

Trick - Return CPU to UI Thread

- 在前端Runtime做耗时的任务，会阻碍UI的响应，整个页面卡死
- 需要把控制权交还给UI Thread
 - `(1..100).each {|x| many very heavy calculation }`
- `==>`
 - `def single_calc(x)`
 - `one very heavy calculation`
 - `setTimeout(->() { single_calc(x+1) }, 1)`
 - `end`

Trick - Delayed Evaluation

- 如果控件拖动，需要更新非常耗时的运算，那么每次拖动小步都会计算从而让拖动的用户体验非常卡。
- 解决方法是，合并拖动的行为，让用户拖动到目标位置，用户停下来不再拖动之后，再进行耗时的计算
- ```
on_change () {
 • clearTimeout($saved_timeout)
 • $saved_timeout = setTimeout (->() { very heavy calculation }, 1000)
}
```

 # 用户1000ms没有动这个控件，就开始做耗时计算