A small demonstration to start with

We can type some simple text, and then add some Stata code; but note that all input is together and all output is together

```
Stata:
sysuse auto
list make mpg if mpg < 18, sepby(foreign)
```

Adding graphs works easily, also

```
Stata: fig, figfmt=eps
twoway (lfitci mpg price)(scatter mpg price), by(foreign) ///
  legend(order(3 "Domestic" 4 "Foreign"))
```

Complicated steps can be added and not echoed in preparation for another picture

```
Stata: hide, !echo
webuse lbw
logit low age lwt smoke ui
```

There is no problem splitting code into separate blocks, because they are all running in the same Stata session in the background.

```
Stata: fig, figfmt=eps, hide, !echo
lroc, title(A diagnostic model for low birthweight) caption(Source: Hosmer and
Lemeshow)
```

Now, we'll change the look of the output. This particular choice is done for emphasis.

```
Stata: outstyle=Yellow Bold, prompt="stata . "
```

```
Stata:
sysuse auto, clear
Stata:
list make mpg if strpos(make,"Olds")
```

If we would like to go back to defaults, this is simple enough:

```
Stata: -outstyle, prompt=". "
```

Here we will show the command, but save the output for later. The -xi- construct in Stata is used for categorical variables:

```
Stata:label=xi, saveout
xi: regress mpg weight i.rep78
```

If you look at the output, you will see that the lowest-numbered class is used as the reference class by default:

```
out:xi
```

This is the one advantage of splitting input and output.