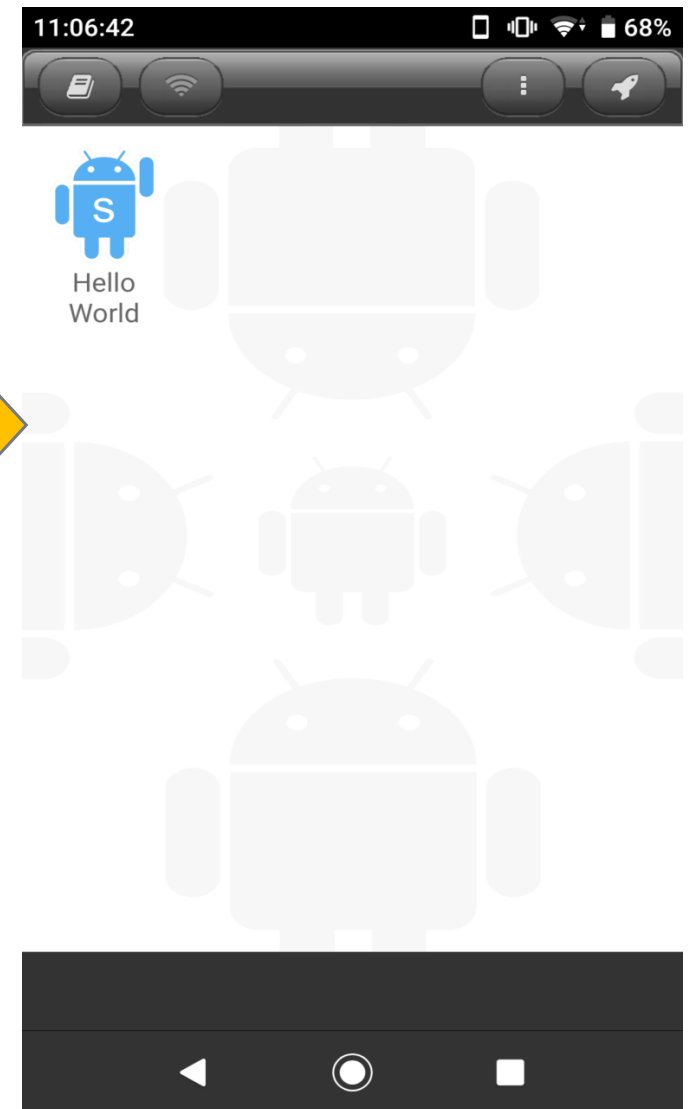
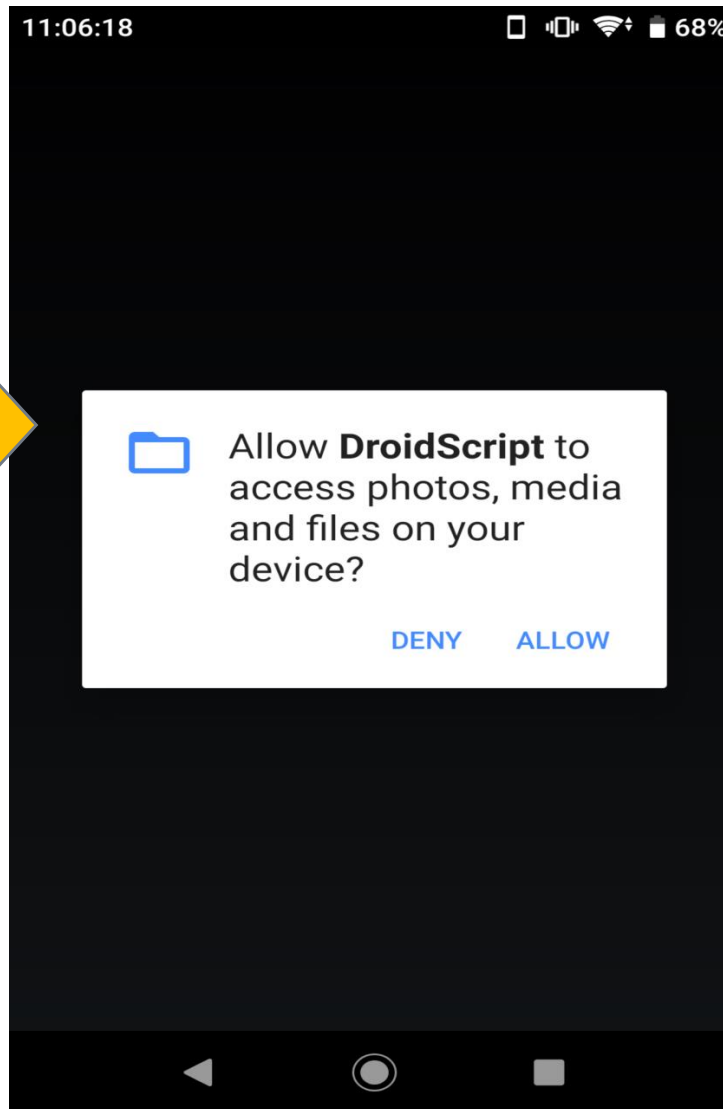
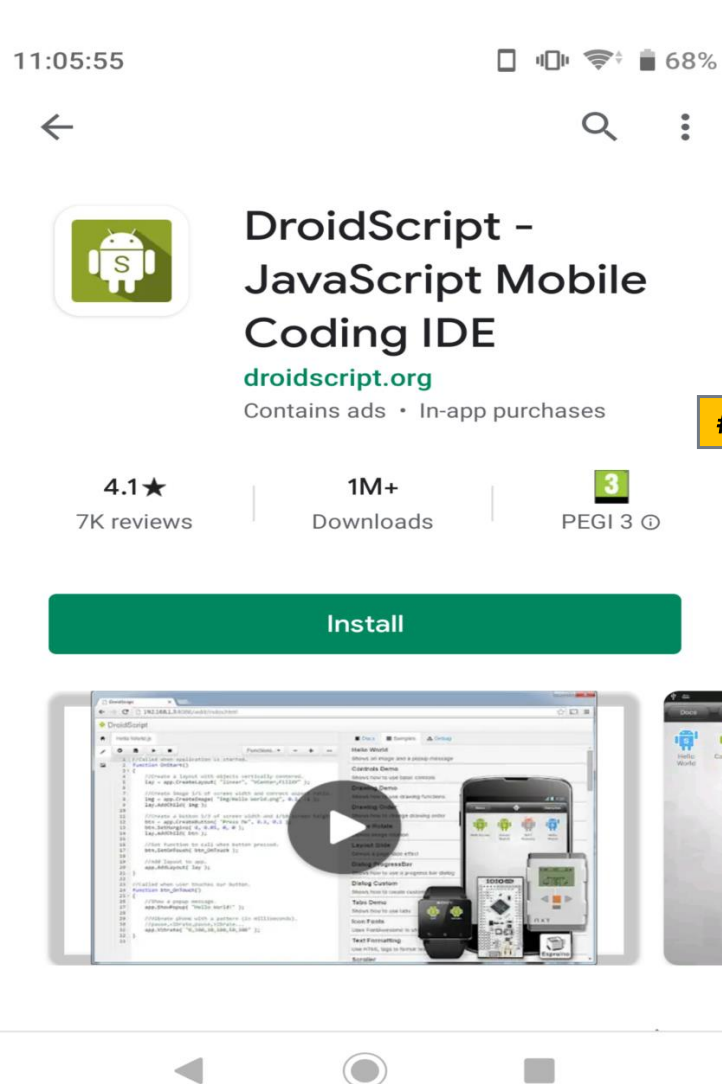
You can literally code anywhere with DroidScript as it's not cloud based and doesn't require an internet connection!

Unlike other development tools which take hours to install and eat up gigabytes of disk space, you can install DroidScript start using it within 30 seconds!

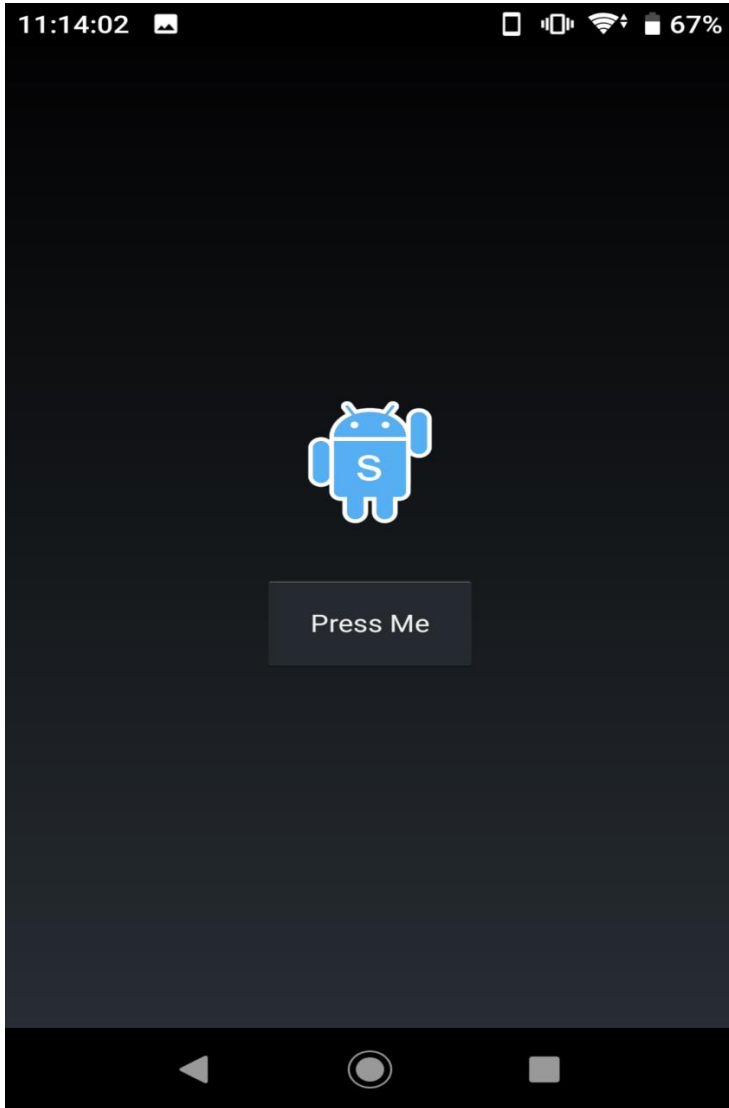
Website: <http://droidscript.org/>



Install the tool in Your Google-Android Device – steps (i.e. from: Google Play Store)

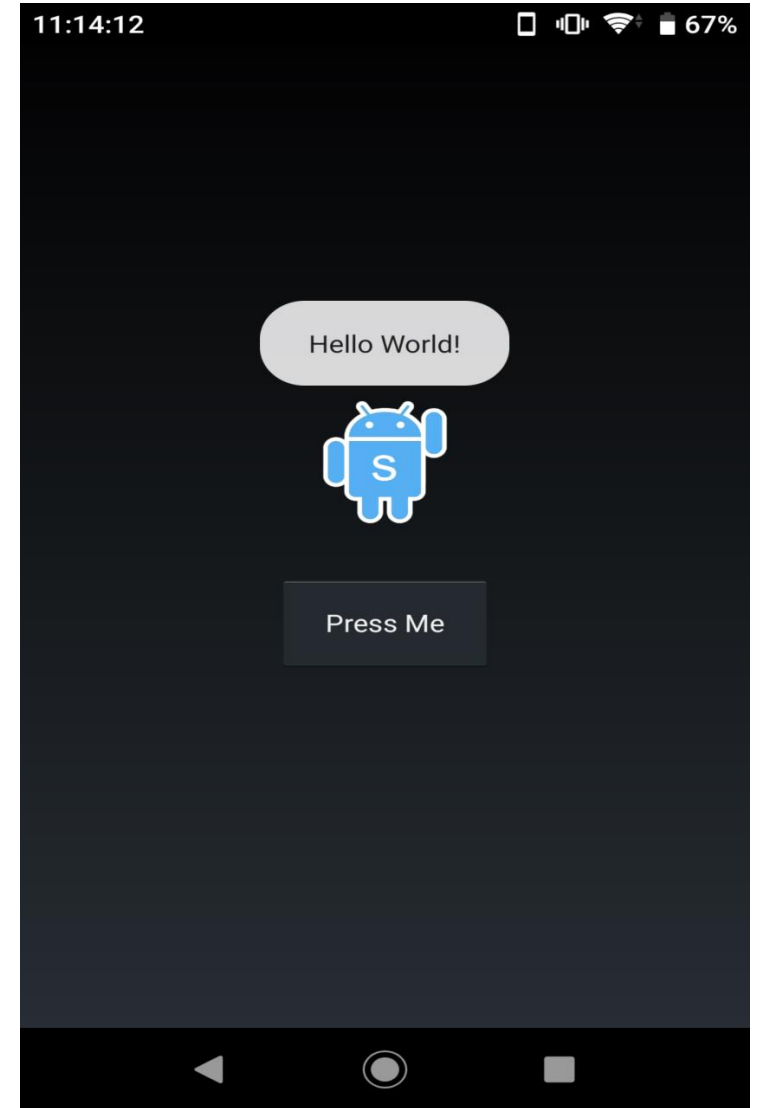


Now you can run – Basic world-famous example: “Hello World”

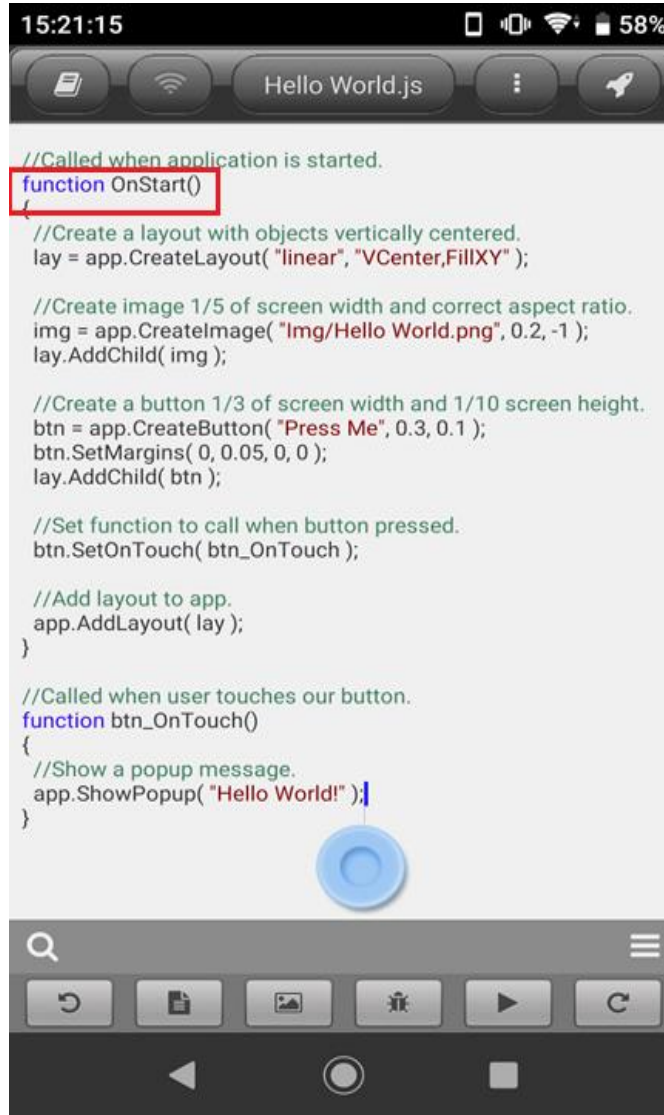
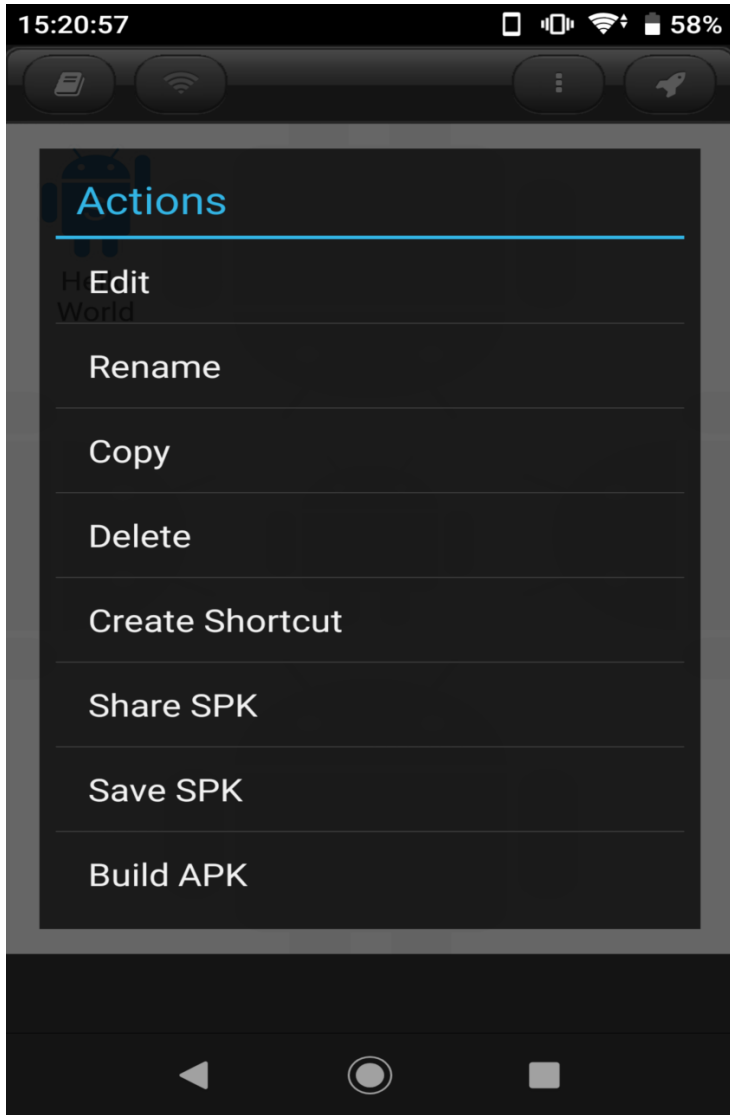


By click on the button named “Press Me”, at your Google-Android mobile phone.

Then your first Google-Android app is running and working fine!



“Hello World” app: how does it work? Let us examine the code for 1st time:



This code example for “HelloWorld” app, is based on JavaScript coding language.

JavaScript is easy to learn for any programmer and for any age.

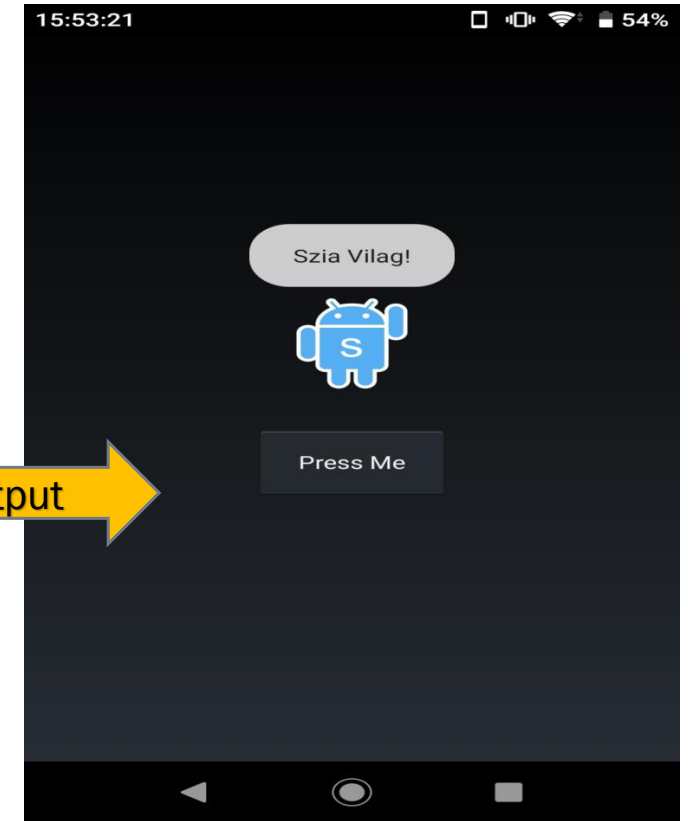
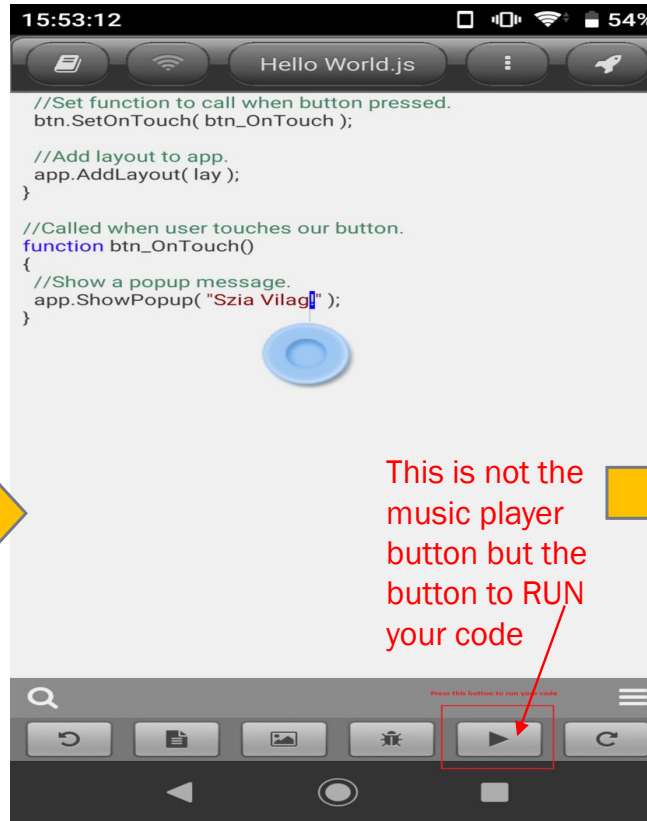
In the past until now is used for Web-Site HTML programming, but now for mobile technology also can be used to create Mobile Apps. Let us [learn our first rules here with DroidScript coding:](#)

- 1) *DroidScript JavaScript apps always start with a function called “onStart ()”*
- 2) *To comment your coding in JavaScript, always use this character “//” like this:
// Give your code a meaning...*
- 3) *“app” is a global variable for now!*
- 4) *All DroidApp code file names ends with a “js” → JavaScript code file*

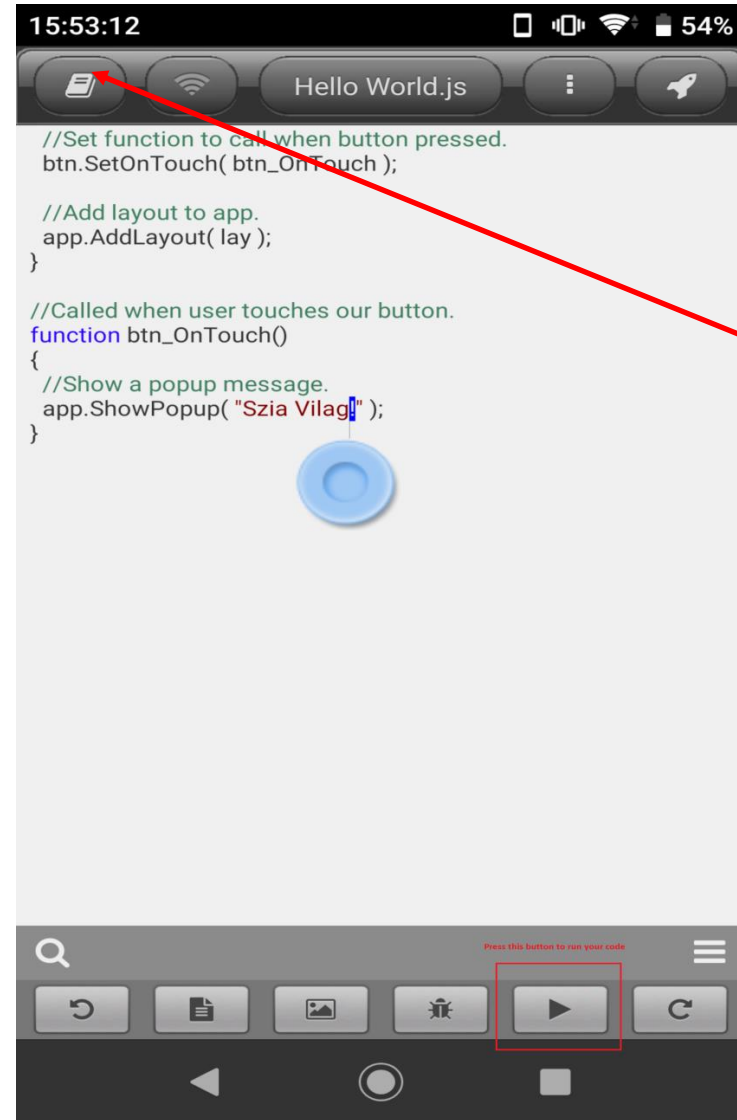
“Hello World” app: how does it work? Continue...



- With the help of “app” global variable, you can create a new Button, a new Image and a new screen layout, but let us keep it simple for now, since we don’t know yet how to code in JavaScript!
- Example to do some fun, change text of “Hello World!” to something funnier like “Szia Vilag!”, in your language, it can be Hungarian or Arabic, just try to change this code line statement and run the DroidApp again: `app.ShowPopup(“Szia Vilag!”); // check the results now`



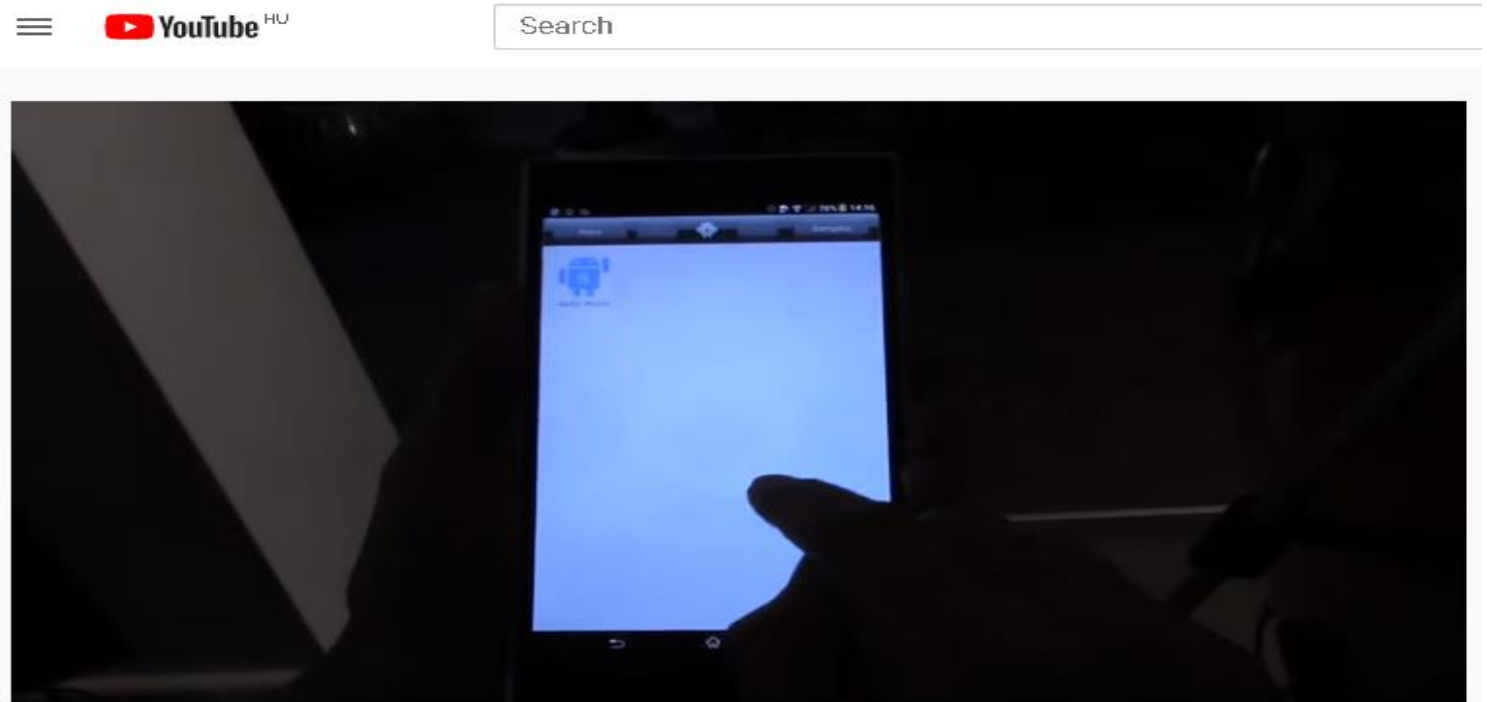
10 minutes YouTube video to learn the same thing from original Author



<https://youtu.be/v2HVARMz6B0>

If you don't understand anything from video, no worries! I will teach you how to code in DroidScript too!

The same video link you can reach it, using top bar button, as indicated here with red arrow.



Top bar-buttons: How to use it?



Documentation button (you will find here, video tutorial YouTube, Introduction to tool, Reference to complete tool and its language, JavaScript quick reference and Plugins reference)



Examples/Demos button: important examples from beginner level to more advanced ones (really good!)

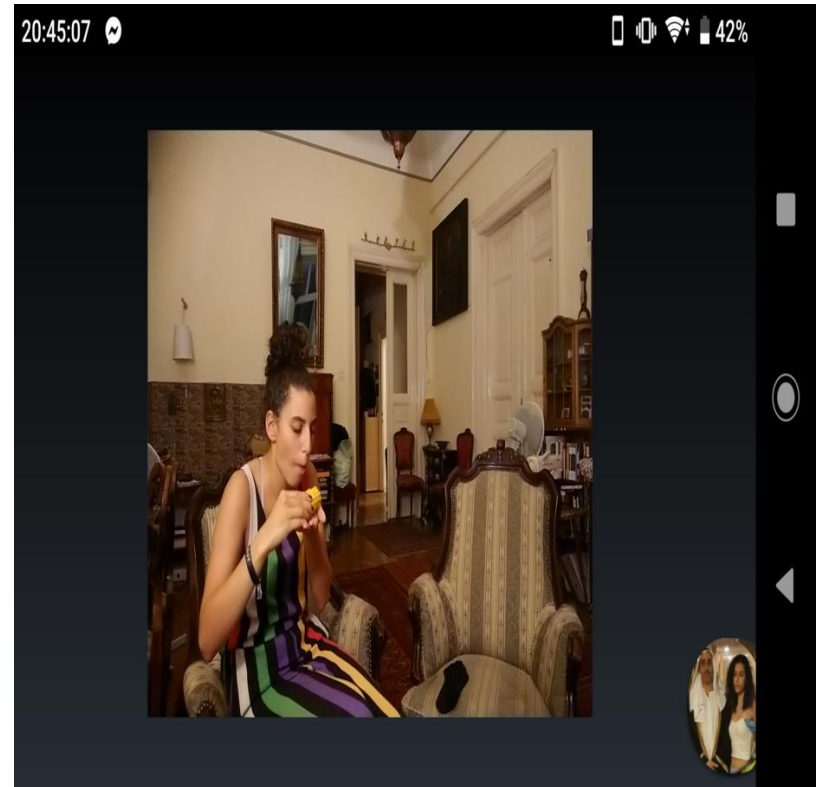
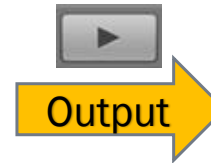
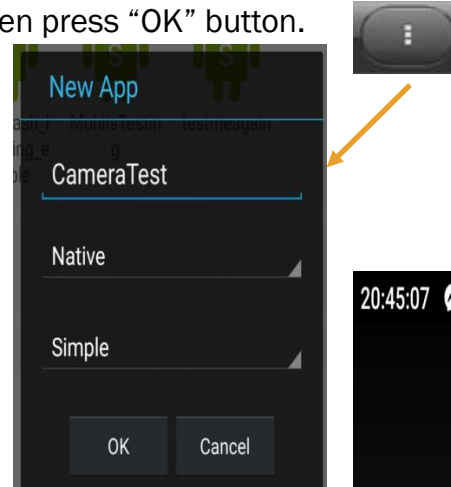


Menu button (you will find here, New code to create or app, tool Settings, News, About and Demos)

Another example: How to use the Camera from DroidScript code?

- Let us create a new code named "CameraTest.js", how to do that? Very simple, by clicking on "three vertical dots" menu and New command. No need to type the .js file extension, only name is needed. Then press "OK" button.
- From there we can create a new Mobile App code script.
- Below a sample code script and then you can easily, just Run it or Play it:

```
//Called when application is started:  
function OnStart(){  
  
    //Fix orientation to landscape since  
    //most phones cameras work this way.  
    app.SetOrientation("Landscape");  
  
    //Create horizontal layout that fills the screen.  
    lay = app.CreateLayout("Linear", "Horizontal,FillXY,VCenter");  
  
    //Create camera view control:  
    cam = app.CreateCameraView(0.6, 0.8);  
  
    cam.SetMargins(0, 0, 0.04, 0);  
    lay.AddChild(cam);  
  
    //Add main layout to app:  
    app.AddLayout(lay);  
    cam.StartPreview();  
}
```



Until now, what we can understand from two Code Examples (rules)?

1. When creating a new Google-Android DroidScript mobile app, we need to apply these basic rules to make your coding successful and your mobile app working fine (i.e. without crashing), hence these are the **coding screen rules**:
2. Always use “app” global variable to create screen Layout (i.e. it can be horizontally screen layout like landscape or vertical one. We will say more examples about available screen mobile layouts).

Code example:

```
app.SetOrientation( "Landscape" );  
lay = app.CreateLayout( "linear", "VCenter, FillXY" );
```

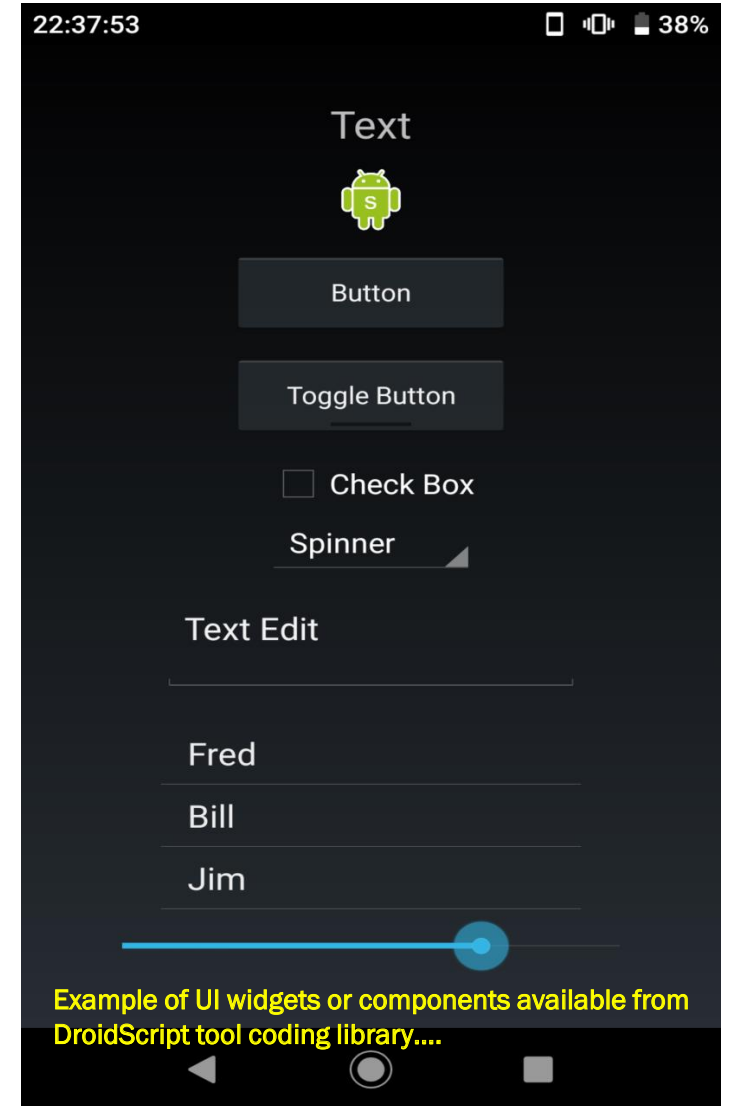
3. Create the screen components (i.e. child objects), they call it Widgets or UI components (User Interface), for example: UI label, button, text area, chart, image, list, progress-bar, etc. Code example:

```
btn = app.CreateButton( "Click-me", 0.5, 0.1 );  
lay.AddChild( btn );
```

4. Add above UI components (Widgets) to your created new layout. Code example: `app.AddLayout(lay);`
5. All above points (1, 2, 3, 4) under function named “`OnStart()`”
6. Then later you can add screen click events, what we call it in programming: **event-handling**. Basically responding to end-user clicks and behaviour (main code logic).

Code example:

```
function btn_onTouch() { ..... }
```

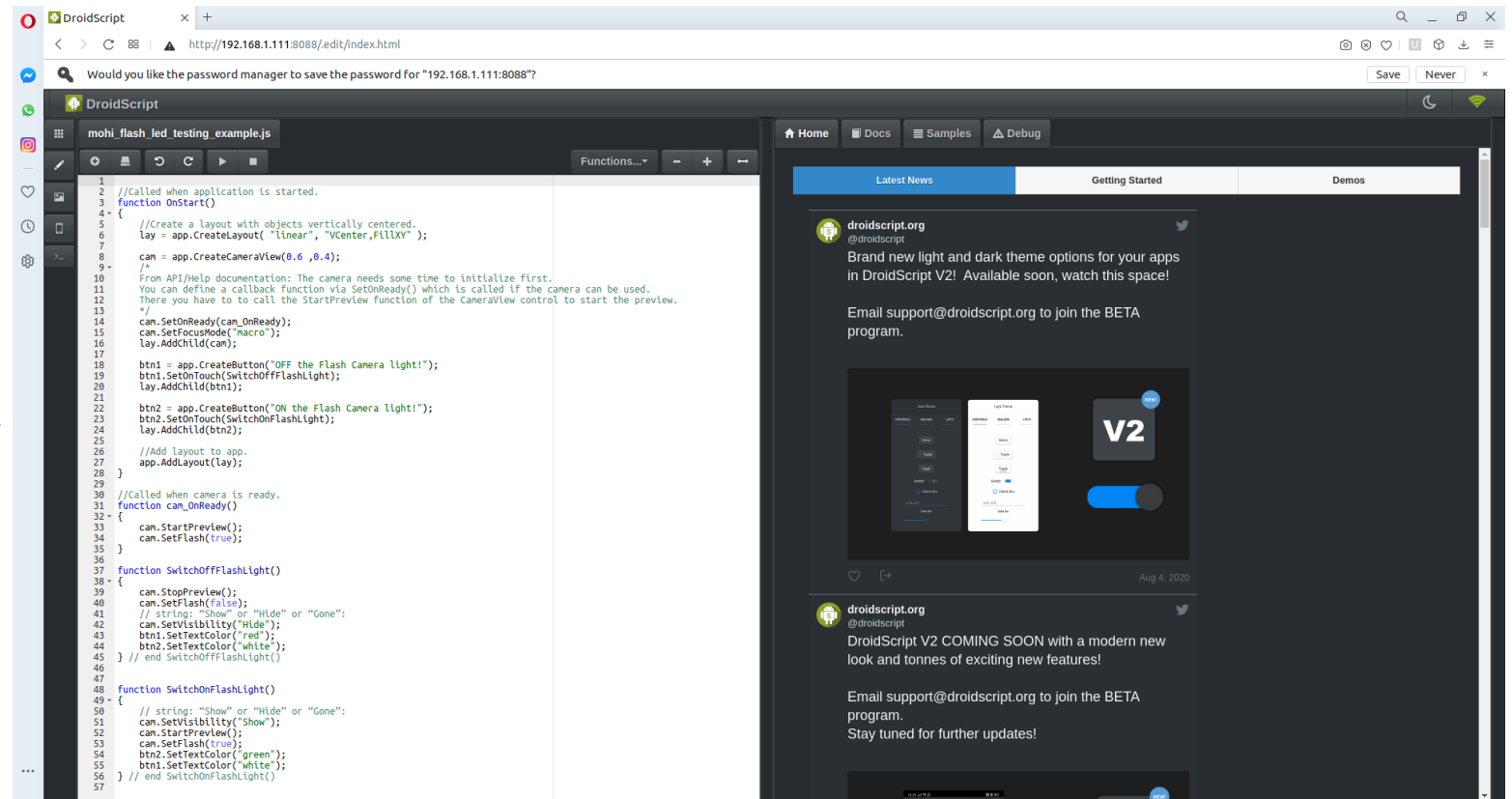
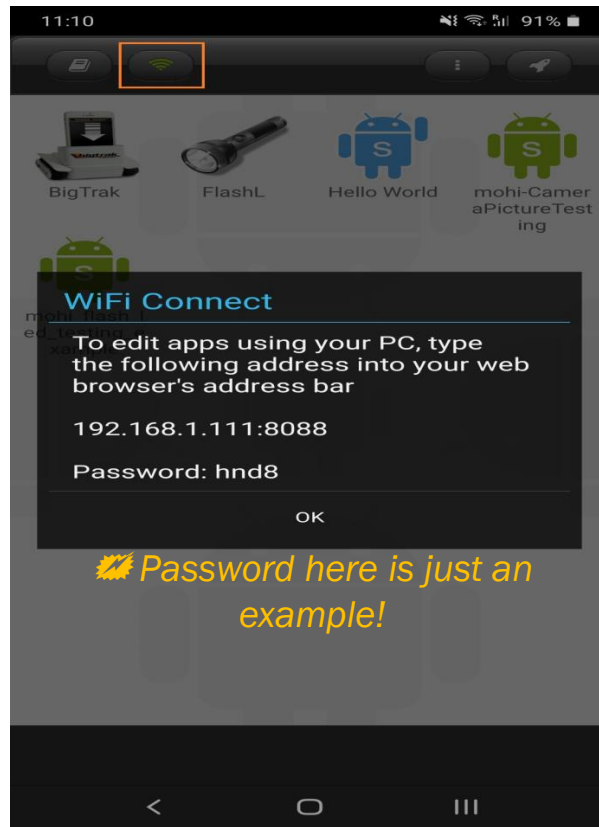


Where to go from here?

- Learn more code examples like, how to use Camera, Bluetooth, WIFI, etc. controls. I will include more examples to do step by step Google Android-app coding, in the next slides.
- Learn JavaScript language either from DroidScript tool itself: Documentation part has a very good introduction to JavaScript,
 - or from here: <http://droidscript.org/javascript/Basics.html> ,
 - or from here: <https://www.w3schools.com/js/> ,
 - or from here: <https://javascript.info/intro>
- See more code DEMOS from tool itself and learn how smart people had created these smart examples/Mobile apps.
- Read, code and practice, practice and practice.

Better way for Code Editing – Browser based IDE (Integrated Development Environment)

- Your mobile device is normally a small size, so to code better; DroidScript made a browser-based coding possibility easier and smoothly to run you coded apps from a typical browser like Google Chrome, Mozilla Firefox, Ms-Edge or Opera. *How to do that? Follow below screen steps:*



Better way for Code Editing – Browser based IDE (Integrated Development Environment) – **continue...**

- With the Play button, you can run your code (app) and with Stop button, you can stop your app. Each time you run your app, after finish your code testing, then you need to press the Stop button, as shown at the right screenshot:

The screenshot displays the DroidScript IDE interface. The top part shows a browser window with the URL `http://192.168.1.111:8088/edit/index.html`. Below the browser, there's a notification asking to save a password. The main IDE area is titled "DroidScript" and shows a code editor with the following JavaScript code:

```
1 //Called when application is started.
2 function onStart()
3 {
4     //Create a layout with objects vertically centered.
5     lay = app.CreateLayout( "linear", "VCenter,FillXY" );
6
7     can = app.CreateCameraView(0.6 ,0.4);
8     /*
9     From API/Help documentation: The camera needs some time to initialize first.
10    You can define a callback function via SetOnReady() which is called if the camera can be used.
11    There you have to to call the StartPreview function of the CameraView control. to start the preview.
12    */
13    can.SetOnReady(can_OnReady);
14    can.SetFocusMode("macro");
15    lay.AddChild(can);
16
17    btn1 = app.CreateButton("OFF the Flash Camera light!");
18    btn1.SetOnTouch(SwitchOffFlashLight);
19    lay.AddChild(btn1);
20
21    btn2 = app.CreateButton("ON the Flash Camera light!");
22    btn2.SetOnTouch(SwitchOnFlashLight);
23    lay.AddChild(btn2);
24
25    //Add layout to app.
26    app.AddLayout(lay);
27 }
28
29 //Called when camera is ready.
30 function can_OnReady()
31 {
32     can.StartPreview();
33     can.SetFlash(true);
34 }
35
36
37 function SwitchOffFlashLight()
38 {
39     can.StopPreview();
40     can.SetFlash(false);
41     // string: "Show" or "Hide" or "Gone":
42     can.SetVisibility("Hide");
43     btn1.SetTextColor("red");
44     btn2.SetTextColor("white");
45 } // end SwitchOffFlashLight()
46
47
48 function SwitchOnFlashLight()
49 {
50     // string: "Show" or "Hide" or "Gone":
51     can.SetVisibility("Show");
52     can.StartPreview();
53     can.SetFlash(true);
54     btn2.SetTextColor("green");
55     btn1.SetTextColor("white");
56 } // end SwitchOnFlashLight()
57
```

The IDE interface includes a toolbar with buttons for Home, Docs, Samples, and Debug. On the right side, there's a social media feed showing tweets from @droidscript. The tweets mention "Brand new light and dark theme options for your apps in DroidScript V2!" and "DroidScript V2 COMING SOON with a modern new look and tonnes of exciting new features!". The tweets also include a link to `support@droidscript.org` to join the BETA program.

Common buttons for the Code Editor – Browser-based IDE



- You here browser other Apps you have created or used from Demo/Sample available smart apps.
- Code text JavaScript editor: code completions, syntax highlighter, etc.
- Assets: you can use external resources for your code like images, sounds, HTML, others.
- Last two buttons are for the Premium paid version. It's really worth paying for this tool and support them!

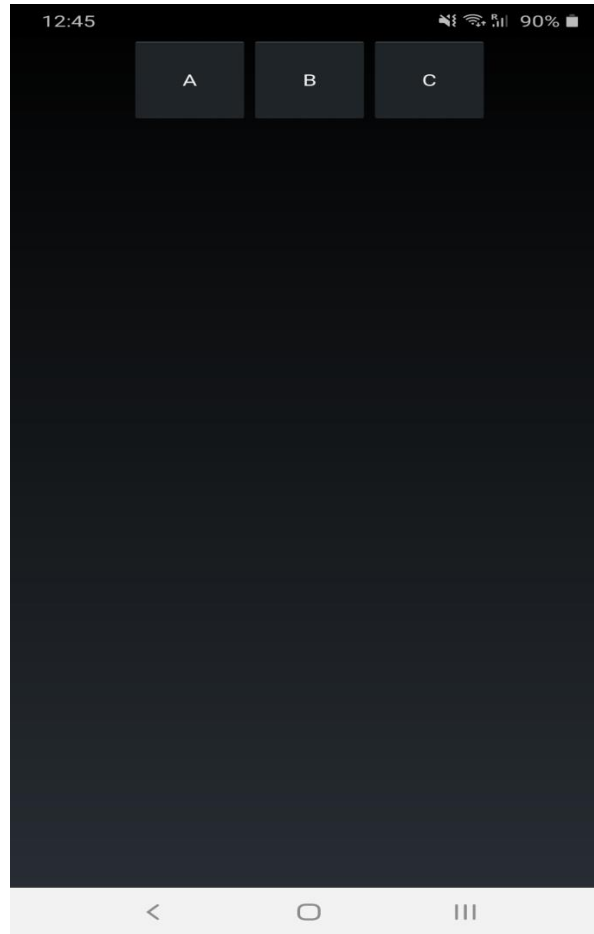


- Create a new code script file, basically new App in DroidScript tool.
- Save your work or you can press <Ctrl-s>, from your notebook keyboard.
- Undo: during editing you can undo the last code typed action or press <Ctrl-z> , from your notebook keyboard.
- Redo: during editing you can Redo the last code typed action or press <Ctrl-y> , from your notebook keyboard.
- Run your app code.
- Stop running your app code.

***Then let us do more examples using
Browser-based IDE...***

Layout example (Horizontal UI layout):

```
function onStart () {  
  
    lay = app.CreateLayout("Linear", "Horizontal,FillXY");  
  
    btnA = app.CreateButton("A", 0.2, 0.1);  
    lay.AddChild(btnA);  
  
    btnB = app.CreateButton("B", 0.2, 0.1);  
    lay.AddChild(btnB);  
    btnC = app.CreateButton("C", 0.2, 0.1);  
  
    lay.AddChild(btnC);  
    app.AddLayout(lay);  
}
```



✦ **CreateButton()**: you can create a simple clickable button, but DroidScript code library has a vast range of controls, UI components, Camera, Bluetooth, etc. to let you create a small app in hours or even minutes...

✦ You can display UI components/controls in a vertical layout-based or in a horizontal way (i.e. what we call it in programming: alignment or layouting), let us see how you can code this using this tool.

- Layouts are container objects (i.e. like in HTML DIV tag) which are used to visually organize graphical objects (controls), such as text, buttons and images on the screen, example:

```
layRef = app.CreateLayout( type, options )
```

There are 3 types of layout: "Linear", "Absolute" and "Frame".

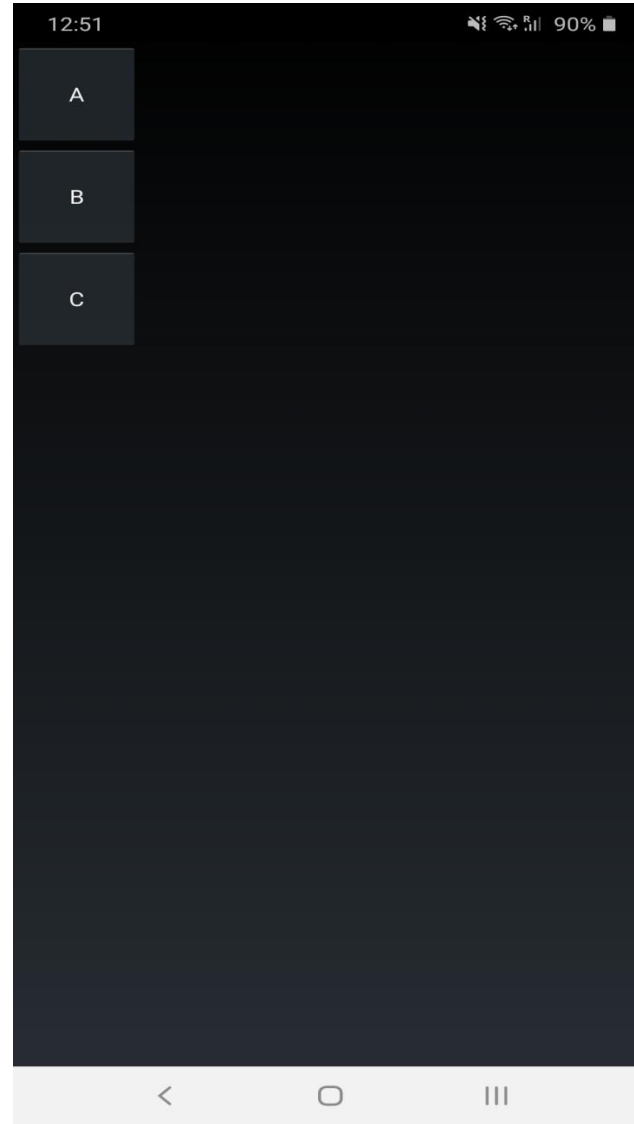
Options possible values:

"Left", "Top", "Right", "Bottom", "Center", "H/VCenter", "Wrap", "Horizontal", "Vertical", or "FillX/Y".

- Here we will experience and practice the "Linear" UI layout, rest you can do it by yourself.

Layout example (Vertical UI layout):

```
function OnStart () {  
  
    lay = app.CreateLayout("Linear", "Vertical");  
  
    btnA = app.CreateButton("A", 0.2, 0.1);  
    lay.AddChild(btnA);  
  
    btnB = app.CreateButton("B", 0.2, 0.1);  
    lay.AddChild(btnB);  
  
    btnC = app.CreateButton("C", 0.2, 0.1);  
    lay.AddChild(btnC);  
  
    app.AddLayout (lay);  
}
```



✍ You can display UI components/controls in a vertical layout-based or in a horizontal way (i.e. what we call it in programming: alignment or layouting), let us see how you can code this using this tool.

- Layouts are container objects (i.e. like in HTML DIV tag) which are used to visually organize graphical objects (controls), such as text, buttons and images on the screen, example:

```
layRef = app.CreateLayout( type, options )
```

There are 3 types of layout: “Linear”, “Absolute” and “Frame”.

Options possible values:

“Left”, “Top”, “Right”, “Bottom”, “Center”, “H/VCenter”, “Wrap”, “Horizontal”, “Vertical”, or “FillX/Y”.

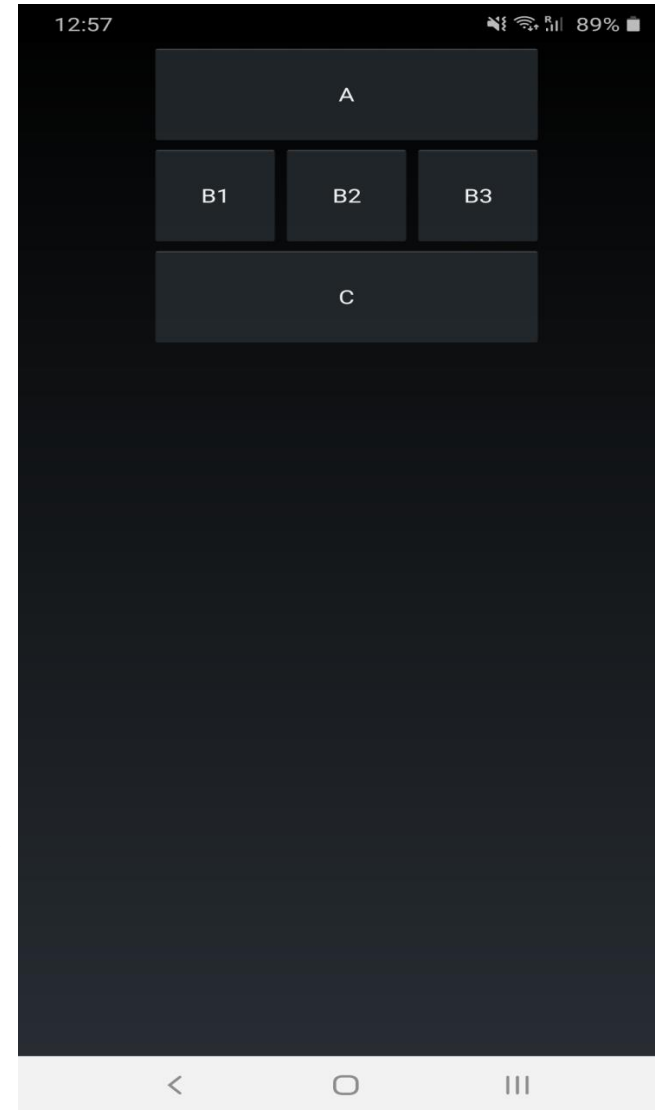
- Here we will experience and practice the “Linear” UI layout, rest you can do it by yourself.

Common Layout example to use

```
function OnStart () {  
  
    layVert = app.CreateLayout("Linear", "Vertical,FillXY");  
  
    btnA = app.CreateButton("A", 0.6, 0.1);  
    layVert.AddChild(btnA);  
  
    layHoriz = app.CreateLayout("Linear", "Horizontal");  
    layVert.AddChild(layHoriz);  
  
    btnB1 = app.CreateButton("B1", 0.2, 0.1);  
    layHoriz.AddChild(btnB1);  
  
    btnB2 = app.CreateButton("B2", 0.2, 0.1);  
    layHoriz.AddChild(btnB2);  
  
    btnB3 = app.CreateButton("B3", 0.2, 0.1);  
    layHoriz.AddChild(btnB3);  
    btnC = app.CreateButton("C", 0.6, 0.1);  
  
    layVert.AddChild(btnC);  
  
    app.AddLayout(layVert);  
}
```

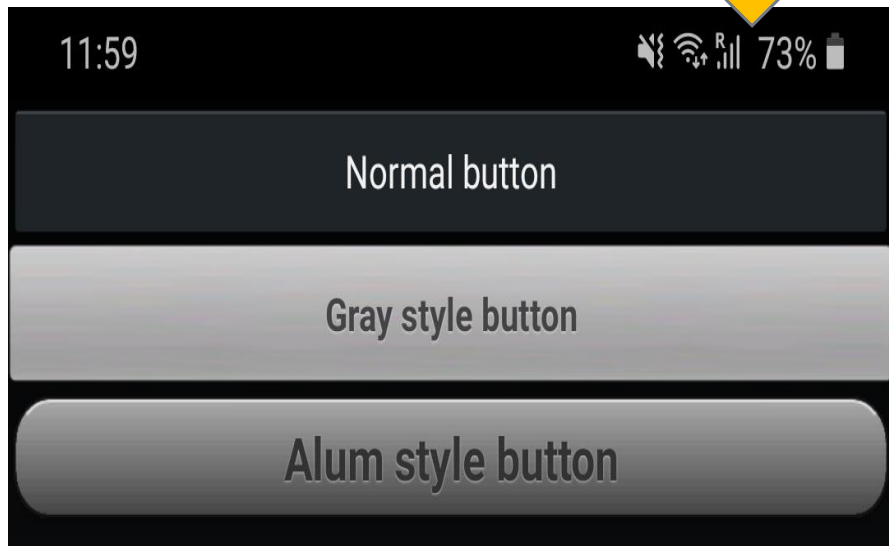


(Combined UI layout):



Buttons example (with Different-Style):

```
function OnStart () {  
  
    lay = app.CreateLayout("Linear", "Vertical,FillXY");  
  
    b1 = app.CreateButton("Normal button", -1, -1, "FillX");  
    lay.AddChild(b1);  
  
    b2 = app.CreateButton("Gray style button", -1, -1, "FillX,Gray");  
    lay.AddChild(b2);  
  
    b3 = app.CreateButton("Alum style button", -1, -1, "FillX,Alum");  
    lay.AddChild(b3);  
  
    app.AddLayout(lay);  
}
```



- Buttons created with the CreateButton method are used for interacting with the app (i.e. to interact with mobile end-user):

btn = app.CreateButton(text, width, height, options)

This is done by calling a **callback function** after clicking on the button. You can specify such a callback function with the SetOnTouch Method:
btn.SetOnTouch(btn_OnTouch);

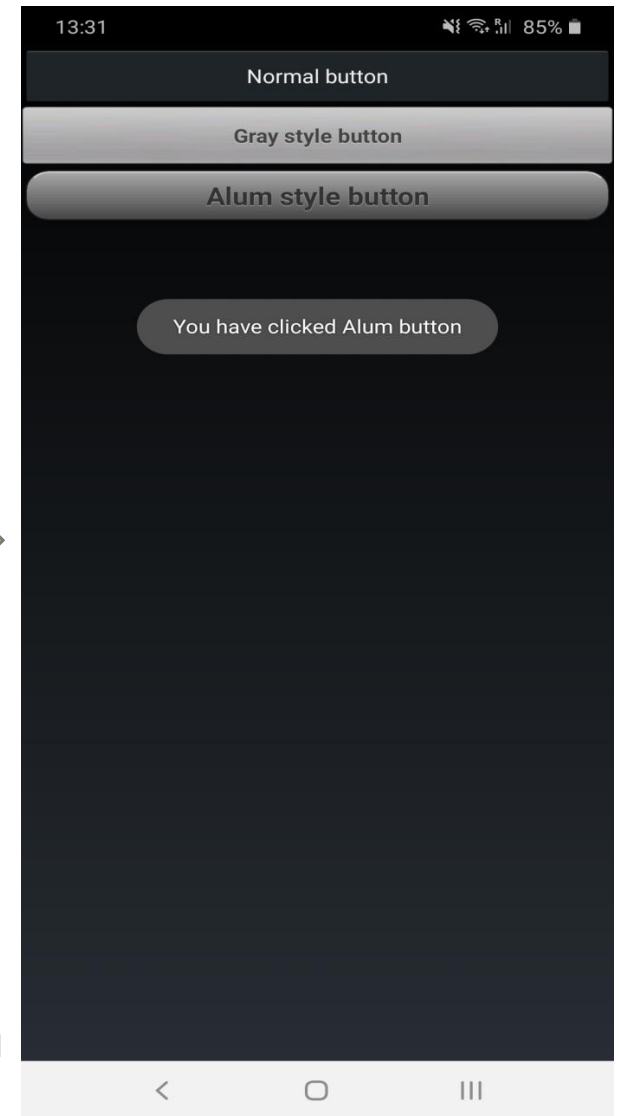
```
function btn_OnTouch() {  
    // your button click logic code goes here  
}
```

- width: possible values: 0..1 like 0.5, 1, -1, 0.7, etc.
- height: possible values: 0..1 like -1, 0.5, 0.1, etc.

Callback function definition (source: [Wikipedia](#)): In [computer programming](#), a **callback**, also known as a "call-after function", is any [executable code](#) that is passed as an [argument](#) to other code; that other code is expected to *call back* (execute) the argument at a given time. This execution may be immediate as in a **synchronous callback**, or it might happen later as in an **asynchronous callback**. [Programming languages](#) support callbacks in different ways

Buttons example with Touch/Click user interaction (i.e. Callback function):

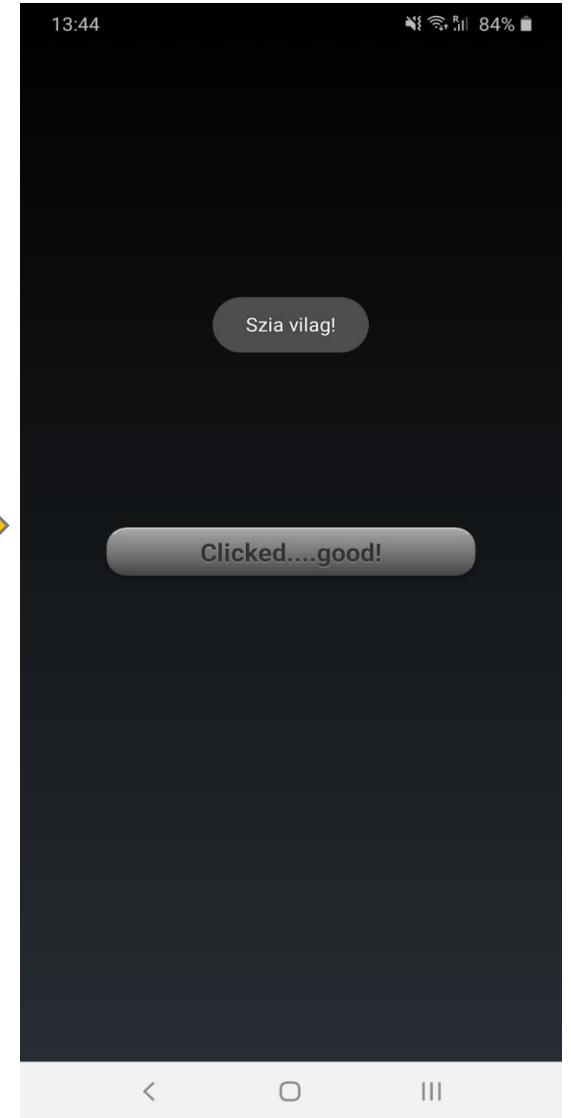
```
function OnStart () {  
  
    lay = app.CreateLayout("Linear", "Vertical,FillXY");  
    b1 = app.CreateButton("Normal button", -1, -1, "FillX");  
    b1.SetOnTouch(YouTouchedMe1);  
    lay.AddChild(b1);  
  
    b2 = app.CreateButton("Gray style button", -1, -1, "FillX,Gray");  
    b2.SetOnTouch(YouTouchedMe2);  
    lay.AddChild(b2);  
  
    b3 = app.CreateButton("Alum style button", -1, -1, "FillX,Alum");  
    b3.SetOnTouch(YouTouchedMe3);  
    lay.AddChild(b3);  
  
    app.AddLayout(lay);  
}  
  
function YouTouchedMe1 () {  
    app.ShowPopup("You have clicked Normal button");  
}  
  
function YouTouchedMe2 () {  
    app.ShowPopup("You have clicked Gray button");  
}  
  
function YouTouchedMe3 () {  
    app.ShowPopup("You have clicked Alum button");  
}
```



✍ **ShowPopup():** inform or show any text message to mobile end-user. Used commonly in many examples and important method to know about it.

Another Button example with Touch/Click user interaction:

```
function onStart () {  
  
    /* Create a linear basic layout, linear means either vertical or horizontal type  
    orientation for UI controls, most common one used in DroidScript JavaScript coding,  
    example from API documentations (help): Alignment can be "Left", "Right", "Bottom" or  
    "VCenter", which means vertically centred.*/  
    lay = app.CreateLayout("Linear", "VCenter,FillXY");  
  
    /* From API/Help documentation: You can customize the visual look in many ways by  
    either using one of the existing style options "Alum" or "Gray", or you can use  
    "Custom" and define a custom style with the SetStyle() method: also: Setting the width  
    and height to -1 whilst using the "FillX" option will allow it to fill the layout  
    width: */  
  
    // Try this below first to see it:  
    //btn = app.CreateButton("Alum button style", -1, 1, "FillX,Alum");  
  
    // Then try this button style:  
    btn = app.CreateButton("Alum button style", 0.7, -1, "Alum");  
  
    // Tell button which function/action to call to handle your needed UI logic:  
    btn.SetOnTouch(GreetingsFunction);  
  
    // Any controls/components must be added to layout,otherwise will not appear to user:  
    lay.AddChild(btn);  
    app.AddLayout(lay);  
}  
  
function GreetingsFunction () {  
    app.ShowPopup("Szia vilag!");  
    btn.SetText("Clicked....good!");  
}
```



✍ **SetText():** You can set the caption label text of any button.

Make an Image-Clickable like a Button example:

```
// Called when application is started.
function OnStart() {

    // Create a layout with objects vertically centred.
    lay = app.CreateLayout("linear", "VCenter,FillXY");

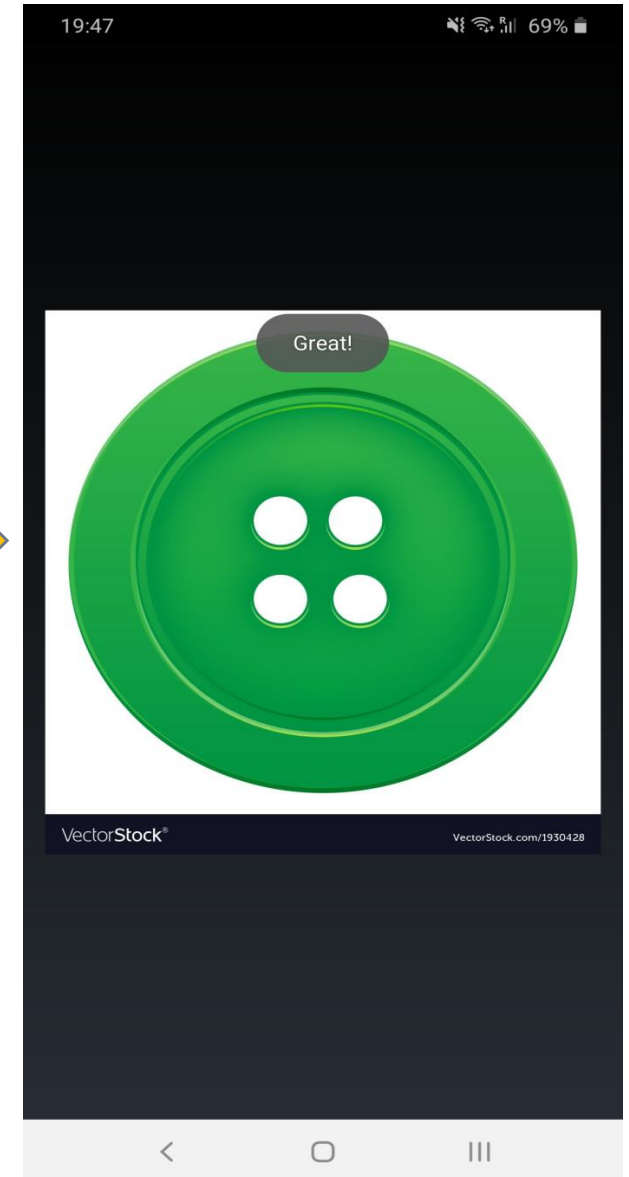
    // Create an image and add it to the layout.
    img = app.CreateImage("Img/button_image_icon_downloaded.jpg");

    // Add a Callback function to handle when you touch the image:
    img.SetOnTouch(TouchMe);

    lay.AddChild(img);

    //Add layout to app:
    app.AddLayout(lay);
}

function TouchMe() {
    app.ShowPopup("Great!");
}
```



How to add the image as a resource to your app?
Click on Assets button and upload needed images (i.e. resource) to your mobile app and you can reach it from this storage mobile path: "/img/..."

Flash-Led example (i.e. Torch Switch ON/OFF from your device):

```
//Called when application is started.
function onStart() {

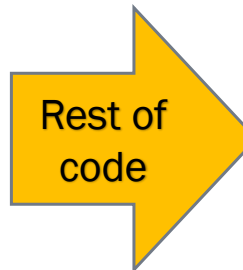
    //Create a layout with objects vertically centred:
    lay = app.CreateLayout("linear", "VCenter,FillXY");

    cam = app.CreateCameraView(0.6 ,0.4);
    /* From API/Help documentation: The camera needs some time to
    initialize first. You can define a callback function via SetOnReady()
    which is called if the camera can be used. There you have to call
    the StartPreview function of the CameraView control to start the
    preview.
    */
    cam.SetOnReady(cam_OnReady);
    cam.SetFocusMode("macro");

    lay.AddChild(cam);
    btn1 = app.CreateButton("OFF the Flash Camera light!");
    btn1.SetOnTouch(SwitchOffFlashLight);
    lay.AddChild(btn1);

    btn2 = app.CreateButton("ON the Flash Camera light!");
    btn2.SetOnTouch(SwitchOnFlashLight);
    lay.AddChild(btn2);

    //Add layout to app:
    app.AddLayout(lay);
}
```



```
//Called when camera is ready to be used:
function cam_OnReady() {

    cam.StartPreview();
    //enable the Flash camera led as ON:
    cam.SetFlash(true);
}

function SwitchOffFlashLight() {

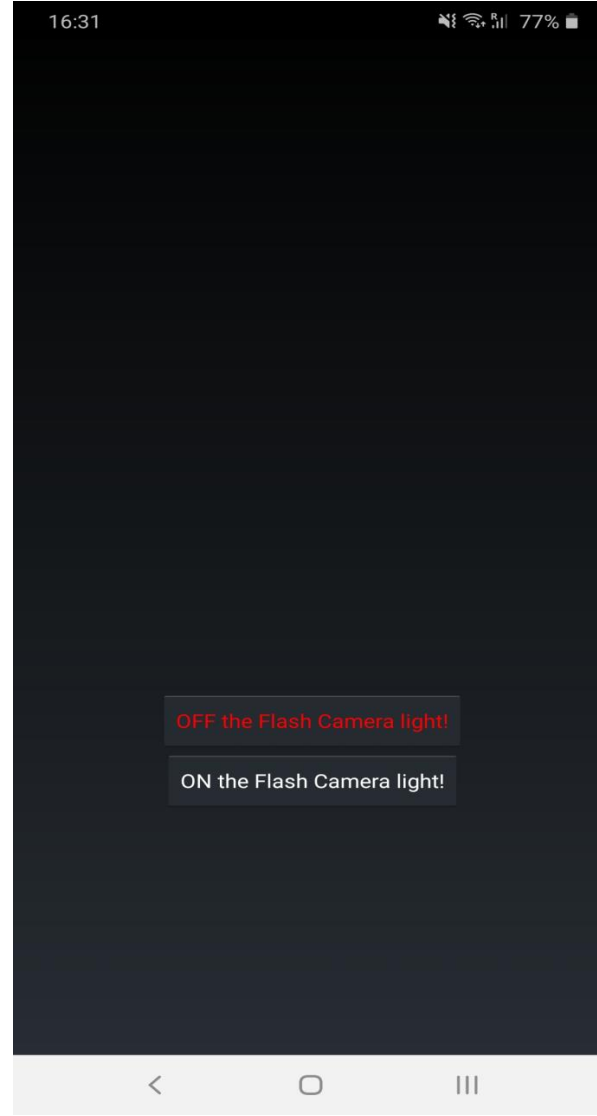
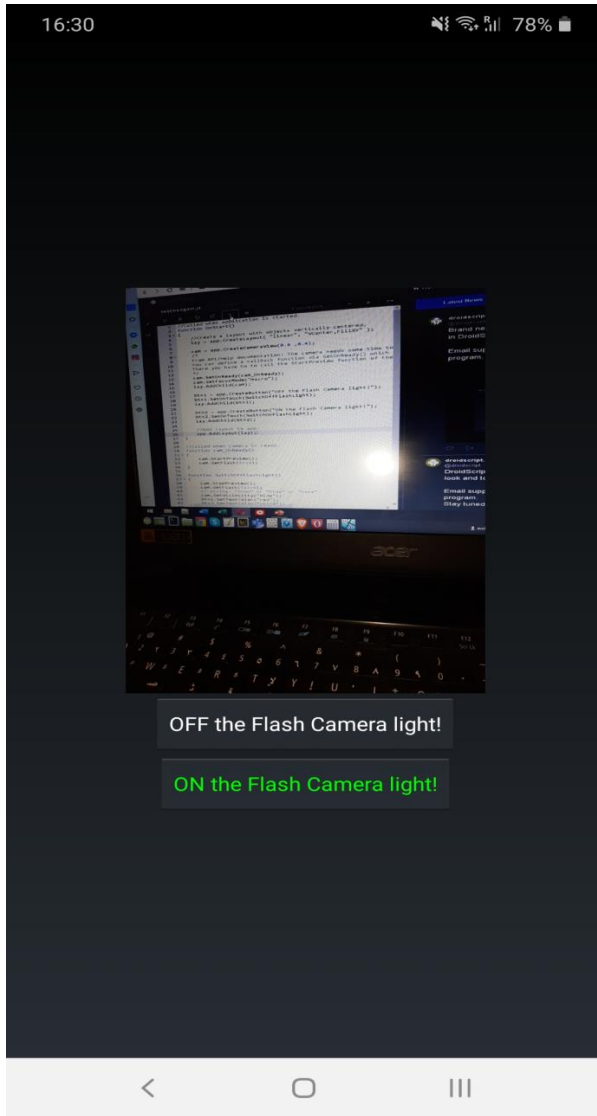
    cam.StopPreview();
    //disable the Flash camera led as OFF:
    cam.SetFlash(false);

    // string: "Show" or "Hide" or "Gone":
    cam.SetVisibility("Hide");
    btn1.SetTextColor("red");
    btn2.SetTextColor("white");
} // end SwitchOffFlashLight()

function SwitchOnFlashLight() {

    // string: "Show" or "Hide" or "Gone":
    cam.SetVisibility("Show");
    cam.StartPreview();
    cam.SetFlash(true);
    btn2.SetTextColor("green");
    btn1.SetTextColor("white");
} // end SwitchOnFlashLight()
```


Flash-Led example (i.e. Torch Switch ON/OFF from your device) – continue...



- You can comment your code using either “//” for one line comment or “/*...*/” for more than one line of comments, we call it in programming “block comment”.
- Another thing we can learn from this Camera code script, how to use Camera API instructions, to use its Flash led light as Torch.
- From Help documentation for Camera control, it’s documented as: **CameraViews** are used to access the device camera.

```
cam = app.CreateCameraView( width, height, options )
```

You can use the “Front” option to show the front camera instead of the default back camera.

The camera needs some time to initialize first.

You can define a callback function via **setOnReady** which is called if the camera can be used.

There you should call the **startPreview** function of the **CameraView** control to start the preview.

- ✍ width: possible values: 0..1 like 0.5, 1, -1, 0.7, etc.
- ✍ height: possible values: 0..1 like -1, 0.5, 0.1, etc.

The screenshot shows the DroidScript IDE with two panes. The left pane contains the source code for a JavaScript application named 'testmeagain.js'. The code includes an 'OnStart' function that creates a linear layout, a camera view, and two buttons. One button is for turning the flash off, and the other is for turning it on. There is also a 'cam_OnReady' function and a 'SwitchOffFlashLight' function. The right pane shows the documentation for the 'CreateCameraView' control, explaining that it is used to access the device camera and providing examples of how to use it, including setting width, height, and options, and how to handle the camera's ready state.

```
1 //Called when application is started.
2 function OnStart()
3 {
4     //Create a layout with objects vertically centered.
5     lay = app.CreateLayout( "linear", "vCenter,FillXY" );
6
7     cam = app.CreateCameraView(0.6 ,0.4);
8     /*
9     From API/Help documentation: The camera needs some time to
10    You can define a callback function via SetOnReady() which
11    There you have to to call the StartPreview function of the
12    */
13    cam.SetOnReady(cam_OnReady);
14    cam.SetFocusMode("macro");
15    lay.AddChild(cam);
16
17    btn1 = app.CreateButton("OFF the Flash Camera light!");
18    btn1.SetOnTouch(SwitchOffFlashLight);
19    lay.AddChild(btn1);
20
21    btn2 = app.CreateButton("ON the Flash Camera light!");
22    btn2.SetOnTouch(SwitchOnFlashLight);
23    lay.AddChild(btn2);
24
25    //Add layout to app.
26    app.AddLayout(lay);
27 }
28
29 //Called when camera is ready.
30 function cam_OnReady()
31 {
32     cam.StartPreview();
33     cam.SetFlash(true);
34 }
35
36 function SwitchOffFlashLight()
37 {
38     cam.StopPreview();
39     cam.SetFlash(false);
40     // string: "Show" or "Hide" or "Gone":
41     cam.SetVisibility("Hide");
42     btn1.SetTextColor("red");
43     btn2.SetTextColor("white");
44 }
```

Where to go from here?

- Learn more code examples like, how to use WIFI, Bluetooth, other controls. I will include more examples to do a step by step Google Android-app coding, in the next slides.
- Browse the documentation (Docs) for each control like Camera, Layout, GPS, etc.
- Read, code and practice, practice and practice.

Mobile Phone Contact – How to reach them? – Example

```
//Called when application is started:
function onStart(){

    lay = app.CreateLayout("linear", "VCenter,FillXY");
    btnChoose = app.CreateButton("Choose Phone", 0.5, 0.1);

    btnChoose.SetOnTouch(btnChoose_OnTouch);
    lay.AddChild(btnChoose);
    app.AddLayout(lay);
}

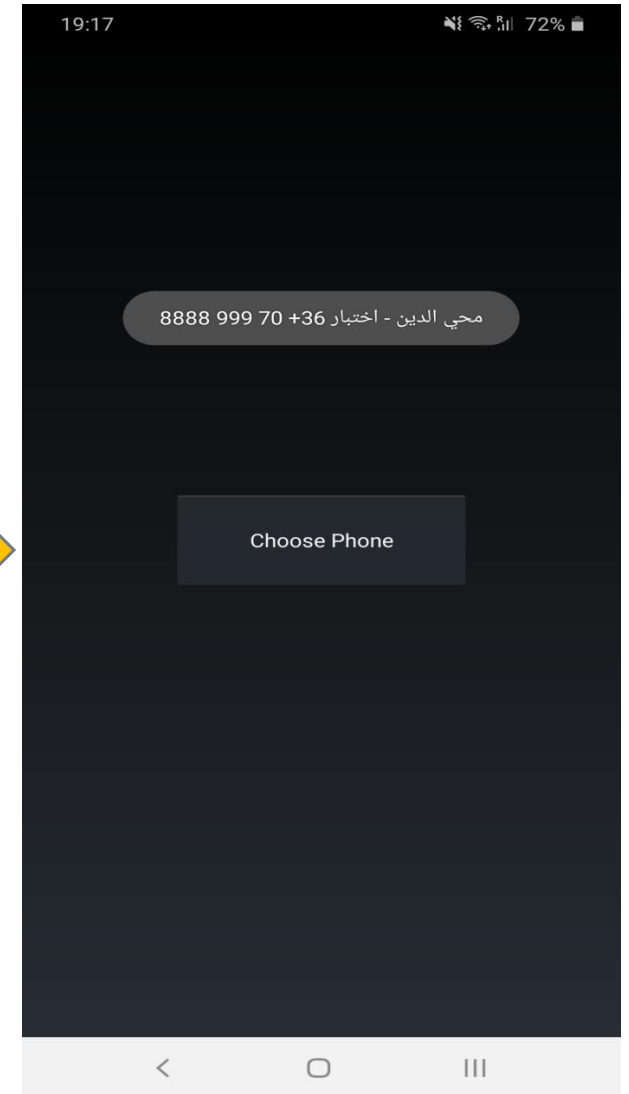
function btnChoose_OnTouch(){

    /* As per API docs, ChooseContact opens the Contacts app so that the user can
    select the name, or either the mobile phone no or e-mail address: */

    // Type here for ChooseContact can be either "Phone" or "Email":
    app.ChooseContact("Phone", OnPhoneChoose);
}

// Callback function:
function OnPhoneChoose(name, mobileNo){

    // "+" This is called text or string concatenations (text merge):
    app.ShowPopup(name + " " + mobileNo);
}
}
```



How to Capture a Camera Photo? – Example

```
function onStart() {
```

```
//Fix orientation to landscape since most phone cameras work this way:  
app.SetOrientation("Landscape");
```

```
//Create horizontal layout that fills the screen:  
lay = app.CreateLayout("Linear", "Horizontal,FillXY,VCenter");
```

```
// Set the app background to a specific repeated pattern:  
lay.SetBackground("/res/drawable/pattern_carbon", "repeat");
```

```
//Create camera view control as we done it in previous code samples:  
cam = app.CreateCameraView(0.6, 0.8);  
cam.SetOnReady(cam_OnReady);  
cam.SetMargins(0, 0, 0.04, 0);
```

```
lay.AddChild(cam);
```

```
//Create vertical layout on right for other UI stuff:  
layRight = app.CreateLayout("Linear", "Vertical");  
lay.AddChild(layRight);
```

```
//Create record button:  
btn = app.CreateButton("Capture", 0.2, 0.15, "Alum");  
btn.SetMargins(0, 0.06, 0, 0);  
btn.SetOnTouch(CaptureMe);  
layRight.AddChild(btn);  
//Add main layout to app:  
app.AddLayout(lay);
```

```
//Create a folder for photo saving at your mobile storage:  
recFolder = "/sdcard/photos_captured";  
app.MakeFolder(recFolder);  
}
```

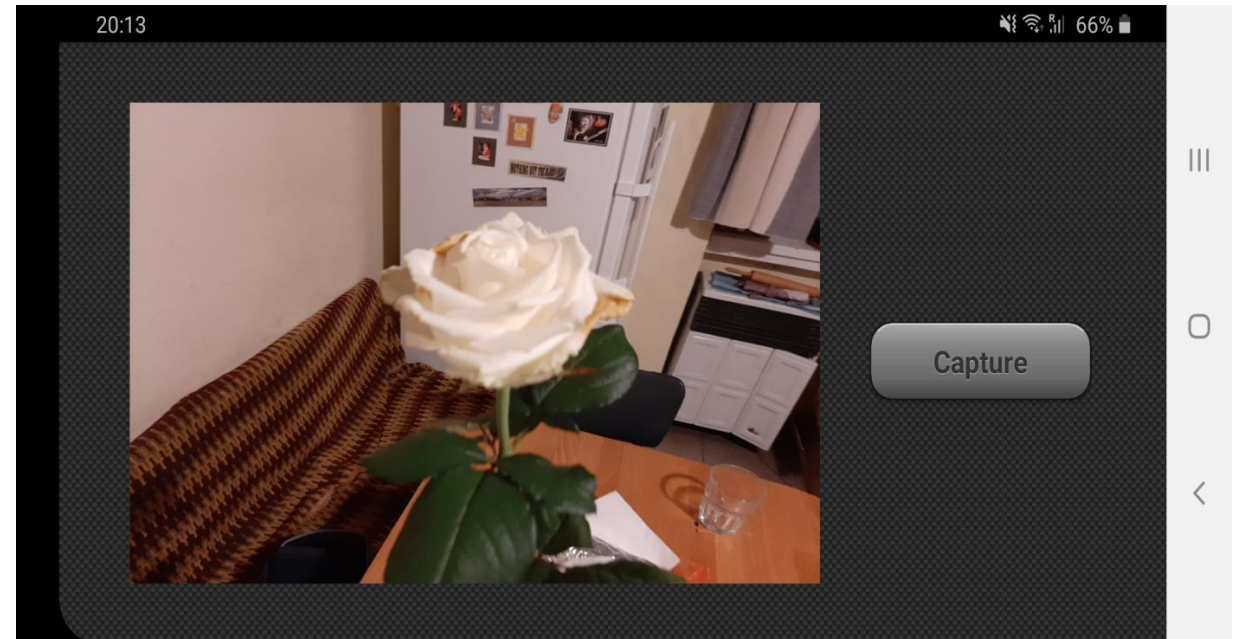
Rest of
code

```
//Called when camera is ready:  
function cam_OnReady() {
```

```
// start the camera preview mode:  
cam.StartPreview();  
} //Handle Capture photo button:
```

```
function CaptureMe() {
```

```
//Start recording and update button:  
cam.TakePicture(recFolder+"/testPhoto.jpg");  
}
```



How to use GPS – Locator/Location Position example

```
// Before running this app, switch ON your device Location:
function onStart () {

    // Use the GPS device location, this is how to use this GPS location API:
    loc = app.CreateLocator ("GPS,Network");

    // To get updates about your current position use the SetOnChange callback function:
    loc.SetOnChange (gpsLocation_OnChange);

    // Start the GPS position location:
    loc.Start();
    app.ShowPopup ("Locating your position...Make sure to switch ON your GPS location!");
}

// This callback function will run until you stop or terminate the app:
function gpsLocation_OnChange (pos) {

    var msg = pos.latitude + ", " + pos.longitude;

    app.ShowPopup ("Your location (Latitude, Longitude): " + msg);

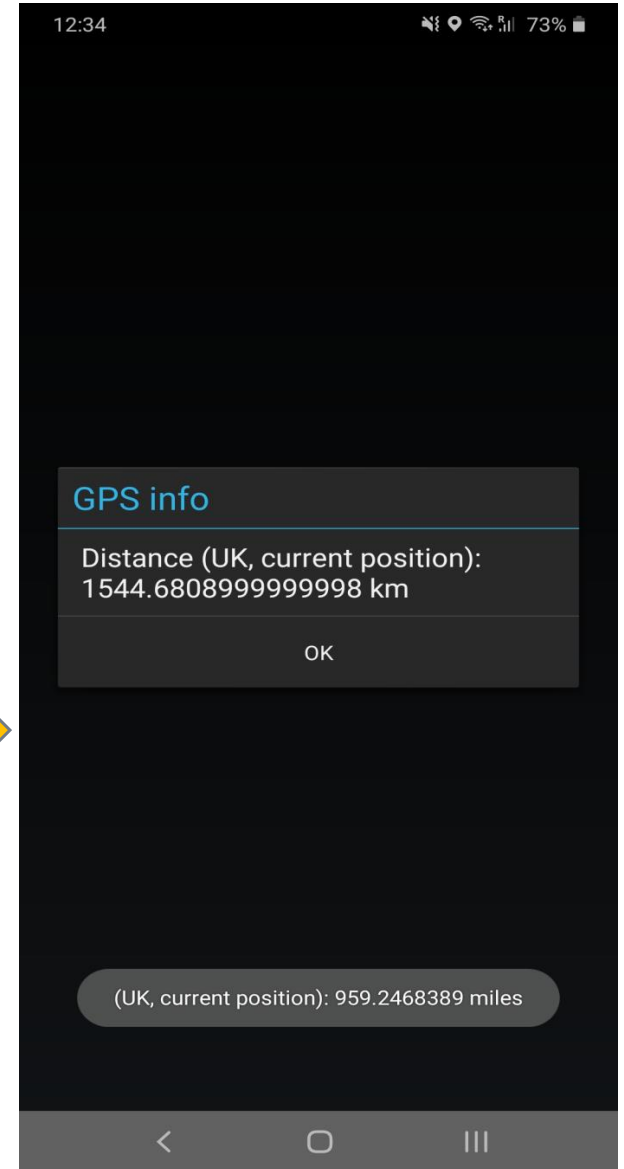
    // Measure distance between current location and UK:
    var distanceInMeters = loc.GetDistanceTo (52.3555177, -1.1743197);
    var distanceInKiloMeters = distanceInMeters / 1000; // distance in KM (number division)
    var distanceInMiles = distanceInKiloMeters * 0.621; // distance in Miles (multiplication)

    // Alert shows a simple text dialog with a colored title and a 'Ok' button to exit:
    app.Alert ("Distance (UK, current position): " + distanceInKiloMeters + " km", "GPS info");

    app.ShowPopup ("(UK, current position): " + distanceInMiles + " miles", "Bottom,Long");
}
```



Output

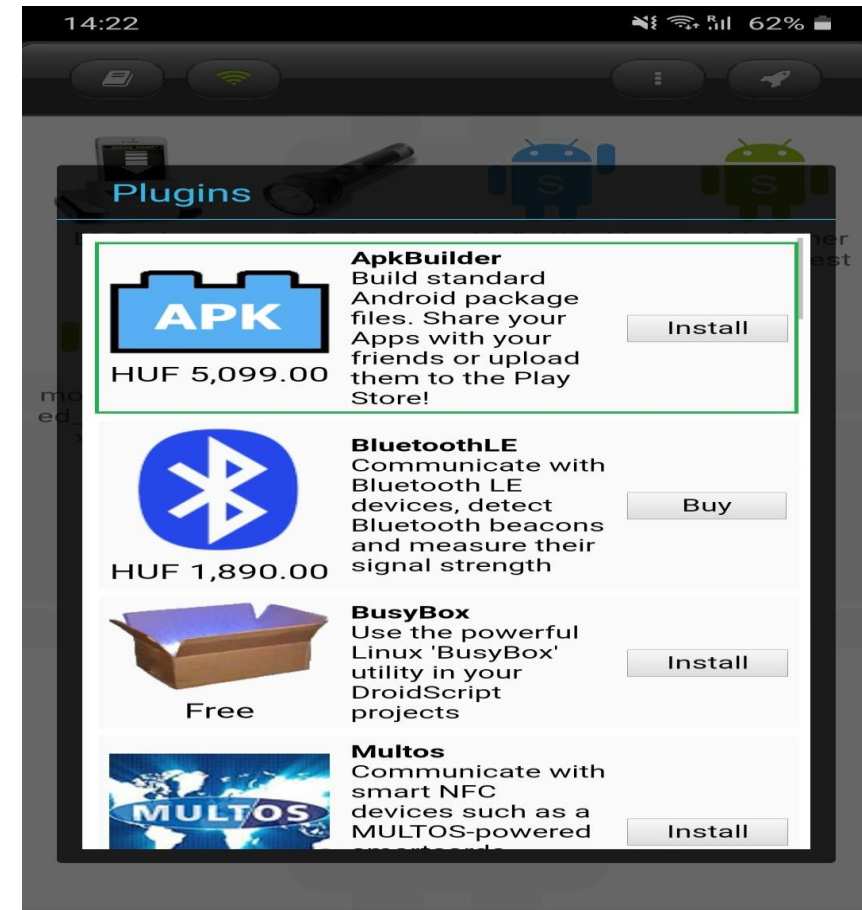


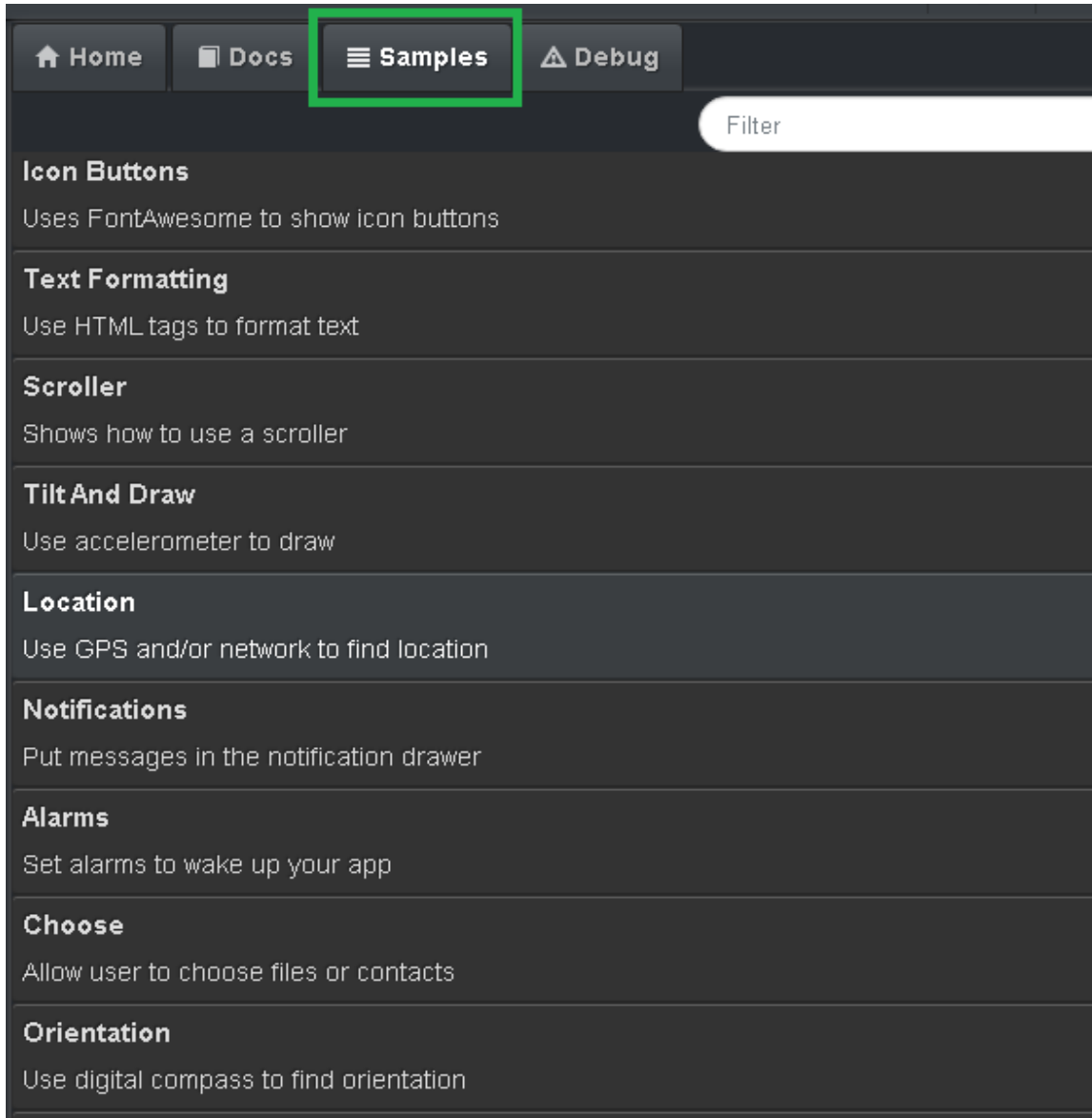
How to share your Mobile App with other?

Answer: APK (Android Package)



Install this plugin to build APK file for sharing with Others or to publish to Google Play Store





Where to go from here?

- Learn more code examples like, how to use Notification, Text editor, File-Handling, WIFI, Sound, Audio/Music, Voice-Commands, Video-Player, Calculator, WebView, Encryption, IOT device, SMS, Games, Database, Google-Analytics, E-mail, Web-Services, Charts, etc.
- Browse the documentation (Docs) for each control like Camera, Layout, GPS, etc.
- Read, code and test, practice and more practice.
- Publish your Mobile App to Google Play-Store.

Thank you...شكرا