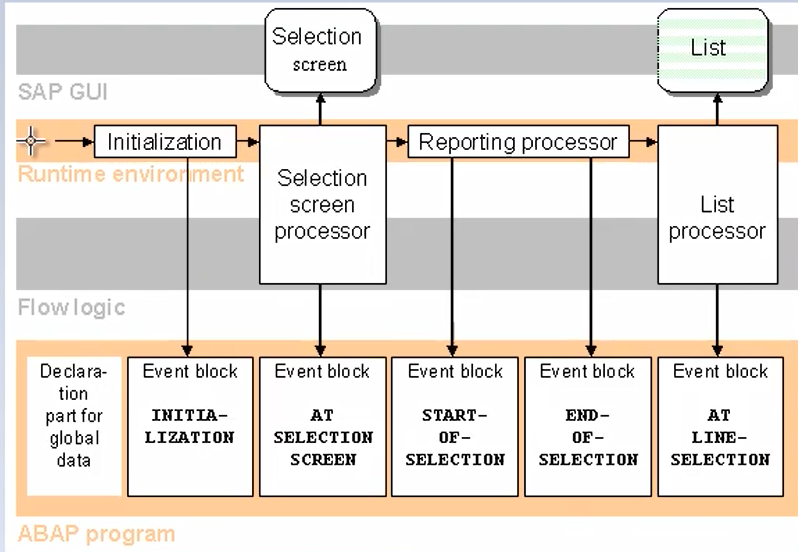
For us to be able to build our selection screens and use selection screens in our program, the first thing we must understand are things called events. Events are what we call processing blocks. And what I mean is just a section of code that's specific for the selection screens.

Now the structure of an event starts with the event keyword but doesn't have an ending event keyword. The end of the event block of code is just implicit, because the start of the next event block will terminate the first one.

So, what I mean is, the event that we define will end when a new event starts, such as a sub program, or our report ends.



We're at the end of our code. Now, when we run our executable programs, the program is controlled by a predefined process in the runtime environment. And a series of processes are called one after another. These processes trigger events for which we can define event logs for in our program.

When we start our program, there are certain events that get fired in a certain order.

And you can see we have a selection screen. And then we have our list output. Now when our program starts, the first thing the system does is check your program to see if you have included any processing blocks. And if you have, it'll determine which ones you are using and will follow this sequence of events to execute those blocks. So, we can see when the program starts, our program global variables are declared, so the system reserves memory space.

And then the runtime environment will look to see if you have an initialization event block, and if you do, this event block will be executed.

Now bear in mind, initialization event block will only be run once and that's at the start of your program and the things you include in the initialization block of code are things like setting up initial values for fields in your selection screen.

The runtime environment will then check whether a selection screen has been defined in your report. So, if at least one input field is being used, then control will be passed to the selection screen processor.

Now, this selection screen processor will display the screen to the user, and once it is shown, the user can then interact with the screen, filling fields, select data from drop down boxes, etc. And when the user clicks on execute to run the report after filling in all the fields, a screen processor will then trigger the at selection screen event block.

And this is where you can write your own code to check all the entries the user has keyed into the individual fields on the selection screen. If the user has entered any incorrect values, your code can capture these mistakes and instead of processing your program, you can force the selection screen to be displayed again to the user, to allow them to correct any errors.

Now of course, if they make some errors, you need to tell the user where the errors are, so you will use error messages to inform the user they need to check the entries they have made and key in some correct values before the program can be run.

Now this type of action can just run round in a big loop. If the user keeps making mistakes, you can capture those mistakes and present the screen again and again. Once the user has no errors, or at least your code hasn't captured any errors, then the runtime environment passes control to the start of selection event block. This event block can hold your code that you create for setting up the values of internal tables, internal fields used within your program, and maybe writing out header information on the report. Now there are other event blocks, as you can see, in the rest of this diagram. And there are many that don't appear on the diagram. But the ones we have been through are the main event blocks used to display a selection screen to capture user input, which we can then use to process the rest of our program. Once all the event blocks that you have defined in your program have been processed, the ABAP processor will then execute the rest of the program and pass control to the list processor to output your report, which is this section right here. So once the program's done everything It can send the output to the screen for the user to see. Now just say, you see this down arrow here. I would just explain that one. Up to now, we've been producing output on the screen for the user to read. Well, these list screens can be interactive as well. So, for example, if you wanted to drill down on a report. You can double-click a line or double-click an individual field. And the list processor will capture the action and fire the at selection screen event block. And that's where you can include more code to determine what the user clicked on and then carry out some other activity, like going to read some more tables to find some more detailed information that you can then output to the screen. But we're not going to dig into that. We're going to focus on the selection screen and making sure the user enters the right values for our report. And make yourself a selection screen as a good interface using various text boxes and radio buttons that make our program look a bit more inviting.