**Video Script: Section 6 Video 3 – rendering text**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Description | Action on screen | Narration |
| 1 | Introduction  (Outcome and why it is desirable)  This should give the viewer an idea of the outcome of the task at the beginning of the videos and set the stage and expectations of the viewer. | Refer to PPT | In this video, we’re going to see how to generate textual content on the webpage. |
| 2 | Context(Problem/Solution)  Present the viewer with a real-world solution and how the situation would pose as a challenge. It always helps to draw the viewer's attention using a use-case. Metadata template can be used here. |  | It is often useful to show textual information to the user, e.g. a table of values or some statistics. |
| 3 | Guidance (How to do it and how it works): | Open RStudio  Run in the console:  library(“shiny”)  runApp(“activity\_06\_03”) | Launch Rstudio, go to the folder 'section 6' and run in the console:  library(“shiny”)  runApp(“activity\_06\_03”) |
| 4 |  | An interactive webpage appears in a browser.  A description...  Play with the controls. | Choose a dataset and a number of observations to see the content changing.  You can see three types of textual outputs in this page:   * Free text * Formatted text, as you would see it the console for example. * A formatted table. |
| 5 |  | Open UI.R and server.R in the editor: click on the folder activity\_06\_03 and click on UI.R and server.R | Open UI.R and server.R in the editor. |
| 6 |  |  | In the previous video, we saw the function *plotOutput* in UI.R, used together with *renderPlot* in server.R, which was used for sending a graph to the webpage. |
| 7 |  | Go to the source of UI.R | This time, we use textOutput in UI.R |
| 8 |  | Go to the source of server.R | together with renderText in server.R to display free text.  The expression in renderText must be anything that can be passed to the R function cat(), typically a string or a vector of strings.  Note that the property of the output in server.R: introduction, matches with the argument of the textOutput in UI.R |
| 9 |  | Go to the source of UI.R | To capture the content of the console, for example the output of a function, we use verbatimTextOutput in UI.R |
| 10 |  | Go to the source of server.R | together with renderPrint in server.R. |
| 11 |  | Go to the source of UI.R | For tables, you need to use a combination of tableOutput in UI.R |
| 12 |  | Go to the source of server.R | And renderTable in server.R, to generate a formatted table, from a data frame or a matrix. |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 | Conclusion:The video concludes by showing the viewer that the goal has been achieved, and reminding them why they should be happy about that. A PowerPoint summary slide with the key points emphasized would make it easier for the viewer to remember what was covered in the video | Back to PPT | In this video you have seen how to generate and display textual content, either as free text, formatted text or in a table, by using three different pairs of functions: (renderText, textOutput), (renderPrint, verbatimTextOutput) and (renderTable, tableOutput).  In the next video, we’ll see in more details how Shiny works, with reactive programming. |