

Top 实践报告

13331038 程键生

对 TOP 的理解:

1. 面向的 thinking 是抽象的，不具体的，接近人类思维的想法。这种编程方法有赖于各种简洁和“语义化”的编程语言。
2. 自上而下模块化的实现，清晰的流程和逻辑。

对 TOP 的尝试:

以下是 S5 的 coding 过程

First Thinking: make them clickable

```
index.ls
1  $ ->
2  add-clicking-handler-to-all-buttons !-> Robot.click-with-message!
3  add-clicking-handler-to-the-bubble!
4  add-resetting-when-leave-app!
5
```

```
add-clicking-handler-to-all-buttons= (next-step) !->
  add-clicking-to-a-handler !-> next-step!
  add-clicking-to-b-handler !-> next-step!
  add-clicking-to-c-handler !-> next-step!
  add-clicking-to-d-handler !-> next-step!
  add-clicking-to-e-handler !-> next-step!
```

Further, I need two class to store statics and some general functions

```
#each "button" has its handler and I need a Robot to call them one by one
class Robot
  @buttons = $ '#control-ring .button'
  @bubble = $ '#info-bar'
  @sequence = [0 to 4]
  @char-sequence = ['A' to 'E']
  @cursor = 0
  @state = 'unclicked'

  /* this function is used to start the click in-order as well as the callback function of a click*/
  @click-cur-button-and-get-next=!->
```

```

class Button
  @buttons = []
  @disable-unclicked-buttons = (this-button)-> [butt
  @enable-unclicked-buttons = (this-button)-> [butto
  @all-button-is-done = ->
    [return false for button in @buttons when butt
    return true
  @reset-all = !-> [button.reset! for button in @but

  (@dom, @good-message, @bad-message, @callback) !->

```

In_detail: the process after a click is described in class Button

```

@dom.click !~>
  if @state is 'enabled'
    @@@disable-unclicked-buttons @
    @wait!
    @fetch-number-and-show!

```

```

fetch-number-and-show: !->
  $.get '/?timestamp='+ Math.random!, (number, result)!~>
    if (@state is 'waiting')
      @red-dot .text number
      @@@enable-unclicked-buttons @
      @current-sum+= parse-int number
      @done!
      @bubble-check!
      @error-check @current-sum

```

As well as the Robot

```

@click-at-random-order=!->
  if @cursor is @sequence.length
    set-timeout !->
      Robot.bubble.click!
      $ ('#message-bar .saying') .text "大气泡：楼主异步调用战斗力感人，目测不超过"+Robot.sum
      $ ('#info-bar') .text Robot.sum
    , 0
  else
    @state='ran-clicked';
    set-timeout !->
      Robot.click-cur-button-and-get-next-randomly!click!
    , 800

```