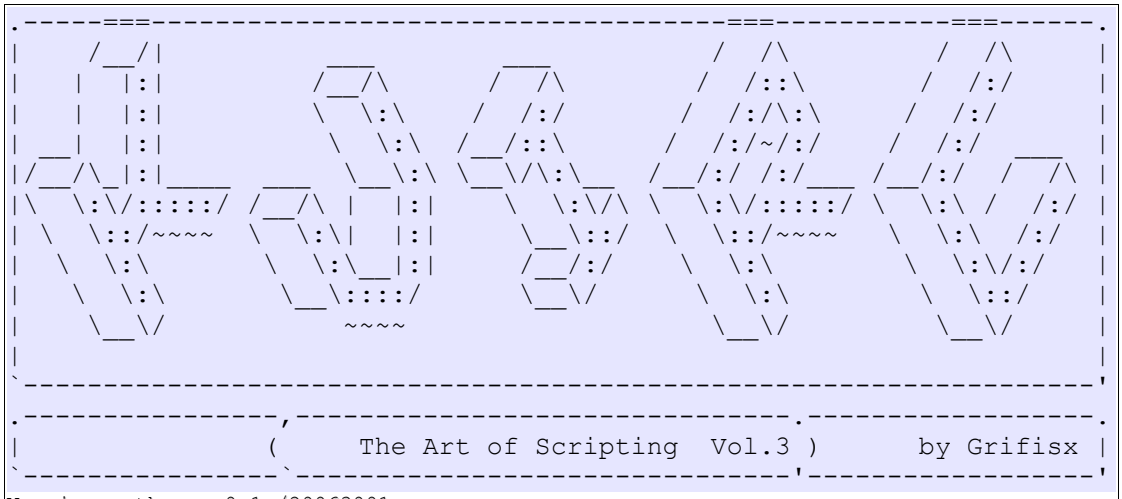

Version: etherea-0.1a/20063001

Events and Popups

Start:

In this tutorial we'll learn about events and have some notes on popups.

Events:
We'll learn how to automatize KVIrc actions, in order to have it behave automatically on some fulfillings.
The IRC world is populated of events; events are situations that can happen, as in the real world thu.
For example, if someone gives out the operator status in any channel (the @ sign), it's a very good thing to thank; all of us once replied (a reaction to the event yet happened, ie: "I am an Op") in such instinctive way saying "<nick> tnx" or "<nick> thanks for op =)".
Now let's see how to make KVIrc work automatically (funny, hu? =D) - please note in many channels automatic scripts are not allowed or appreciated and will happen someone will tell you "*take off the auto greet!*"; be sure you'll learn how to create a script being not that annoying.

Open the Event Editor (Settings->Toolbar->Show Scripting and click on the button with the blue "e" or use the shortcut CTRL+ALT+3) and see what's on there: on the left there's a list with all the events KVIrc reacts on, while on the right side there's a black panel showing the value that $0, $1, $2 -and so on- do acquire on the happening of specified event.
Clicking on the first event on the left side will help understand better: ***OnAccelKeyPressed***
This event turns out for true everytime any "*AccelKey*" is pressed, those are the quick keys KVIrc does recognize; actually are: from F2 to F12, from F1 to F12 holding the SHIFT key, from 0 to 9 holding the CTRL key.

Then *$0* corresponds to the holded key.

Use your brain now: right click on *OnAccelKeyPressed* event and create a "New Handler", then rename the "default" one into something more nice and start writing into the black panel (as done for aliases).

```
If($0=="F2") { echo F2 Key holded down;};
If($0=="F3") { echo F3 Key holded down;};
If($0=="F4") { echo F4 Key holded down;};
```

Or, as for aliases, runu the command
/event(OnAccelKeyPressed,MyEvent1){ If($0=="F2") { echo F2 Key holded down;};}

And you'll see the gestore *MyEvent1* containing the code was added to the *OnAccelKeyPressed* event without opening the Event Editor (syntax for the command is **event(<event>,<gesture_name>{code;};)**; this command is really useful.

Have a chance you want to thank a friend for giving you the Op status on a channel, or a Voice or a HalfOp. Find out in the list these voices:
OnMeHalfOp
OnMeOp
OnMeVoice

Highlight the first one and notice that $0 shows the nickname who did the halfop, $1 shows the username of the person who did the command and $2 shows the host of the person; now let's add a handler (right click and select "New Handler") naming it, for example, "TnxFor".
Now put the code inside the handler, in order that the code itseld will have to be executed once the event goes for true:

```
msg $target \[ $0 \]tnx for the halfop on $target
```

The *$target* function returns the recipient of the current window (nickname of the user if chatting on a query or the name of the channel); in this case it's returning the name of the channel (it could be even be used the function *$chan.name*).
Check the functions behaves in the official handbook, there are several important and useful ones other than *$target, $me* and *$chan.name.*

If you want to improve your event with some pleasant random message, have a look at this code:

```
%tmp=$rand(3)
if(%tmp==0){msg $target \[ $0 \]tnx for the halfop on $target ;}
if(%tmp==1){msg $target Tnx  $0 ^_^ ;}
if(%tmp==2){msg $target Thank you :* ;}
if(%tmp==3){msg $target ..ohoho I'm an halfop ^_^  tnx  ;}
```

What if the owner of channel #00uglyandnasty_r89234 (this channel does not exists – believe it!) is not likely about to love this kind of appreciations?

Just do this in order not to have an auto message on that channel:

```
if($chan.name!="#00uglyandnasty_r89234"){
%tmp=$rand(3)
if(%tmp==0){msg $target \[ $0 \]tnx for the halfop on $target }
if(%tmp==1){msg $target Tnx  $0 ^_^ }
if(%tmp==2){msg $target Thank you :* }
if(%tmp==3){msg $target I'm an halfop ^_^  tnx  to $0}
}
```

This will solve the problem as for the previous condition (*if($chan.name!="#00uglyandnasty_r89234")*) we made it able to thank everything but the channel #00uglyandnasty_r89234.

This can be done for any kind of event such as OnMeBan, OnMeOp and so on.

Now we want to write the text by changing the colors of all the lettersm instead of default white (on black backgrounds) or black (on white backgrounds).
So go on the event: ***OnTextInput***
This event turns true when the return key is hitted and the written text is sent. We need to make our script modifies our simple text to turn it automatically into a colored text, without having to change the color of every letter with the command CTRL+K (color).
If you did read the other tutorials you already know a similar code in an alias (Tutorial ***AOS_Vol.01(Cycle, Conditions and Aliases***) where the script was greeting people in channel with colored text), we'll use the same code in this event, having an eye on *$0* contains the written text (note: this code will not be explained as you can find all the right explanation in tutorial ***AOS_Vol.01(Cycle, Conditions and Aliases***)):

```
%szFrase=$0
%idx=1
while(%idx!=($str.len(%szFrase)+1))
{
      %sztmp =  $str.section(%szFrase,"",%idx,%idx)
      %szFraseColorata=%szFraseColor$k($rand(15))%sztmp
      %idx++;
}
say %szFraseColor  ;
halt;
```

Now hit the "*apply*" button and try to write something in any channel or query (even to yourselves).
As you can see text is now all colored; to disable it and make everything catch its origin just right click on the handler and select "Disable Handler" (or delete it if you don't want it anymore).

Same thing can be done if you'd like to mirror-write, ie on the contrary; here's some code to test:

```
%szFrase=$0
%idx=$($str.len(%szFrase)+1)
while(%idx!=0)
{
      %sztmp =  $str.section(%szFrase,"",%idx,%idx)
      %szFraseColor=%szFraseColor%sztmp;
      %idx--;
}
say %szFraseColor  ;
halt;
```

As you can see the text is back recomposed, starting from the last letter leading up to the first: *%idx* in fact is equal to lenght of text(Frase)+1, so on the very first turn *%idx* will be equal to the position of the last letter of the phrase. Using function

$str.section() it just takes the last one and on the second turn it will be equal to the penult one and so on (as *%idx--* decrements on the contrary of previous example where it was incrementing).

Let's understand it better, following step by step the happening on the text "Hola Grifisx":

```
%szFrase="Hola Grifisx"
%idx=$($str.len(%szFrase)+1)
%idx has been assigned the value of 12: $str.len("Hola Grifisx")+1)
as the text is composed by 12 letters and index starts fromo 0 (so
the +1);
while(%idx!=0)
Until %idx is different from 0;
{
%sztmp =  $str.section(%szFrase,"",%idx,%idx)
On the first turn %sztmp will be equal to $str.section("Hola
Grifisx","",12,12) meanly equal to "x";
%szFraseColor=%szFraseColor%sztmp;
On the first turn %szFraseColorata=""x"; then decrement index;
%idx--;
Looping: on the second turn %sztmp is equal to $str.section("Hola
Grifisx","",11,11) namely equal to "s" and then %szFraseColor="x""s";
On the third turn $str.section("Hola Grifisx","",10,10) namely "s"
and again %szFraseColor="xs""i"
and so on until the end on %szFraseColor="xsifirg  aloh"
}
say %szFraseColor  ;
halt;
```

If you were not able to understand it just stop for a minute and read again, slowly and do some tries. You should have clear in mind some of the most important concepts yet learnt or next script won't be that easy.

Create the alias *lolwritecolor* and put this code inside:

```
class(lolwritecolor,object)
{
      rainbow()
      {
            event(OnTextInput,msgTextColor)
            {
                  %szFrase=$0
                  %idx=1
                  while(%idx!=($str.len(%szFrase)+1))
                  {
                        %sztmp =  $str.section(%szFrase,"",%idx,%idx)

                        %szFraseColor=%szFraseColor$k($rand(15))%szt-
mp
                        %idx++;
                  }
                  say %szFraseColor  ;
                  halt;
            }
      }
      mirror()
      {
```

```
            event(OnTextInput,msgTextColor)
            {
                  %szFrase=$0
                  %idx=$($str.len(%szFrase)+1)
                  while(%idx!=0)
                  {
                        %sztmp =  $str.section(%szFrase,"",%idx,%idx)
                        %szFraseColor=%szFraseColor%sztmp;
                        %idx--;
                  }
                  say %szFraseColor  ;
                  halt;
            }
      }
      disable()
      {
            event(OnTextInput,msgTextColor){}
      }
}
switch($0){
      case("d")
      {
            %tmp=$new(lolwritecolor);
            %tmp->$disable();
            break;
      }
      case("r")
      {
            %tmp=$new(lolwritecolor);
            %tmp->$rainbow();
            break;
      }
      case("m")
      {
            %tmp=$new(lolwritecolor);
            %tmp->$mirror();
            break;
      }
      default{
            echo $k(5) $u() "Usage:"
            echo $k(5) \[ $k(10) $u()"Rainbow mode"$u() $k(5) \]$o -
/lolwritecolor r
            echo $k(5) \[ $k(10) $u()"Mirror mode"$u() $k(5) \]$o -
/lolwritecolor m
            echo $k(5) \[ $k(10) $u()"Disable script"$u() $k(5) \]$o
- /lolwritecolor d
            break;
      }
}
```

First of all we do create a class with 3 functions, those are useful to imprint (first two functions) and to remove (last function) the event. See it without the code:

```
class(lolwritecolor,object)
{
      rainbow()
```

```
	{
		event(OnTextInput,msgTextColor)
		{
			CODE
		}
	}
	mirror()
	{
		event(OnTextInput,msgTextColor)
		{
			CODE
		}
	}
	disable()
	{
		event(OnTextInput,msgTextColor){}
	}
}
```

As you can see the functions are used to insall and to remove command code **event(<Eventname>,<my handler name>){ <code>;}**
It is really very handy as we can install and remove events using a simple command.
By the way, to chose the function to execute, let's use the construct *switch/case* that will allow to have a closer look to the parameter given to the alias, so when handling on "*r*" a new object is created by the class *%tmp=$new(lolwritecolor);* recalling thefunction *$raibow()(%tmp->$rainbow();)* that will install the rainbow-effect code into **OnTextInput** event.
We could have just been putting the whole code in the event and then use a Global Variable (for example: *%ColorMode*) and let the client behave on behalf of its equalty to "MIRROR" or "RAINBOW". Here's the tiny code:

```
if(%ColorMode==”mirror”)
{
	CODE
}
if(%colorMode==”rainbow”)
{
	CODE
}
```

Variable *%ColorMode* can be set within an alias and the "disabled" status wouldn't be needed as in the case the variable wasn't set (just emptying it) in both cases (MIRROR and RAINBOW) conditions wouldn't have been fulfilled and nothing'd have been happened.

All this means waste checks (putting the event the client should check the value of that variable in every input) and easypeasy.. we do like hard stuff =D.

On Popups:
It's time to have big time with menus you can see right clicking on any window in KVIrc (or in the nicklist or upon a nickname).

Having a look again on the previous script you can see it is a bit

awkwardy typing everytime, for example, "*/lolwritecolor r*" to use the RAINBOW mode and "*/lolwritecolor d*" to disable it. It would be more handy right clicking on the channel window and have a very own cascading menu where to choose to write in rainbow color or in mirror mode or to disable the script itself.

Let's open the "*Popup Editor*" (*Scripting->Edit Popups* or use the shortcut CTRL+ALT+4 or use the "P" button in the scripting bar) and create our new cascade menu: right click in any empty space in the popup list, select "New Popup" and then change the name using the field on the upright (note: do not hit the TEST button, the popupname will change by itself). Name it "*wColor*" or whatever you may like best.
Now we need to create three elements inside the popup: on the right side, in the white upper panel, right click and select "*New Element Below*" (repeat for three times), then highlight them and rename them (using the first white field in the near below side labeled "*Text:*"): "*Write Rainbow*", "*Write Mirror*" and "*Disable Script*".
If you want you can choose three icons (to see all the available icons run: Tools->Show Icon Table) and set the identificative number (or its path) in the field "*Icon:*".

Once you're done let's wite the code that should be executed when an element of the menu is selected.
Select the element you wish to write the code for, for example "*Write* Rainbow", then in the black box right down, put the code that will run the alias for writing in rainbow mode:

```
echo Mode Rainbow Writing Active.
lolwritecolor r
```

The same for "*Write Mirror*" and for "*Disable Script*":

```
echo Mode Mirror Writing Active.
lolwritecolor m
```

And:

```
echo Write Color Disabled.
lolwritecolor d
```

As you can see popups work as aliases, they can house every kind of code and the advantage is the code will be executed only when the menu item is selected and there's no need to recall it typing */aliasname*.
Now this menu must appear when right clicking on channel windows, near the default KVIrc itemlists.

Highlight the popup **channeltextview** and create, on the very end of all alements, a new "*New Separator*" and then "*New External Menu*" anc rename it in "*Write Color*" (using the field "*Text:*") and do associate the external menu already created by writing in the "*External Menu*" field the name we chose (for example *wColor*) or associating even an ID (writing a random number such as 1006969001 in the "Element ID*:*" field) that result pretty useful if we would like to redistribute our script as an addon with install/deinstall utility..

And now in any window channel right click and.. enjoy your menu.

Now type:
/ECHO STOP

- - - - - - -- - - - - - - - -- - - - - - - - - -- - - - - -- - - - -
"You see things; and you say `*Why?*' But I dream things that never were; and I say `*Why not?*"
(George Bernad Shaw)
- - - - - - - - -- - - - - - - - -- -- - - - - - - - - - - - - - - - -
Grifisx