

## Usando a facade UI



### Progress Dialog Boxes

O desafio aqui é determinar como calcular o progresso. Quando feito um download de um arquivo, por exemplo, o tamanho do arquivo fornece um parâmetro de medida do progresso.



### Outros Diálogos



### Mais diálogos



### Make a Toast

```
import android.  
droid = android.Android()  
nome = droid.getString("hello", "Qual seu nome?")  
print nome # nome é uma tupla  
droid.makeToast("hello, %s" % nome.result)
```

# Interface Gráfica

## *Programando Android*

## Usando a facade UI



### Progress Dialog Boxes

O desafio aqui é determinar como calcular o progresso. Quando feito um download de um arquivo, por exemplo, o tamanho do arquivo fornece um parâmetro de medida do progresso.



### Outros Diálogos



### Mais diálogos



### Make a Toast

```
import android
droid = android.Android()
nome = droid.getInput("Hello", "Qual seu nome?")
print nome # nome eh uma tupla
droid.makeToast("Hello, %s" % nome.result)
```

# Interface Gráfica

*Programando Android*

## Usando a facade UI



# Make a Toast

```
import android
```

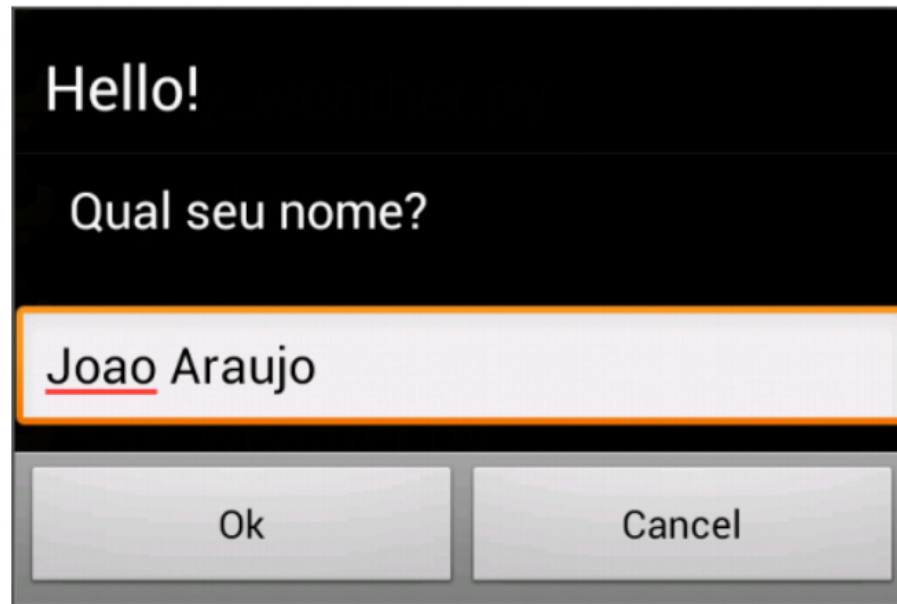
```
droid = android.Android()
```

```
nome = droid.getInput("Hello!", "Qual seu nome?")
```

```
print nome # nome eh uma tupla
```

```
droid.makeToast("Hello, %s" % nome.result)
```





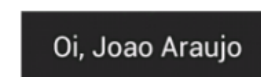
result é o valor da string lida

E no terminal, você vê:  
`Result(id=0, result=u'Joao Araujo', error=None)`

Cada resultado possui um único ID, para rastreamento

Cada resultado retorna também um código de erro.

Finalmente, você vê, por um breve período:



result é o valor da string lida

E no terminal, você vê:

```
Result(id=0, result=u'Joao Araujo', error=None)
```

Cada resultado retorna também um código de erro.

Cada resultado possui um único ID, para rastreamento

Finalmente, você vê, por um breve período:

Oi, Joao Araujo

```
# coding: UTF-8
import android
droid = android.Android()

droid.dialogCreateAlert('Configuração do
Diálogo', 'Escolha qualquer número de itens e pressione
OK')
droid.dialogShow()
```

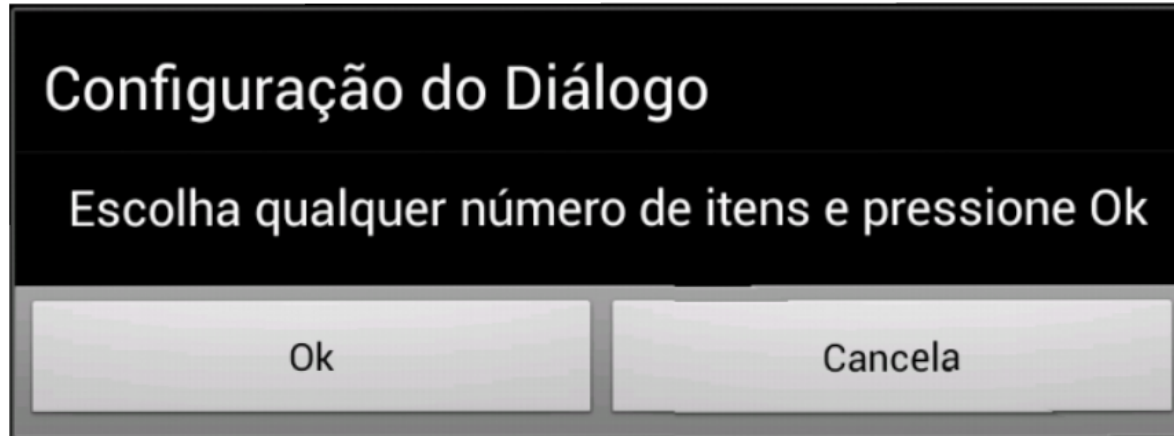
## Configuração do Diálogo

Escolha qualquer número de  
itens e pressione Ok



```
# coding: UTF-8
import android
droid = android.Android()

droid.dialogCreateAlert('Configuração do
Diálogo', 'Escolha qualquer número de itens e pressione
OK')
droid.dialogSetPositiveButtonText('Ok')
droid.dialogSetNegativeButtonText('Cancela')
droid.dialogShow()
```



Para saber o que o usuário teclou, você deve usar `droid.dialogGetResponse()`

```
>>> response = droid.dialogGetResponse()
```

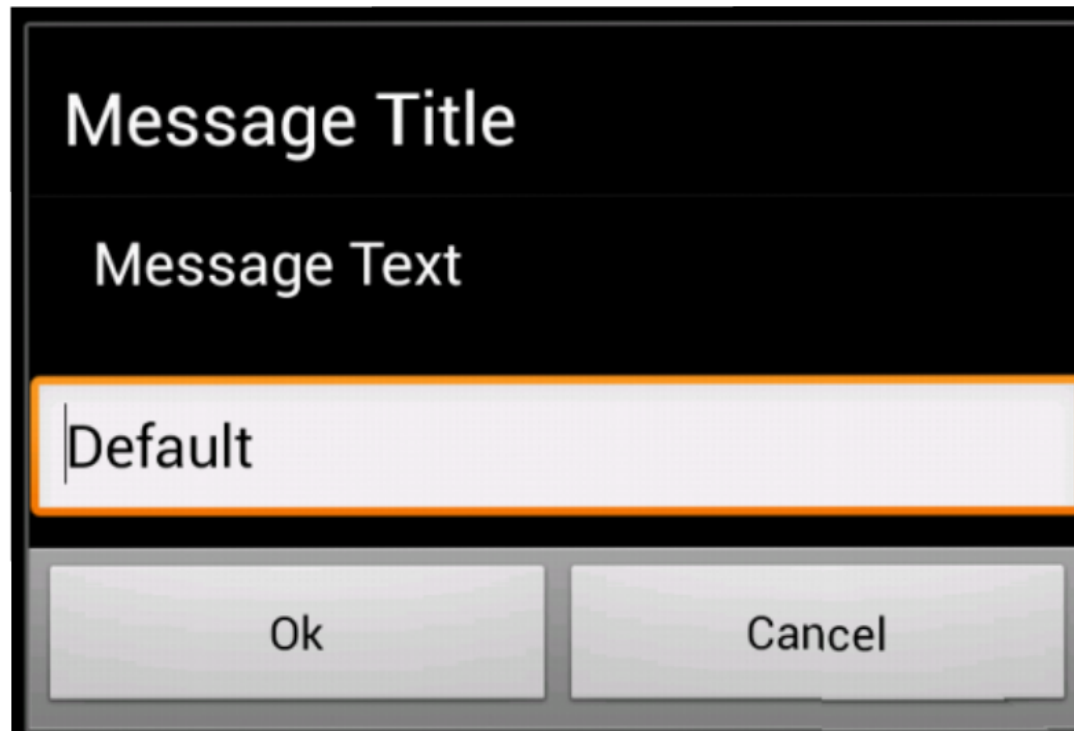
```
>>> response
```

```
Result(id=10, result={u'canceled': True}, error=None)
```

O comando:

```
ans = droid.dialogGetInput("Message Title","Message  
Text","Default").result
```

Guarda o valor na variável ans.



Se o usuário pressiona Cancel, você obtém:

`Result(id=0, result=None, error=None)`

`dialogGetResponse` retorna a última ação do usuário

```
>>> droid.dialogSetItems(['Um', 'Dois', 'Tres', 'Quatro', 'Cinco'])
Result(id=16, result=None, error=None)
>>> droid.dialogShow()
Result(id=17, result=None, error=None)
>>> droid.dialogGetResponse()
Result(id=18, result={u'canceled': True, u'which': u'positive'},
error=None)
>>> droid.dialogShow()
Result(id=19, result=None, error=None)
>>> droid.dialogGetResponse()
Result(id=20, result={u'item': 2}, error=None)
```

## Configuração do Diálogo

Um

Dois

Tres

Quatro

Cinco

Ok

Cancela

## Analizando a saída

```
>>> droid.dialogGetResponse()  
Result(id=18, result={u'canceled': True, u'which': u'positive'},  
error=None)
```

Diz que o usuário não escolheu nenhum item, pressionando o botão positivo, no caso 'Ok'.

## Analizando a saída

```
Result(id=20, result={u'item': 2}, error=None)
```

Este resultado diz que o usuário escolheu uma das opções, no caso '2' que se traduz por 'Tres' nas opções.

Não temos nenhum valor para o botão, pois o diálogo fecha tão logo algum dos itens seja escolhido.  
Precisamos apenas de um botão de cancela, neste caso.

Se o usuário escolhe Cancela, temos

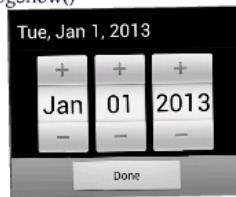
```
>>> droid.dialogGetResponse()  
Result(id=22, result={u'canceled': True, u'which':  
u'neutral'}, error=None)
```



# Outros Diálogos

## Data

```
droid.dialogCreateDatePicker(2011)
droid.dialogShow()
```



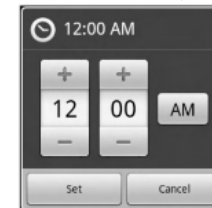
Podemos escolher um dia inicial, colocando ano, mês e dia, nessa ordem

```
>>> droid.dialogGetResponse()
Result(id=27, result={u'year': 2011, u'day': 6, u'which': u'positive',
u'month': 3},
error=None)
```

```
>>> date = droid.dialogGetResponse().result
>>> date
{u'year': 2011, u'day': 7, u'which': u'positive', u'month': 3}
>>> date["year"]
2011
>>> date["month"]
3
>>> date["day"]
7
```

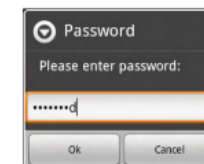
## createTimePicker

```
>>> droid.dialogCreateTimePicker()
Result(id=9, result=None, error=None)
>>> droid.dialogShow()
Result(id=10, result=None, error=None)
```



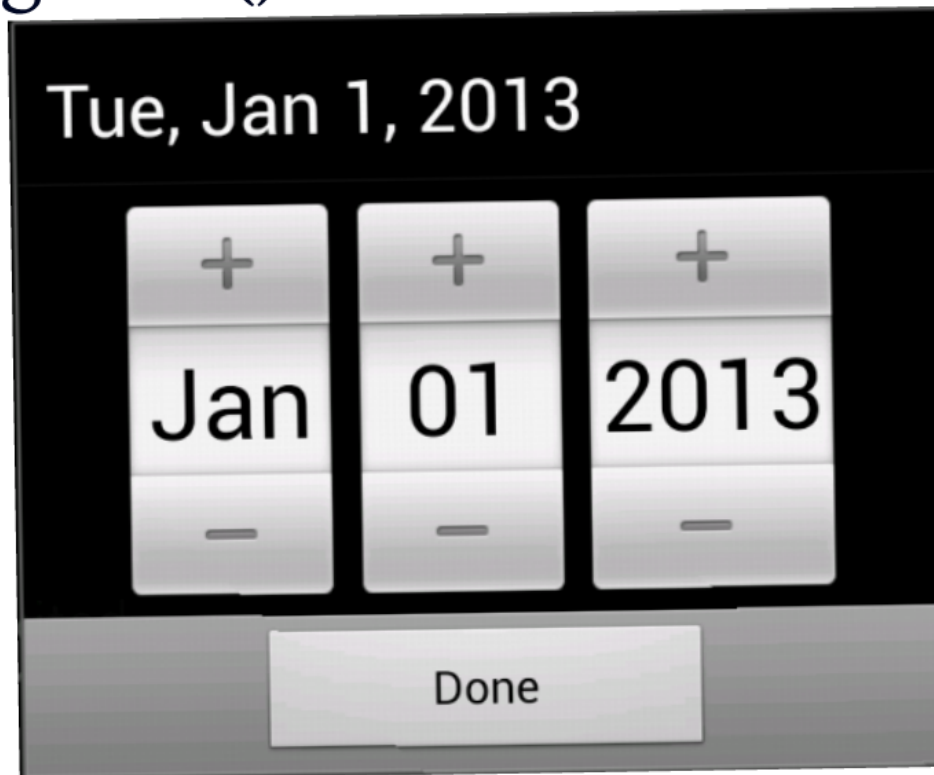
## dialogGetPassword()

```
Não precisa do dialogShow()
>>> droid.dialogGetPassword()
Result(id=5, result=u'Password', error=None)
>>> droid.dialogGetResponse()
Result(id=6, result={u'which': u'positive', u'value': u'Password'}, error=None)
>>> droid.dialogGetPassword()
Result(id=7, result=None, error=None)
>>> droid.dialogGetResponse()
Result(id=8, result={u'which': u'negative', u'value': u''}, error=None)
```



## Data

```
droid.dialogCreateDatePicker(2011)  
droid.dialogShow()
```



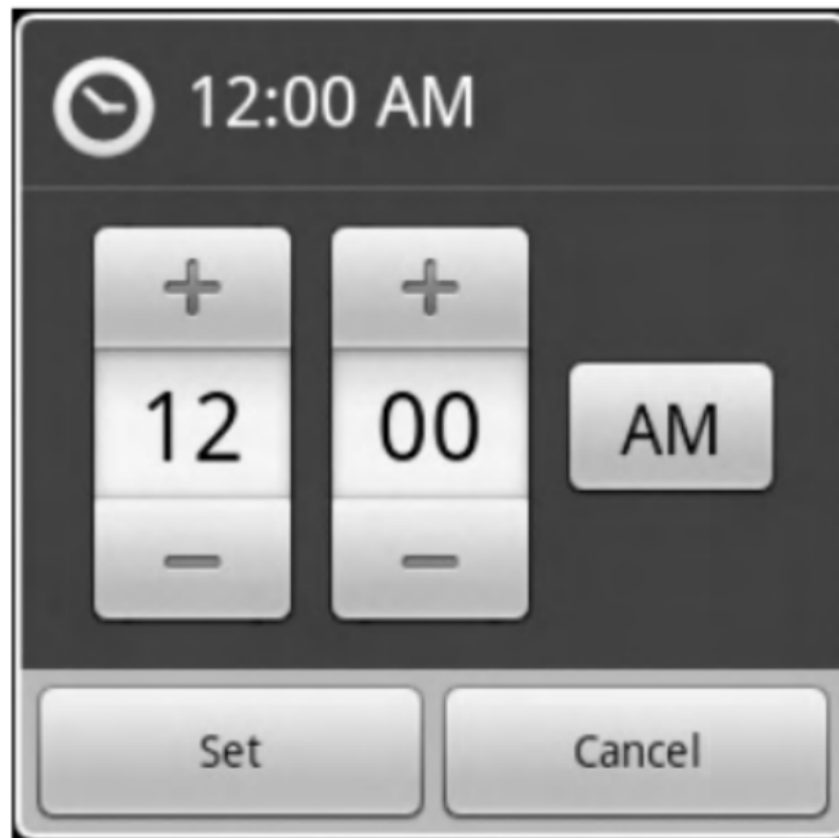
Podemos escolher um dia inicial, colocando ano, mês e dia, nessa ordem

```
>>> droid.dialogGetResponse()
Result(id=27, result={u'year': 2011, u'day': 6, u'which': u'positive',
u'month': 3},
error=None)
```

```
>>> date = droid.dialogGetResponse().result
>>> date
{u'year': 2011, u'day': 7, u'which': u'positive', u'month': 3}
>>> date["year"]
2011
>>> date["month"]
3
>>> date["day"]
7
```

## createTimePicker

```
>>> droid.dialogCreateTimePicker()  
Result(id=9, result=None, error=None)  
>>> droid.dialogShow()  
Result(id=10, result=None, error=None)
```



# dialogGetPassword()

Não precisa do dialogShow()

```
>>> droid.dialogGetPassword()
```

```
Result(id=5, result=u'Password', error=None)
```

```
>>> droid.dialogGetResponse()
```

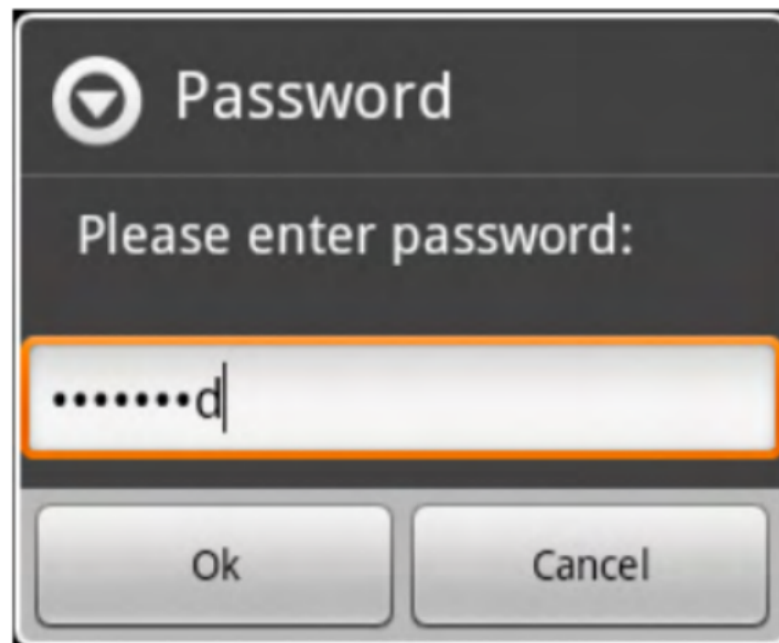
```
Result(id=6, result={u'which': u'positive', u'value': u'Password'}, error=None)
```

```
>>> droid.dialogGetPassword()
```

```
Result(id=7, result=None, error=None)
```

```
>>> droid.dialogGetResponse()
```

```
Result(id=8, result={u'which': u'negative', u'value': u''}, error=None)
```



# Progress Dialog Boxes

O desafio aqui é determinar como calcular o progresso. Quando feito um download de um arquivo, por exemplo, o tamanho do arquivo fornece um parâmetro de medida do progresso.

## dialogCreateHorizontalProgress

Não precisa de nenhum parâmetro

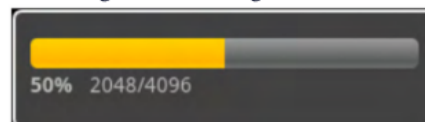
```
droid.dialogCreateHorizontalProgress()  
droid.dialogShow()
```



## Atualizando o progresso

dialogSetMaxProgress permite mudar o valor máximo e dialogSetCurrentProgress permite atualizar o valor do progresso

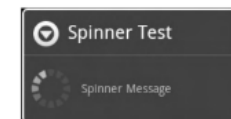
```
droid.dialogSetMaxProgress(4096)  
droid.dialogSetCurrentProgress(2048)
```



Algumas vezes o usuário deve apenas saber que algo está sendo feito

```
droid.dialogCreateSpinnerProgress("Spinner Test", "Spinner Message")
```

Para fechar, chame dialogDismiss



# dialogCreateHorizontalProgress

Não precisa de nenhum parâmetro

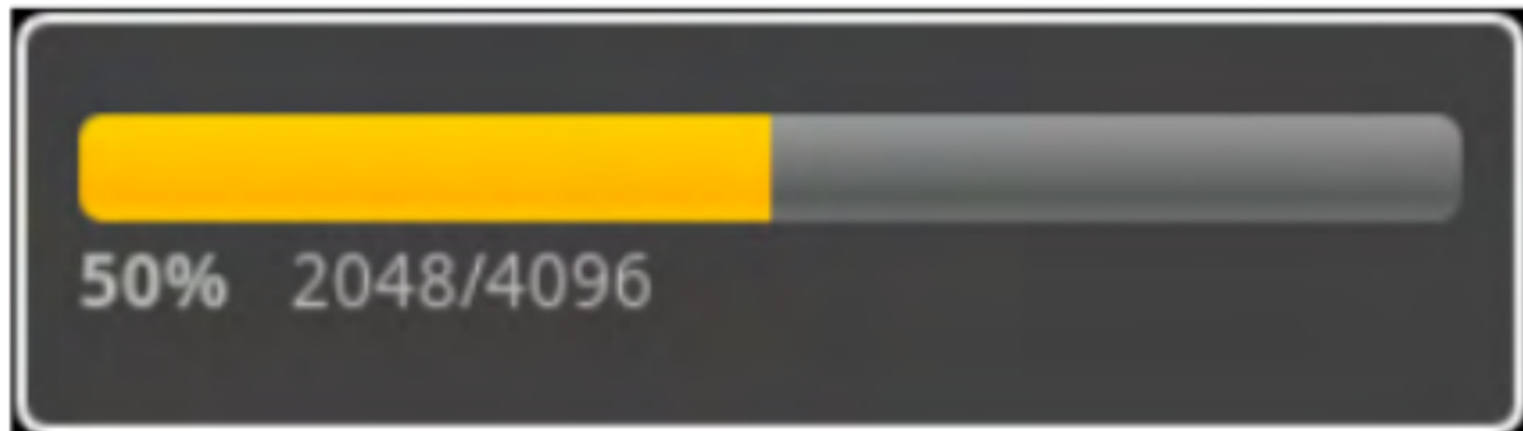
```
droid.dialogCreateHorizontalProgress()  
droid.dialogShow()
```



## Atualizando o progresso

`dialogSetMaxProgress` permite mudar o valor máximo e  
`dialogSetCurrentProgress` permite atualizar o valor do  
progresso

```
droid.dialogSetMaxProgress(4096)  
droid.dialogSetCurrentProgress(2048)
```

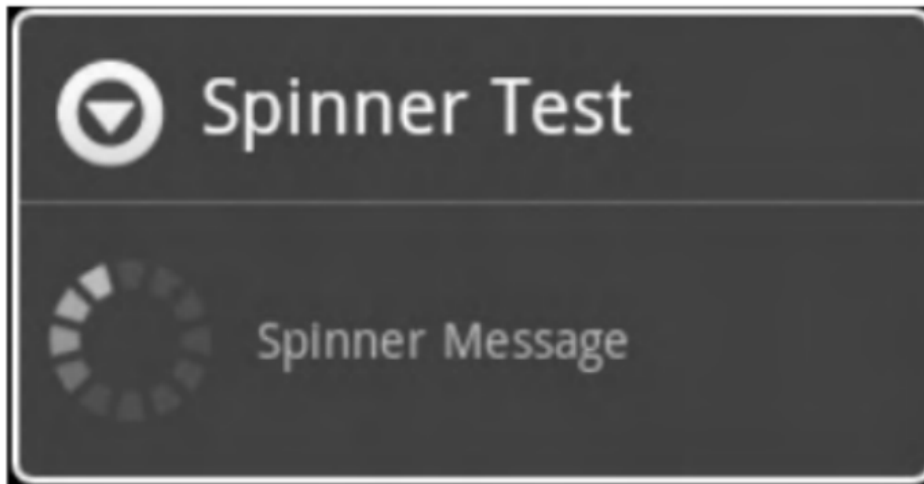




## Algumas vezes o usuário deve apenas saber que algo está sendo feito

```
droid.dialogCreateSpinnerProgress("Spinner Test","Spinner Message")
```

Para fechar, chame `dialogDismiss`



# Mais diálogos

## Modais e não-modais

Diálogos modais bloqueiam e esperam até que o usuário interaja com o diálogo.

```
import android
droid=android.Android()
droid.dialogCreateAlert("I like swords.", "Do you like swords?")
droid.dialogSetPositiveButton("Yes")
droid.dialogSetNegativeButton("No")
droid.dialogShow()
droid.startLocating()
while True: # Wait for events for up to 10 seconds.
    response=droid.eventWait(10000).result
    if response==None: # No events to process. exit.
        break
    if response["name"]=="dialog": # When you get a dialog event, exit loop
        break
print response
```

Gera:



## Continuação

```
# Have fallen out of loop. Close the dialog
droid.dialogDismiss()
if response==None:
    print "Timed out."
else:
    rdialog=response["data"] # dialog response is stored in data.
    if rdialog.has_key("which"):
        result=rdialog["which"]
        if result=="positive":
            print "Yay! I like swords too!"
        elif result=="negative":
            print "Oh. How sad."
    elif rdialog.has_key("canceled"): # Yes, I know it's misspelled.
        print "You can't even make up your mind?"
    else:
        print "Unknown response=",response
print droid.stopLocating()
print "Done"
```

Lista de arquivos com dialogCreateAlert

```
import android, os
droid=android.Android()
list=[]
for dirname, dirnames, filenames in os.walk('/sdcard/s4ta/scripts'):
    for filename in filenames:
        list.append(filename)
droid.dialogCreateAlert('/sdcard/s4ta/scripts')
droid.dialogShow()
file = droid.dialogGetResponse().result
print(list[file])
```

Resultado de quando se pressiona o botão menu



Se o usuário pressionar o botão Sensible, teremos:  
{'name': 'dialog', 'response': 'Sensible', 'data': {'star\_off': 1}}



Options Menu

```
import android
droid=android.Android()
droid.addOptionsMenuItem("Silly", "silly", None, "star_on")
droid.addOptionsMenuItem("Sensible", "sensible", "1", "star_off")
droid.addOptionsMenuItem("None", "none", "0", "star_off")
print "Hit menu to see extra options."
print "Will timeout in 10 seconds if you hit nothing."
droid.webViewShow("/sdcard/s4ta/scripts/blank.html")
while True: # Wait for events from the menu.
    response=droid.eventWait(10000).result
    if response==None:
        break
    print response
    if response["name"]=="options":
        break
print "And done."
```

# Modais e não-modais

Diálogos modais bloqueiam e esperam até que o usuário interaja com o diálogo.

```
import android
droid=android.Android()
droid.dialogCreateAlert("I like swords.", "Do you like swords?")
droid.dialogSetPositiveButtonText("Yes")
droid.dialogSetNegativeButtonText("No")
droid.dialogShow()
droid.startLocating()
while True: # Wait for events for up to 10 seconds.
    response=droid.eventWait(10000).result
    if response==None: # No events to process. exit.
        break
    if response["name"]=="dialog": # When you get a dialog event, exit loop
        break
print response
```

## Continuação

```
# Have fallen out of loop. Close the dialog
droid.dialogDismiss()
if response==None:
    print "Timed out."
else:
    rdialog=response["data"] # dialog response is stored in data.
    if rdialog.has_key("which"):
        result=rdialog["which"]
        if result=="positive":
            print "Yay! I like swords too!"
        elif result=="negative":
            print "Oh. How sad."
    elif rdialog.has_key("canceled"): # Yes, I know it's misspelled.
        print "You can't even make up your mind?"
    else:
        print "Unknown response=",response
print droid.stopLocating()
print "Done"
```



I like swords.

Do you like swords?

Yes

No

## Options Menu

```
import android
droid=android.Android()
droid.addOptionsMenu("Silly","silly",None,"star_on")
droid.addOptionsMenu("Sensible","sensible","I bet.,"star_off")
droid.addOptionsMenu("Off","off",None,"ic_menu_revert")
print "Hit menu to see extra options."
print "Will timeout in 10 seconds if you hit nothing."
droid.webViewShow('file:///sdcard/sl4a/scripts/blank.html')
while True: # Wait for events from the menu.
    response=droid.eventWait(10000).result
    if response==None:
        break
    print response
    if response["name"]=="off":
        break
print "And done."
```

## Resultado de quando se pressiona o botão menu



Se o usuário pressionar o botão Sensible, temos:

```
{u'data': u'I bet.', u'name': u'sensible', u'time': 1301074971174000L}
```

## Lista de arquivos com dialogCreateAlert

```
import android, os

droid=android.Android()
list = []

for dirname, dirnames, filenames in os.walk('/sdcard/sl4a/scripts'):
    for filename in filenames:
        list.append(filename)
droid.dialogCreateAlert('/sdcard/sl4a/scripts')
droid.dialogSetItems(list)
droid.dialogShow()
file = droid.dialogGetResponse().result
print(list[file])
```



**Gera:**



