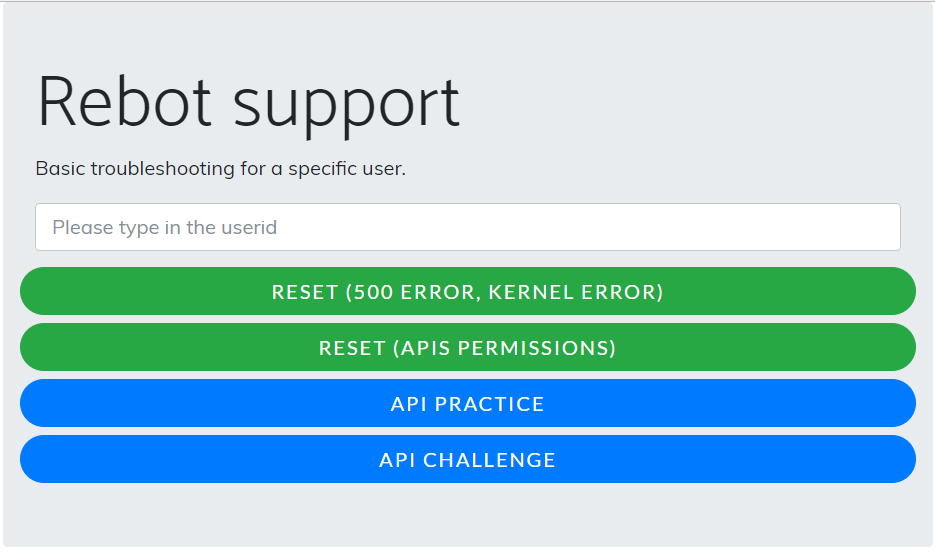
Instructions to Trainers:

* This is just a guide. Use it for reference during the hands-on sessions.
* Website: <https://rebot.chat>
* Training notes are available on the website.
* All relevant links for the workshop are on the website above.
* Good idea to hands-on the following items before the workshop:
  + Telegram – install on your phone
  + Repl.it – refresh some python syntax
  + Jupyter – Try Level 02 & Level 04 to get a feel of the interface.
* Support: <https://rebot.chat/support.html>. Visit the support page on your phone / laptop. If a user cannot access jupyter - ask for his userid, then click on the RESET (500 Internal Server Error) / API.
* Can /deleteprofile to remove a user’s session

* Test APIs: Test that a user’s API is working with the blue buttons. Else they will definitely not see their changes in Telegram.



|  |  |
| --- | --- |
| Team A (39TR2):  Kai Jun, ISPVMF  Javan  Su Ming | Maji  Dennis  Andrew Young  [Sang] Guest visit |
| Team B (39TR1):  Raymond, ISPVMF  Weihan  Wisanto | Gabriel Wong  Thomas See - done  Yash – done [Sang] Guest visit |
| Team C (38TR2):  Gabriel Loh, ISPVMF  Danny  Varun | Pjay Magat  Maheen Syed  Jianlin Luo  [Sang] Guest visit |

**Level 01: Basic Python (REPL.IT)**

* Part 1: Give them an overview (see slide)
* Go to rebot.chat -> select Level 1
* Login as anonymous
* Left-hand side code. Right-hand side output.
* Run your code & save your changes.
* Part 2: Hello World

|  |
| --- |
| print(“hello world”)  print(“hello <your own name>”) |

Explain that this is a code ritual. Most developers start learning by writing the piece of code.

* Part 3-A: Variable

|  |
| --- |
| name = “John”  print(“Hello “ + name)  print(“Hello “ + name + “ and ” + name) |

* Part 3-B: Variable Input

|  |
| --- |
| print(“Please type in your name:”) # eg. To improve the user experience  name = input()  print(“Hello “ + name) |

* Part 4-A: Different variable types (string, int)

|  |  |
| --- | --- |
| #Wrong  first = “5”  second = “6”  sum = first + second  print(“Sum = ”, sum)  # Ans: 56 – treated as string | #Right  first = 5  second = 6  sum = first + second  print(“Sum = “, sum)  # Ans: 11 – treated as number |

* Part 4-B: Different Variable Inputs

|  |  |
| --- | --- |
| #Wrong  print(“Please key in two numbers”)  first = input()  second = input()  sum = first + second  print(“Sum = ”, sum)  # Ans: 56 – treated as string | #Right  print(“Please key in two numbers”)  first = int(input())  second = int(input())  sum = first + second  print(“Sum = “, sum)  # Ans: 11 – treated as number |

* Part 5: IF-ELSE. Continue from 4-B

|  |  |
| --- | --- |
| #Wrong  print(“Please key in two numbers”)  first = input()  second = input()  sum = first + second  if sum >= "100":  print("Sum is 100 or greater")  else:  print(“Sum is less than 100")  # “100” should be 100, without double quotes  # Error: TypeError: '>=' not supported between instances of 'int' and 'str' | #Right  print(“Please key in two numbers”)  first = int(input())  second = int(input())  sum = first + second  if sum >= 100:  print("Sum is 100 or greater")  else:  print(“Sum is less than 100") |

**Level 02: Basic Chatbot (Using Jupyter)**

* Part 1: Give them an overview (see slide)
* Click on Practice
* Green buttons – “Try your Code” & “Send to Rebot”
* Need Telegram installed
* Make sure most people can see the files before moving on. Let facilitators help out if cannot.
* Part 2: Simple hello world

|  |
| --- |
| import rebot  def send\_to\_rebot(input):    message = rebot.createEmptyMessage()  message.addText("Hello World!")  return message |

* Part 3: Type your name in Telegram

|  |
| --- |
| import rebot  def send\_to\_rebot(input):    message = rebot.createEmptyMessage()  message.addText(“Hello ” + input + “!”)  return message |

* Part 4: If-Else

|  |
| --- |
| import rebot  def send\_to\_rebot(input):  if "movie" in input "best" in input:  response = “Star Wars”  else:  response = “No clue..”  message = rebot.createEmptyMessage()  message.addText(response)  return message |
|  |

**Level 03: Basic Html (Using REPL.IT)**

* Part 1: Give them an overview (see slide)
* example.html They can view example.html from <https://rebot.chat/example.html>
* Fork - make a copy
* Make sure most people can see the files before moving on. Let facilitators help out if cannot.

\*\* FYI:

* [repl.it](http://repl.it) blocks connections to external sites. That is why we cannot scrap directly from a real site.
* rebot.chat is on https, not http.
* Part 2: Go through a 5-step explanation. Stick to “Containers” concept

|  |  |  |
| --- | --- | --- |
| Step 1 | Import libraries  Common libraries developed for reusability to avoid code redundancy | import rebot |
| Step 2 | Document filename  eg. myreport.doc | filename = "example.html"  print(filename) |
| Step 3 | Extract the raw content from the document. ie. byte code | content = rebot.getContent(filename)  print(content) |
| Step 4 | Format the content into a readable HTML format that the code understands. | html = rebot.getHtml(content)  print(html) |
| Step 5 | Search data | data = html.search(“p”)  print(data) |

* Part 3: Above will cover the basics for Level 03.

But that’s too easy, ask them to search for:

|  |
| --- |
| data = html.search(“span”)  print(data) # this is the wrong answer. they need to specify the class  data = html.search(“span”, “class”, “target”)  print(data) |

* Part 4: Explore inputs(). Depending on time, allow them to modify example.html and test their changes

|  |
| --- |
| print(“please enter a filename:”)  filename = input()  print(filename) |

**Level 04: Team Challenge**

* Part 1: Recap (see slide)
* Container concept
* Make sure everyone can see the files before moving on.
* Part 2 - Example A: Get company data (see slide)

|  |
| --- |
| import rebot  def send\_to\_rebot(ticker):    content = rebot.getContent(ticker)  html = rebot.getHtml(content)  **stockName = html.search("h1")**  message = rebot.createEmptyMessage()  message.addText("Here are the results from Yahoo Finance!")  **message.addData("Stock Name", stockName)**  return message |

* Part 3 - Example B: Get beta (see slide)

|  |
| --- |
| import rebot  def send\_to\_rebot(ticker):    content = rebot.getContent(ticker)  html = rebot.getHtml(content)  **beta = html.search("td", "data-test", "BETA-value")**  message = rebot.createEmptyMessage()  message.addText("Here are the results from Yahoo Finance!")  **message.addData("Beta", beta)**  return message |

* Part 4 - Challenge answers

\*\* It is possible that they can get the correct answers by using other identifiers. So, check that the values match with the website information.

|  |
| --- |
| import rebot  def send\_to\_rebot(ticker):    content = rebot.getContent(ticker)  html = rebot.getHtml(content)    **## Challenge 1 (Company name & price)**  **stockName = html.search("h1")**  **tickerPrice = html.search("span", "class", "Trsdu(0.3s) Fw(b) Fz(36px) Mb(-4px) D(ib)")**  prevTickerPrice = html.search("span", "class", "C(black) Fz(24px) Fw(b)")  **## Challenge 2 (Company ratios \*\*These are nested tags)**  **beta = html.search("td", "data-test", "BETA-value")**  **peratio = html.search("td", "data-test", "PE\_RATIO-value")**  **eps = html.search("td", "data-test", "EPS\_RATIO-value")**  **## 3-min Speed Round (Tie-breaker – check with gamemaster)**    previousClose = html.search("td", "data-test", "PREV\_CLOSE-value")  openPrice = html.search("td", "data-test", "OPEN-value")  bid = html.search("td", "data-test", "BID-value")  ask = html.search("td", "data-test", "ASK-value")  dayRange = html.search("td", "data-test", "DAYS\_RANGE-value")  volume = html.search("td", "data-test", "TD\_VOLUME-value")  avgVolume = html.search("td", "data-test", "AVERAGE\_VOLUME\_3MONTH-value")  marketCap = html.search("td", "data-test", "MARKET\_CAP-value")  earningsdate = html.search("td", "data-test", "EARNINGS\_DATE-value")  fwdDividend = html.search("td", "data-test", "DIVIDEND\_AND\_YIELD-value")  exDividend = html.search("td", "data-test", "EXDIVIDEND\_DATE-value")  yearTarget = html.search("td", "data-test", "ONE\_YEAR\_TARGET\_PRICE-value")  **## Challenge 3 (If-else)**  if float(beta) >= 1.5:  recommend = "OMG! TOO RISKY"  else:  recommend = "MODERATE RISK"  **## Challenge 4 (Latest news title)**  latestNews = html.search("a", "class", "Fw(b) Fz(20px) Lh(23px) LineClamp(2,46px) Fz(17px)--sm1024 Lh(19px)--sm1024 LineClamp(2,38px)--sm1024 Td(n) C(#0078ff):h C(#000)")  **## Return results**  message = rebot.createEmptyMessage()  message.addText("Here are the results from Yahoo Finance!")  message.addText(latestNews)  message.addData("Recommendation", recommend)  message.addData("Stock Name", stockName)  message.addData("Ticker Price", tickerPrice)  message.addData("Previous Ticker Price", prevTickerPrice)  message.addData("Previous Close", previousClose)  message.addData("Open Price", openPrice)  message.addData("Bid", bid)  message.addData("Ask", ask)  message.addData("Day Range", dayRange)  message.addData("Volume", volume)  message.addData("Average Volume", avgVolume)  message.addData("Market Cap.", marketCap)  message.addData("Beta", beta)  message.addData("PE Ratio", peratio)  message.addData("EPS", eps)  message.addData("Earnings Date", earningsdate)  message.addData("Forward Dividend", fwdDividend)  message.addData("Ex Dividend", exDividend)  message.addData("1st Year Target", yearTarget)    return message |

