



The enigma2-plugin Tutorial

by emanuel@i-have-a-dreambox.com

english rel. 1* date: 14.02.2010

(sorry for my bad english)

This plugin tutorial is for advanced users. If you've got problems with the following list below, take a look at our [>>> gp-wiki <<](#) .

If you haven't made a python tutorial yet, take a look at [>>>that<<](#).

What do you need:

1. Unix Editor (for exmpl. GNU notepad++.exe (Mswin))
2. Ftp connection to Dreambox
3. a Terminal (telnet/ssh) to Dreambox
4. Basic knowledge in Dreambox BusyBox
5. [Basic knowledge in Python \(use your Dreambox to test\)](#)
6. install lhad plugin Tutorial, Tutorial stored at:
/usr/lib/enigma2/python/Plugins/lhadTutorial/
7. **usr/lib/enigma2/python/Plugins/lhadTutorial/doc/doc_enigma2** (read it also!)
(original ones from enigma2 CVS)
8. have fun

Please no questions about basic python knowledge, this is an enigma2 plugin tutorial!

Table of contents:

- "Our Small Test" - pure print example without Gui (OSD) p. 3-5
- lesson "01 Hallo World" – simple window (Screen) p. 6-7
- lesson "02 Hallo World Message" – Screen with message p. 8-9
- lesson "03 Call My Msg" - Screen with message Yes/No p. 10-11
- lesson "04 My Menulist" – Screen with menulist p. 12-14
- lesson "05 My Shell Prombt" – Screen with menulist and shell commands, output on Screen Console p. 15-17
- lesson "06 Message Input" – Screens with character input, output p. 18-20
- lesson "07 View a picture" - Screen with picture p. 21-23
- lesson "08 Download a picture" - Download a picture and show on Screen p. 24-25
- lesson "09 dynamic Text" - Change a textlabel of the Screens p. 26-27
- lesson "10 Set Auto Sleep" – Screen to config of the dreambox's starting behavior p. 28-30
- lesson "11 Start other plugin" – Screen to Start picture player p. 31-32

If you've made your basic python tutorial well, let's start...

Our Small Test

Ok, if you have installed the Tutorial plugin, start a telnet session to your dreambox and type in:

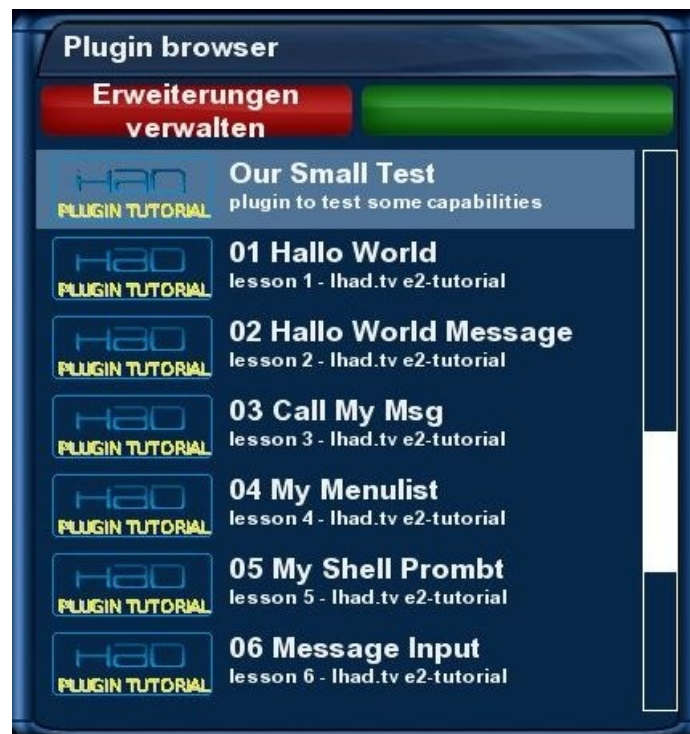
code:
root@dm8000:~# init 4; sleep 4; enigma2

Enigma2 restarts and you can see the enigma2 messages in telnet. All print commands or error messages of your plugin will be shown there. (also during enigma2 startup)

Telnet is developers best friend!

Start the mini plugin from enigma2 cvs doc, for testing.
(/usr/lib/enigma2/python/Plugins/lhadTutorial/OurSmallTest/..)

Open Plugin Browser and select:



Take a look at telnet:

```
code:
...
hdd IDLE!
[IDLE] 251.999181986 120 True
action -> WizardActions ok

<<<<<<<<<< Hello world! >>>>>>>>>

[EPGC] start cleanloop
[EPGC] stop cleanloop
...
```

Caution!

Press [Ctrl c] to stop enigma2.

```
code:
...
- (10) gRC
waiting for gRC thread shutdown
gRC thread has finished
- (9) gLCDDC
- (9) GFBDC
- (9) Font Render Class
- (8) graphics acceleration manager
- (1) Background File Eraser
reached rl -1
close frontend 0
root@dm8000:~#
```

Please wait till the empty input appear in telnet, because enigma need to save configs!!

To start up enigma2 again:

```
code:
root@dm8000:~# init 4; sleep 4; init 3
```

The src to "Our smal Test":

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # "Our Small Test" - taken from enigma2/doc
3
4 from Plugins.Plugin import PluginDescriptor
5
6 def main(session, **kwargs):
7     print "\n<<<<<<<<< Hello world! >>>>>>>>\n"
8
9 def Plugins(**kwargs):
10     return PluginDescriptor(
11         name="Our Small Test",
12         description="plugin to test some capabilities",
13         where = PluginDescriptor.WHERE_PLUGINMENU,
14         icon="../ihad_tut.png",
15         fnc=main)
16
```

Take a look at the module imports:

from Plugins.Plugin (file) /usr/lib/enigma2/python/Plugins/Plugin.py

the **PluginDescriptor** will be loaded. Take always a look at the module files you are importing, in order to know what you are loading, which parameters do you need. For beginner it is a little bit difficult, but that becomes better with time.

In our case it is the module, which prompt our plugin in the Plugin Browser.

If you take a look at the file, you can see there are a lot of PluginDescriptors - more in lesson 10.

Info:

write and edit your enigma2 plugins direct on dreambox. To get your changes working restart enigma2 (see above)

Store you plugins at:

/usr/lib/enigma2/python/Plugins/Extensions/<your Pluginfolder> for example "MyPlugin".

This Tutorial is only an Exception, for better overview, do not store plugins in other folders!

A plugin needs:

a file **__init__.py** (could be empty) and a file **plugin.py** (incl. src).

01 Hallo World

A simple window (screen) with textlabel and exit command.

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 1
3 # by emanuel
4 #####
5
6 from Screens.Screen import Screen
7 from Components.Label import Label
8 from Components.ActionMap import ActionMap
9 from Plugins.Plugin import PluginDescriptor
10
11 #####
12
13 class HalloWorldScreen(Screen):
14     skin = """
15         <screen position="130,150" size="460,150" title="Ihad.tv e2-tutorial
16         lesson 1" >
17             <widget name="myLabel" position="10,60" size="200,40"
18             font="Regular;20"/>
19         </screen>"""
20
21     def __init__(self, session, args = None):
22         self.session = session
23
24         Screen.__init__(self, session)
25         self["myLabel"] = Label("Hello World ;-)")
26         self["myActionMap"] = ActionMap(["SetupActions"],
27         {
28             "cancel": self.close # add the RC Command "cancel" to close
29         }, -1)
30
31     your Screen
32
33 #####
34
35 def main(session, **kwargs):
36     print "\n[Hallo World] start\n"
37
38     session.open(HalloWorldScreen)
39
40 #####
41
42 def Plugins(**kwargs):
43     return PluginDescriptor(
44         name="01 Hallo World",
45         description="lesson 1 - Ihad.tv e2-tutorial",
46         where = PluginDescriptor.WHERE_PLUGINMENU,
47         icon="../ihad_tut.png",
48         fnc=main)
```

In lesson 01 you can see how to build a Screen class. What do you need?

- 1) imports line 6-9 (take a look at the files behind!!)
(/usr/lib/enigma2/python/...)
- 2) Screen class **HalloWorldScreen** line 13-27
- 3) main function (starts the HalloWorldScreen) line 32-35
- 4) PluginDescriptor line 39-45

Explanation HalloWorldScreen class:

Line 14-17 the skin is defined in xml format. 1 screen, 1 widget (Textlabel)
Line19-27 initialisation of the class.

line 19:

__init__ will be called at plugin start (note parameters).
A Screen needs as his first parameter "self", as second a session.
In our example the session parameter comes from the main function.

line 20:

The parameter session (provided by **__init__**) will be saved as class intern global variable **self.session** for further use.

line 22:

Call initialisation function of the class including parameters **self**, **self.session**

line 23:

widget "myLabel" (attributs set in line 16) becomes a static text.

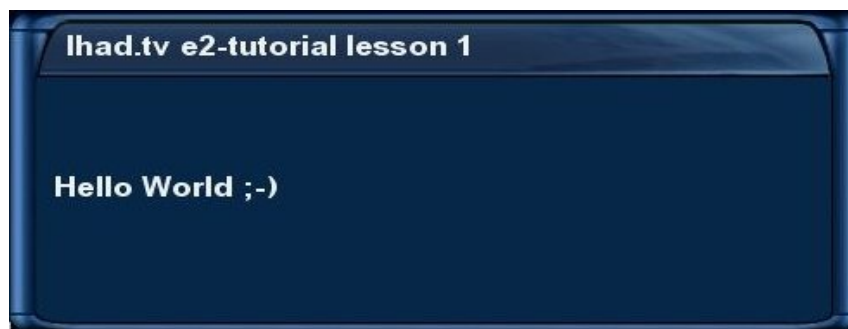
line 24-27:

The ActionMap will be define. In our case the minimum is exit the Screen.
See line 26 **self.close** becomes a RC command "**cancel**" from SetupActions.

note:

ActionMap "SetupActions" is stored in:
/usr/share/enigma2/keymap.xml

result:



02 Hallo World Message

a Screen with message

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 2
3 # by emanuel
4 from Screens.Screen import Screen
5 from Components.Label import Label
6 from Components.ActionMap import ActionMap
7 from Screens.MessageBox import MessageBox
8 from Plugins.Plugin import PluginDescriptor
9
10 #####
11
12 class HalloWorldMsg(Screen):
13     skin = """
14         <screen position="130,150" size="460,150" title="Ihad.tv e2-tutorial
15         lesson 2" >
16             <widget name="myLabel" position="10,60" size="200,40"
17             font="Regular;20"/>
18             </screen>"""
19
20     def __init__(self, session, args = 0):
21         self.session = session
22         Screen.__init__(self, session)
23
24         self["myLabel"] = Label(_("please press OK"))
25         self["myActionMap"] = ActionMap(["SetupActions"],
26         {
27             "ok": self.myMsg,
28             "cancel": self.cancel
29         }, -1)
30
31     def myMsg(self):
32         print "\n[HalloWorldMsg] OK pressed \n"
33         self.session.open(MessageBox, _("Hello World!"), MessageBox.TYPE_INFO)
34
35     def cancel(self):
36         print "\n[HalloWorldMsg] cancel\n"
37         self.close(False, self.session)
38
39 #####
40
41 def main(session, **kwargs):
42     print "\n[HalloWorldMsg] start\n"
43     session.open(HalloWorldMsg)
44
45 #####
46
47 def Plugins(**kwargs):
48     return PluginDescriptor(
49         name="02 Hallo World Message",
50         description="lesson 2 - Ihad.tv e2-tutorial",
51         where = PluginDescriptor.WHERE_PLUGINMENU,
52         icon="../ihad_tut.png",
53         fnc=main)
```


Explanation HalloWorldMsg Class:

As extension to lessen 01, take a look at line 7, 25-26 , 29-35

line 7:

import of Screen class **MessageBox**
(/usr/lib/enigma2/python/Screens/MessageBox.py)

line 25-26:

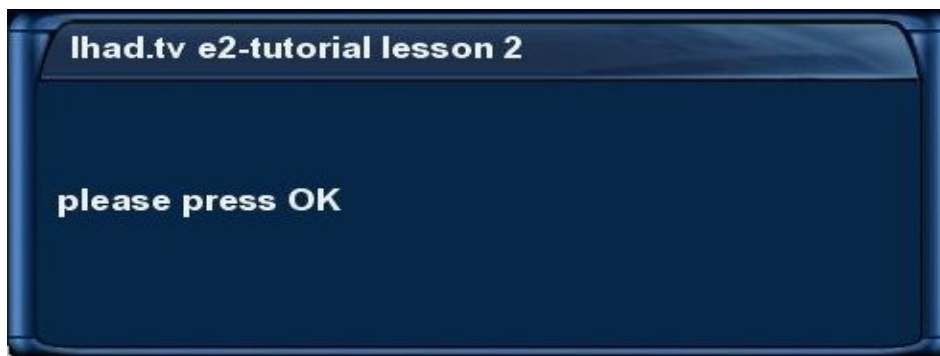
Set class function to ActionMap line 29-35.

line 29-35:

in “**def myMsg**” Screen MessageBox will be called.

in “**def cancel**” normal **self.close** will be called with parameters
(only for test example)

result:



03 Call My Msg

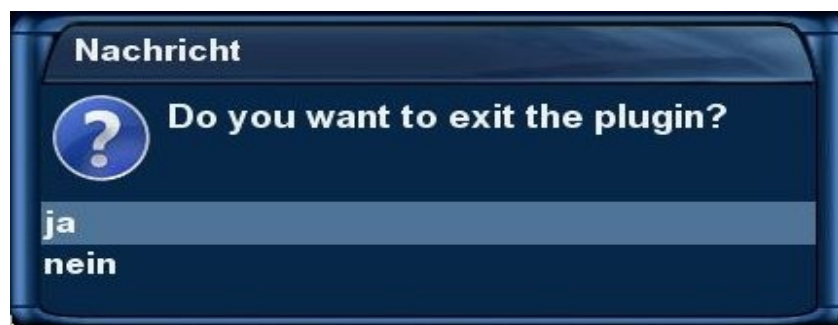
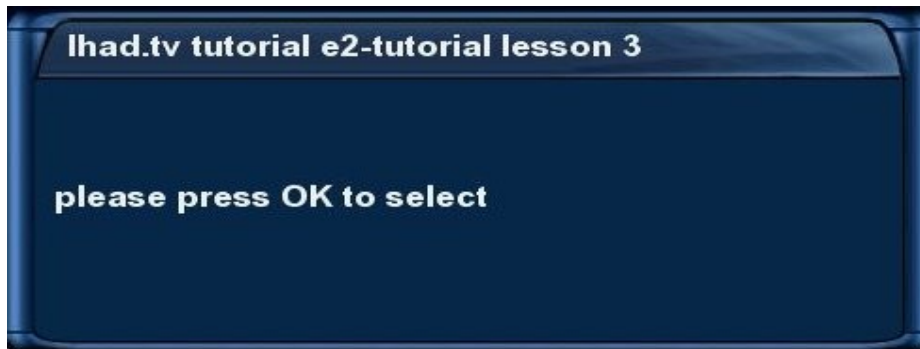
A Screen with message Yes/No

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 3
3 # by emanuel
4 from Screens.Screen import Screen
5 from Components.Label import Label
6 from Components.ActionMap import ActionMap
7 from Screens.MessageBox import MessageBox
8 from Plugins.Plugin import PluginDescriptor
9
10 #####
11
12 class CallMyMsg(Screen):
13     skin = """
14         <screen position="130,150" size="460,150" title="Ihad.tv tutorial e2-
15 tutorial lesson 3" >
16             <widget name="myLabel" position="10,60" size="400,120"
17 font="Regular;20"/>
18             </screen>"""
19
20     def __init__(self, session, args = 0):
21         self.session = session
22         Screen.__init__(self, session)
23
24         self["myLabel"] = Label(_("please press OK to select"))
25         self["myActionMap"] = ActionMap(["SetupActions"],
26 {
27     "ok": self.myMsg,
28     "cancel": self.cancel
29 }, -1)
30
31     def callMyMsg(self, result):
32         print "\n[CallMyMsg] checking result\n"
33         if result:
34             print "\n[CallMyMsg] cancel\n"
35             self.close(None)
36         else:
37             self.session.open(MessageBox, _("Ah, you like the Ihad
38 plugin!\n;-)"), MessageBox.TYPE_INFO)
39
40     def myMsg(self):
41         print "\n[CallMyMsg] OK pressed \n"
42         self.session.openWithCallback(self.callMyMsg, MessageBox, _("Do you
43 want to exit the plugin?"), MessageBox.TYPE_YESNO)
44
45     def cancel(self):
46         print "\n[CallMyMsg] cancel\n"
47         self.close(None)
```

continuation src: CallMyMsg

```
45 #####
46
47 def main(session, **kwargs):
48     print "\n[CallMyMsg] start\n"
49     session.open(CallMyMsg)
50
51 #####
52
53 def Plugins(**kwargs):
54     return PluginDescriptor(
55         name="03 Call My Msg",
56         description="lesson 3 - Ihad.tv e2-tutorial",
57         where = PluginDescriptor.WHERE_PLUGINMENU,
58         icon="../ihad_tut.png",
59         fnc=main)
60
```

result:



04 My Menulist

Screen with menulist

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 4
3 # by emanuel
4 from Screens.Screen import Screen
5 from Components.MenuList import MenuList
6 from Components.ActionMap import ActionMap
7 from Screens.MessageBox import MessageBox
8 from Plugins.Plugin import PluginDescriptor
9
10 #####
11
12 class MyMenu(Screen):
13     skin = """
14         <screen position="100,150" size="460,400" title="Ihad.tv tutorial e2-
15 tutorial lesson 4" >
16             <widget name="myMenu" position="10,10" size="420,380"
17 scrollbarMode="showOnDemand" />
18         </screen>"""
19
20     def __init__(self, session, args = 0):
21         self.session = session
22
23         list = []
24         list.append(("Entry 1"), "one")
25         list.append(("Entry 2"), "two")
26         list.append(("Entry 3"), "tree")
27         list.append(("Exit"), "exit")
28
29         Screen.__init__(self, session)
30         self["myMenu"] = MenuList(list)
31         self["myActionMap"] = ActionMap(["SetupActions"],
32 {
33     "ok": self.go,
34     "cancel": self.cancel
35 }, -1)
36
37     def go(self):
38         returnValue = self["myMenu"].l.getCurrentSelection()[1]
39         print "\n[MyMenu] returnValue: " + returnValue + "\n"
40
41         if returnValue is not None:
42             if returnValue is "one":
43                 self.myMsg("1")
44
45             elif returnValue is "two":
46                 self.myMsg("2")
47
48             elif returnValue is "tree":
49                 self.myMsg("3")
50
51             else:
52                 print "\n[MyMenu] cancel\n"
53                 self.close(None)
```

continuation src: MyMenu

```
53     def myMsg(self, entry):
54         self.session.open(MessageBox, _("You selected entry no. %s!")
55 % (entry), MessageBox.TYPE_INFO)
56
57     def cancel(self):
58         print "\n[MyMenu] cancel\n"
59         self.close(None)
60
61 #####
62
63 def main(session, **kwargs):
64     print "\n[MyMenu] start\n"
65     session.open(MyMenu)
66
67 #####
68
69 def Plugins(**kwargs):
70     return PluginDescriptor(
71         name="04 My Menulist",
72         description="lesson 4 - Ihad.tv e2-tutorial",
73         where = PluginDescriptor.WHERE_PLUGINMENU,
74         icon="../ihad_tut.png",
75         fnc=main)
76
```

Explanation MyMenu class:

line 5:

import of the menulist. Take a look at Components.MenuList file!
(/usr/lib/enigma2/python/Components/MenuList.py)

line 15:

in Screen Skin is only one widget for Menulist defined.

line 21-25:

the python list for Menulist will be build.

line 28:

widget "myMenu" is the **MenuList**, the in line 21-25 made list is used as parameter.

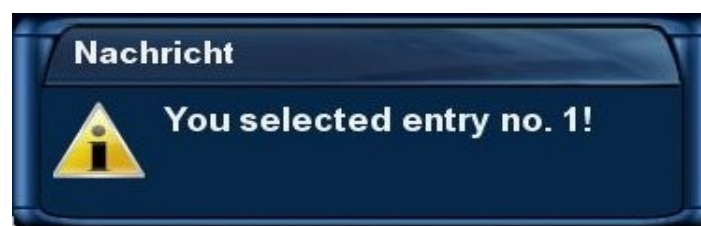
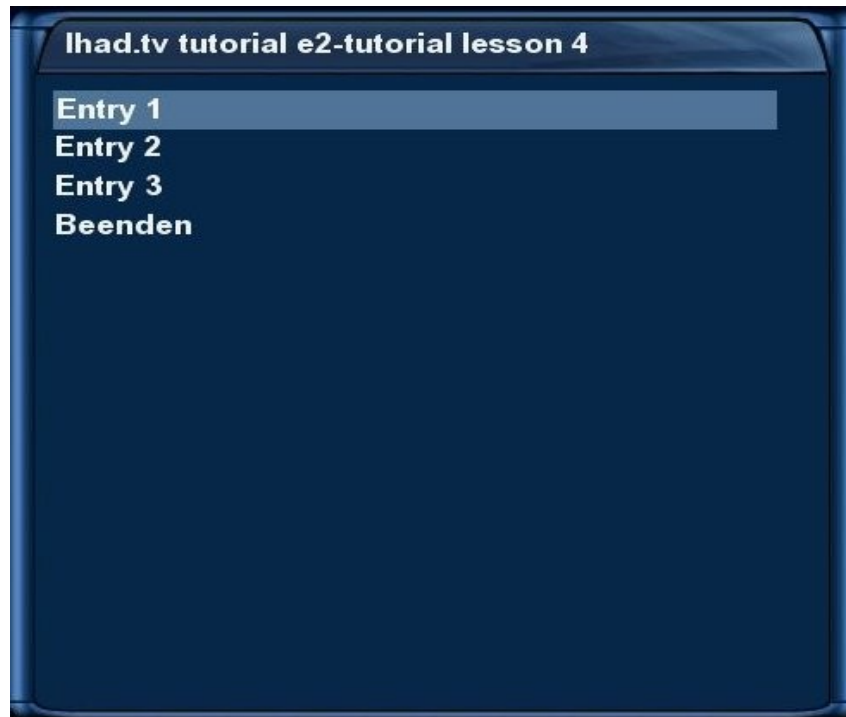
line 31:

RC Command "ok" is set to "self.go".

line 35-51:

`self["myMenu"].l.getCurrentSelection()[1]` provides the second listentry of the selected MenuList Entry, selected by RC Command "OK" list entry. example:
`(_("Entry 1"), "one") => "one"`

result:



05 My Shell Prombt

Screen with menulist and shell commands , outpote on Screen Console

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 5 - copyrights 2010 by emanueel@ihad.tv
3 # by emanuel
4 from Screens.Screen import Screen
5 from Screens.Console import Console
6 from Components.MenuList import MenuList
7 from Components.ActionMap import ActionMap
8 from Plugins.Plugin import PluginDescriptor
9
10 #####
11
12 class MyShPrombt(Screen):
13     skin = """
14         <screen position="100,150" size="460,400" title="Ihad.tv tutorial e2-
15 tutorial lesson 5" >
16             <widget name="myMenu" position="10,10" size="420,380"
17 scrollbarMode="showOnDemand" />
18         </screen>"""
19
20     def __init__(self, session, args = 0):
21         self.session = session
22
23         list = []
24         list.append(("netstat", "com_one"))
25         list.append(("ls -ls /", "com_two"))
26         list.append(("mount", "com_tree"))
27         list.append(("Exit", "exit"))
28
29         Screen.__init__(self, session)
30         self["myMenu"] = MenuList(list)
31         self["myActionMap"] = ActionMap(["SetupActions"],
32 {
33     "ok": self.go,
34     "cancel": self.cancel
35 }, -1)
36
37     def go(self):
38         returnValue = self["myMenu"].l.getCurrentSelection()[1]
39         print "\n[MyShPrombt] returnValue: " + returnValue + "\n"
40
41         if returnValue is not None:
42             if returnValue is "com_one":
43                 self.prombt("/bin/netstat")
44
45             elif returnValue is "com_two":
46                 self.prombt("/bin/ls -ls /")
47
48             elif returnValue is "com_tree":
49                 self.prombt("/bin/mount")
50
51             else:
52                 print "\n[MyShPrombt] cancel\n"
53                 self.close(None)
```

continuation src: MyShPrombt

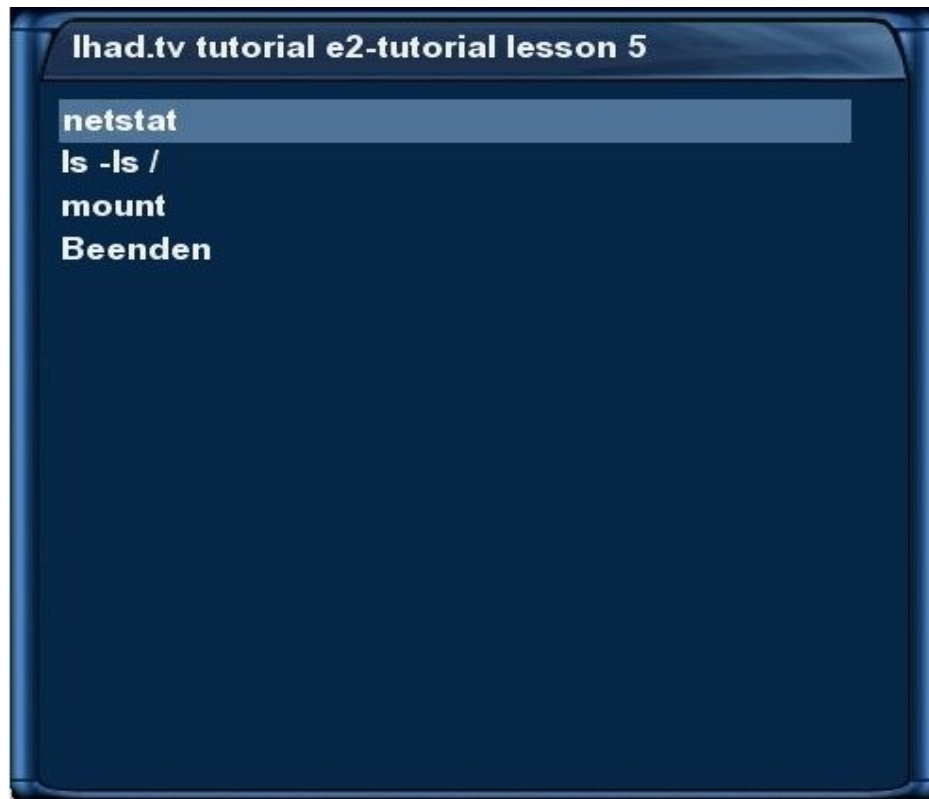
```
53     def go(self):
54         returnValue = self["myMenu"].l.getCurrentSelection()[1]
55         print "\n[MyShPrombt] returnValue: " + returnValue + "\n"
56
57         if returnValue is not None:
58             if returnValue is "com_one":
59                 self.prombt("/bin/netstat")
60
61             elif returnValue is "com_two":
62                 self.prombt("/bin/ls -ls /")
63
64             elif returnValue is "com_tree":
65                 self.prombt("/bin/mount")
66
67             else:
68                 print "\n[MyShPrombt] cancel\n"
69                 self.close(None)
70
71         def prombt(self, com):
72             self.session.open(Console, _("start shell com: %s") % (com), ["%s" %
73 com])
74
75         def cancel(self):
76             print "\n[MyShPrombt] cancel\n"
77             self.close(None)
78
79 #####
80
81 def main(session, **kwargs):
82     print "\n[MyShPrombt] start\n"
83     session.open(MyShPrombt)
84
85 #####
86
87 def Plugins(**kwargs):
88     return PluginDescriptor(
89         name="05 My Shell Prombt",
90         description="lesson 5 - Ihad.tv e2-tutorial",
91         where = PluginDescriptor.WHERE_PLUGINMENU,
92         icon="../ihad_tut.png",
93         fnc=main)
94
```

Explanation MyShPrombt class:

line 5:
import of Console

line 71-72:
“def prombt” starts a Screen Console. As parameter the shell commands are used.
line: 41, 44, 47

result:



06 Message Input

Screens with character input, output

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 6
3 # by emanuel
4 from Screens.Screen import Screen
5 from Components.Label import Label
6 from Components.ActionMap import ActionMap
7 from Components.Input import Input
8 from Screens.InputBox import InputBox
9 from Screens.MessageBox import MessageBox
10 from Plugins.Plugin import PluginDescriptor
11
12 #####
13
14 class MsgInput(Screen):
15     skin = """
16     <screen position="130,150" size="460,150" title="Ihad.tv e2-tutorial
17     lesson 6" >
18         <widget name="myLabel" position="10,60" size="200,40"
19         font="Regular;20"/>
20     </screen>"""
21
22     def __init__(self, session, args = 0):
23         self.session = session
24         Screen.__init__(self, session)
25
26         self["myLabel"] = Label(_("please press OK"))
27         self["myActionMap"] = ActionMap(["SetupActions"],
28         {
29             "ok": self.myInput,
30             "cancel": self.cancel
31         }, -1)
32
33     def myInput(self):
34         self.session.openWithCallback(self.askForWord, InputBox,
35         title=_("Please enter a name for prombt!"), text=" " * 55, maxSize=55,
36         type=Input.TEXT)
37
38     def askForWord(self, word):
39         if word is None:
40             pass
41         else:
42             self.session.open(MessageBox,_(word), MessageBox.TYPE_INFO)
43
44     def cancel(self):
45         print "\n[MsgInput] cancel\n"
46         self.close(None)
```

continuation src: MsgInput

```
44 #####
45
46 def main(session, **kwargs):
47     print "\n[MsgInput] start\n"
48     session.open(MsgInput)
49
50 #####
51
52 def Plugins(**kwargs):
53     return PluginDescriptor(
54         name="06 Message Input",
55         description="lesson 6 - Ihad.tv e2-tutorial",
56         where = PluginDescriptor.WHERE_PLUGINMENU,
57         icon="../ihad_tut.png",
58         fnc=main)
59
```

Explanation **MsgInput** class:

The class `MsgInput` is a little bit like the lesson 03. Only the input does not provide 0 or 1 as result, in this case it is the input of the RC/Keyboard.

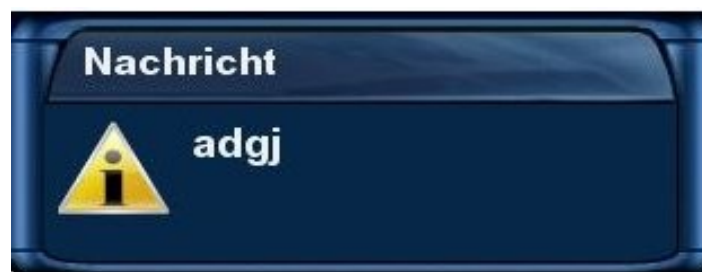
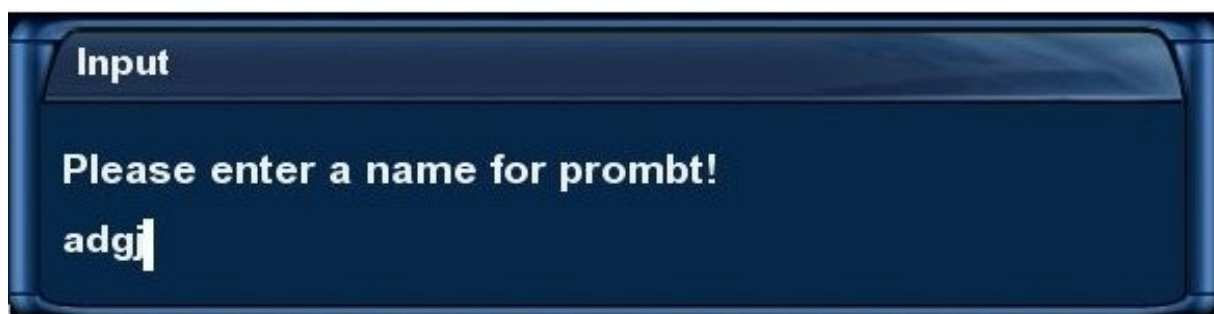
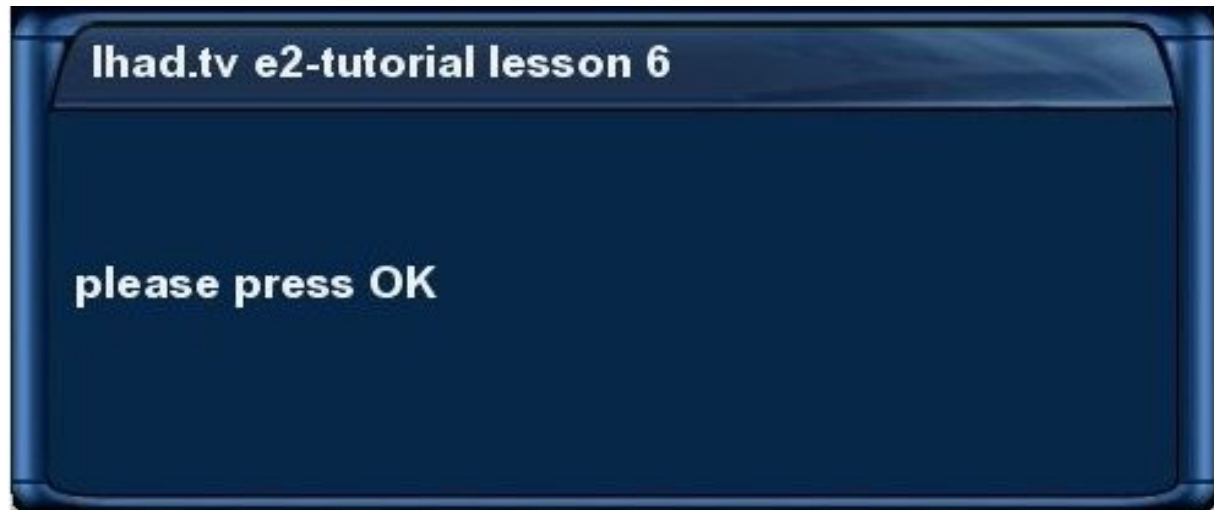
line 7,8:

import from `Input`, `InputBox`

line 31-38:

“`def myInput`” starts `InputBox` Screen. `openWithCallback` provides the result for “`def askForWord`” from `InputBox`.

result:



07 View a picture

Screen with picture

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 7
3 # by emanuel
4 from Screens.Screen import Screen
5 from Components.Label import Label
6 from Components.Pixmap import Pixmap
7 from Components.AVSwitch import AVSwitch
8 from Components.ActionMap import ActionMap
9 from Plugins.Plugin import PluginDescriptor
10 from enigma import ePicLoad
11
12 #####
13
14 class PictureScreen(Screen):
15
16     skin="""
17         <screen name="PictureScreen" position="0,0" size="720,576"
18         title="Picture Screen" backgroundColor="#002C2C39">
19             <widget name="myPic" position="0,0" size="720,576"
20             zPosition="1" alphatest="on" />
21         </screen>"""
22
23     def __init__(self, session, picPath = None):
24         Screen.__init__(self, session)
25         print "[PictureScreen] __init__\n"
26         self.picPath = picPath
27         self.Scale = AVSwitch().getFramebufferScale()
28         self.PicLoad = ePicLoad()
29         self["myPic"] = Pixmap()
30         self["myActionMap"] = ActionMap(["SetupActions"],
31         {
32             "ok": self.cancel,
33             "cancel": self.cancel
34         }, -1)
35
36         self.PicLoad.PictureData.get().append(self.DecodePicture)
37         self.onLayoutFinish.append(self.ShowPicture)
38
39     def ShowPicture(self):
40         if self.picPath is not None:
41             self.PicLoad.setPara([
42                 self["myPic"].instance.size().width(),
43                 self["myPic"].instance.size().height(),
44                 self.Scale[0],
45                 self.Scale[1],
46                 0,
47                 1,
48                 "#002C2C39"])
49             self.PicLoad.startDecode(self.picPath)
```

continuation src: PictureScreen

```
51     def DecodePicture(self, PicInfo = ""):
52         if self.picPath is not None:
53             ptr = self.PicLoad.getData()
54             self["myPic"].instance.setPixmap(ptr)
55
56
57     def cancel(self):
58         print "[PictureScreen] - cancel\n"
59         self.close(None)
60
61     #####
62
63     def main(session, **kwargs):
64         session.open(PictureScreen, picPath =
65         "/usr/share/enigma2/skin_default/icons/dish.png")
66
67     #####
68     def Plugins(**kwargs):
69         return PluginDescriptor(
70             name="07 View a picture",
71             description="lesson 7 - Ihad.tv e2-tutorial",
72             where = PluginDescriptor.WHERE_PLUGINMENU,
73             icon="../ihad_tut.png",
74             fnc=main)
75
```

Explanation PictureScreen class:

The class PictureScreen is an example of how to show a picture in a Screen.

line 6,7,9:

import of QPixmap, AVSwitch, ePicLoad (look at it!)

line 18:

in Screen Skin widget **"myPic"** for the picture will be defined.

line 21:

the Screen class PictureScreen takes from **__init__** a extra Parameter **"picPath"** called from function **"main"** in line 64.

line 26:

Memory for the picture

line 27:

widget **self["myPic"]** becomes a QPixmap() **without parameter!**

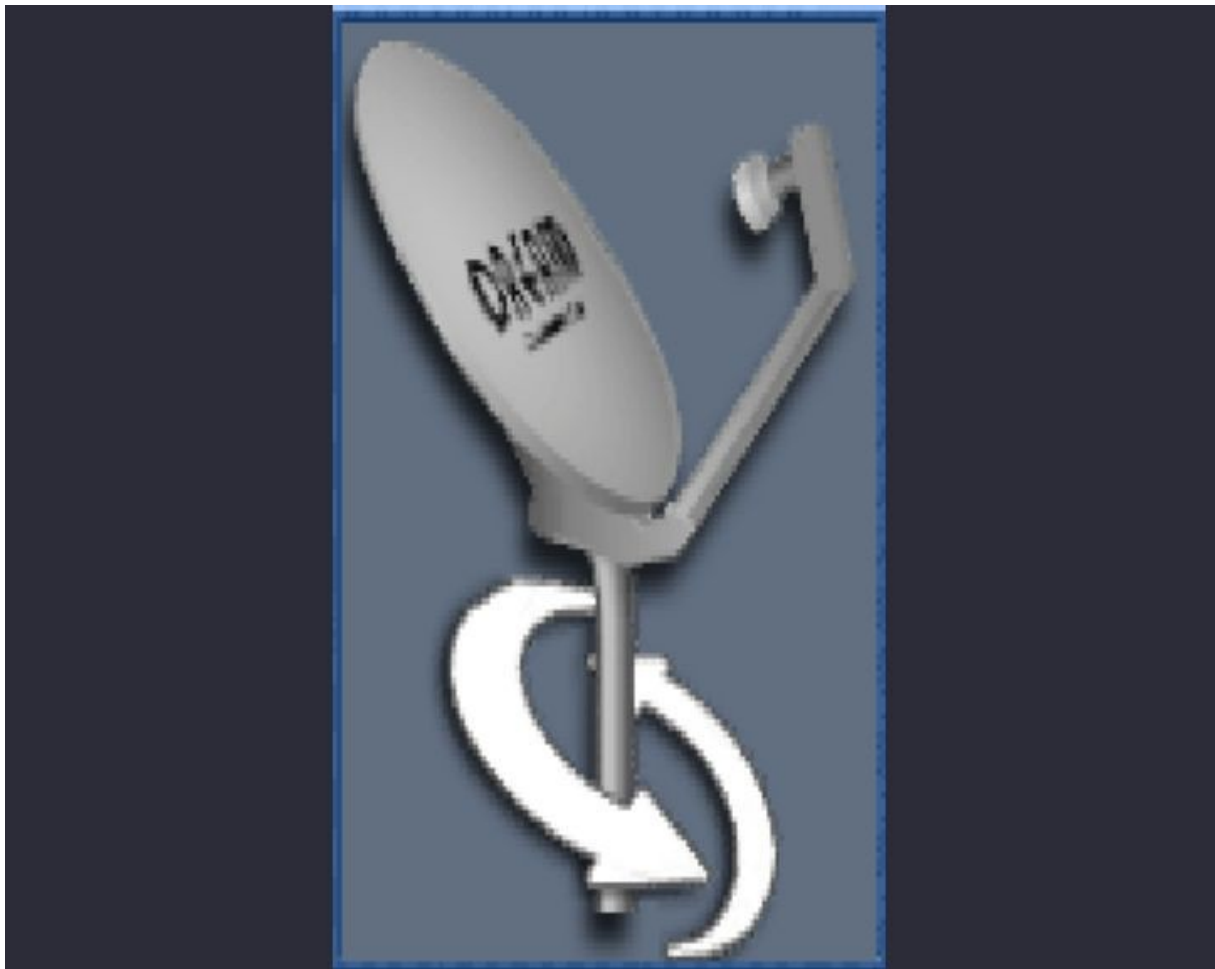
line 34:

loading und decoding of the picture, see line 51-54

line 35:

the Screen picture “onLayoutFinish.append(**self.ShowPicture**)” loads the picture see line 37-48, parameter for the picture will be set.

result:



08 Download a picture

Download a picture and show on Screen from lesson 07

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 8
3 # by emanuel
4 #####
5
6 from twisted.web.client import downloadPage
7 from Screens.MessageBox import MessageBox
8 from Plugins.IhadTutorial.lesson_07.plugin import PictureScreen
9 from Plugins.Plugin import PluginDescriptor
10
11 #####
12
13 class getPicfromUrl(object):
14     def __init__(self, session, url=None, path=None):
15         self.path = path
16         self.session = session
17         self.download(url, path)
18
19     def download(self, url, path):
20         downloadPage(url,
21 path).addCallback(self.downloadDone).addErrback(self.downloadError)
22
23     def downloadError(self, raw):
24         print "[e2Fetcher.fetchPage]: download Error", raw
25         self.session.open(MessageBox, text = _("Error downloading: ") +
26 self.path, type = MessageBox.TYPE_ERROR)
27
28     def downloadDone(self, raw):
29         print "[e2Fetcher.fetchPage]: download done", raw
30         self.session.open(PictureScreen, picPath = self.path)
31
32 #####
33 def main(session, **kwargs):
34     getPicfromUrl(session, "http://www.i-have-a-dreambox.com/images/ihad.jpg",
35 "/tmp/myPic.tmp")
36
37 #####
38 def Plugins(**kwargs):
39     return PluginDescriptor(
40         name="08 Download a picture",
41         description="lesson 8 - Ihad.tv e2-tutorial",
42         where = PluginDescriptor.WHERE_PLUGINMENU,
43         icon="../ihad_tut.png",
44         fnc=main)
```


Explanation getPicfromUrl class:

The class getPicfromUrl imports the Screen from lesson 07 to show the dowloaded picture.

line 6,8:

import of downloadPage,
PictureScreen from lesson 07 (self made Screen)

line 13-28:

class **getPicfromUrl(object)**: is no Screen; but it uses a session to start **PictureScreen** or some **MessageBox** tht modules do need “**session**” as parameter!
see line 14 **__init__**

result:



09 dynamic Text

Change a textlabel of Screen

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 9
3 # by emanuel
4 #####
5
6 from Screens.Screen import Screen
7 from Components.Label import Label
8 from Components.ActionMap import ActionMap
9 from Plugins.Plugin import PluginDescriptor
10
11 #####
12
13 class MyDynaTextScreen(Screen):
14     skin = """
15         <screen position="130,150" size="460,150" title="Ihad.tv e2-tutorial
16         lesson 9" >
17             <widget name="myText" position="10,50" size="400,40"
18             valign="center" halign="center" zPosition="2" foregroundColor="white"
19             font="Regular;22"/>
20             <widget name="myRedBtn" position="10,110" size="100,40"
21             backgroundColor="red" valign="center" halign="center" zPosition="2"
22             foregroundColor="white" font="Regular;20"/>
23             <widget name="myGreenBtn" position="120,110" size="100,40"
24             backgroundColor="green" valign="center" halign="center" zPosition="2"
25             foregroundColor="white" font="Regular;20"/>
26         </screen>"""
27
28     def __init__(self, session, args = 0):
29         self.session = session
30         Screen.__init__(self, session)
31
32         self.text="Press green or ok button to edit text!"
33         self["myText"] = Label()
34         self["myRedBtn"] = Label(_("Cancel"))
35         self["myGreenBtn"] = Label(_("OK"))
36         self["myActionsMap"] = ActionMap(["SetupActions", "ColorActions"],
37         {
38             "ok": self.editMyText,
39             "green": self.editMyText,
40             "red": self.close,
41             "cancel": self.close,
42         }, -1)
43         self.onShown.append(self.setMyText)
44
45     def setMyText(self):
46         self["myText"].setText(self.text)
47
48     def editMyText(self):
49         self.text="I love Ihad.tv!\n:-)"
50         self.setMyText()
```

continuation src: MyDynaTextScreen

```
45 #####
46
47 def main(session, **kwargs):
48     session.open(MyDynaTextScreen)
49
50 #####
51
52 def Plugins(**kwargs):
53     return PluginDescriptor(
54         name="09 dynamic Text",
55         description="lesson 9 - Ihad.tv e2-tutorial",
56         where = PluginDescriptor.WHERE_PLUGINMENU,
57         icon="../ihad_tut.png",
58         fnc=main)
59
```

Explanation MyDynaTextScreen class:

Now this should be no problem for you.

line 26:

`self["myText"] = Label()` with **no parameter** init! Note this is important for `self["myText"].setText(self.text)` in line 39!

line 36:

`self.onShown.append(self.setMyText)` to get text at startup!

result:



continuation src: AutoSleepScreen

```
51     def setSleep(self):
52         self.changed = True
53         if config.plugins.AutoSleep.enable.value:
54             config.plugins.AutoSleep.enable.setValue(False)
55         else:
56             config.plugins.AutoSleep.enable.setValue(True)
57         self.updateSettings()
58
59     def exit(self):
60         if self.changed:
61             config.plugins.AutoSleep.enable.save()
62         self.close(None)
63
64 #####
65
66 def main(session, **kwargs):
67     session.open(AutoSleepScreen)
68
69 #####
70 # start and stop enigma2 & and watch output in telnet
71
72 def autostart(reason, **kwargs):
73     print blank, line
74     if reason == 0:
75         print "[AutoSleep] - autostart sleep enabled: ",
76         config.plugins.AutoSleep.enable.getValue()
77     else:
78         print "[AutoSleep] - autostop sleep enabled: ",
79         config.plugins.AutoSleep.enable.getValue()
80     print tut_vers
81     print line, blank
82
83     if config.plugins.AutoSleep.enable.value:
84         time.sleep(10)
85 #####
86
87 def Plugins(**kwargs):
88     return [
89         PluginDescriptor(
90             where = PluginDescriptor.WHERE_AUTOSTART,
91             fnc = autostart),
92         PluginDescriptor(
93             name = "10 Set Auto Sleep",
94             description = "lesson 10 - Ihad.tv e2-tutorial",
95             where = PluginDescriptor.WHERE_PLUGINMENU,
96             icon = "../ihad_tut.png",
97             fnc = main)]
98
```

Explanation AutoSleepScreen class:

Now this should be no problem for you. New: the Configs and the Autostart PluginDescriptor.

line 10:

import of config, ConfigSubsection, ConfigYesNo

line 14:

config.plugins.AutoSleep = ConfigSubsection() **creating of Enigma sub configuratinos**

line 15:

config.plugins.AutoSleep.enable = ConfigYesNo(default = False)
a variable that 0/1 stores.

line 73-83:

The picture autostart is called by PluginDescriptor in line 91 and checks if **config.plugins.AutoSleep.enable** is set. If it is set, the enigma2 bootup will be stop for 10 sec. Watch this at startup/stop of your dreambox in telnet.

result:



"11 Start other plugin" Screen to start Pictureplayer

```
1 # Ihad.tv enigma2-plugin tutorial 2010
2 # lesson 11
3 # by emanuel
4 #####
5
6 from Screens.Screen import Screen
7 from Screens.MessageBox import MessageBox
8 from Components.Label import Label
9 from Components.ActionMap import ActionMap
10 from Plugins.Plugin import PluginDescriptor
11 from Tools.Directories import fileExists
12
13 #####
14
15 class MyPluginStartScreen(Screen):
16     skin = """
17         <screen position="130,150" size="460,150" title="Ihad.tv e2-tutorial
18 lesson 11" >
19             <widget name="myText" position="10,20" size="400,50"
20 valign="center" halign="center" zPosition="2" foregroundColor="white"
21 font="Regular;22"/>
22             <widget name="myRedBtn" position="10,110" size="100,40"
23 backgroundColor="red" valign="center" halign="center" zPosition="2"
24 foregroundColor="white" font="Regular;20"/>
25             <widget name="myGreenBtn" position="120,110" size="100,40"
26 backgroundColor="green" valign="center" halign="center" zPosition="2"
27 foregroundColor="white" font="Regular;20"/>
28         </screen>"""
29
30     def __init__(self, session, args = 0):
31         self.session = session
32         Screen.__init__(self, session)
33
34         self["myText"] = Label("Press green or ok button to start\nPicture
35 Player plugin!")
36         self["myRedBtn"] = Label(_("Cancel"))
37         self["myGreenBtn"] = Label(_("OK"))
38         self["myActionsMap"] = ActionMap(["SetupActions", "ColorActions"],
39 {
40     "ok": self.startPicplayer,
41     "green": self.startPicplayer,
42     "red": self.close,
43     "cancel": self.close,
44 }, -1)
45
46     def startPicplayer(self):
47         if
48 fileExists("/usr/lib/enigma2/python/Plugins/Extensions/PicturePlayer/plugin.py"):
49             from Plugins.Extensions.PicturePlayer.plugin import *
50             self.session.open(picshow)
51         else:
52             self.session.open(MessageBox, "No Picture Player found!",
53 MessageBox.TYPE_ERROR)
54
55 #####
56
57
```

continuation src: MyPluginStartScreen

```
50 def main(session, **kwargs):
51     session.open(MyPluginStartScreen)
52
53 #####
54
55 def Plugins(**kwargs):
56     return PluginDescriptor(
57         name="11 Start other plugin",
58         description="lesson 11 - Ihad.tv e2-tutorial",
59         where = PluginDescriptor.WHERE_PLUGINMENU,
60         icon="../ihad_tut.png",
61         fnc=main)
62
```

Explanation MyPluginStartScreen class:

Now this should be no problem for you. We imported Screens often in this tutorial.

line 10/42:

import of fileExists, and everything from PicturePlayerplug

result:

