

Avaliação tardia

Quando unimos o poder do *Closure* com o do *Currying*, é possível ganharmos performance através do princípio da avaliação tardia. As funções que fazem uso desse princípio são chamadas de **Funções Lazy**.

Através da avaliação tardia, é possível antecipar todo o processamento que já é possível ser executado, e adiar o que não pode ou não é possível de ser executado naquele momento. Dessa forma, ganha-se tempo.

```
1  function eager(a, b) {
2      // TRECHO apenas para simular um processamento demorado
3      // ignorem os detalhes de implementação
4      const fim = Date.now() + 2500
5      while(Date.now() < fim) {}
6      // FIM DO TRECHO
7
8      const valor = Math.pow(a, 3)
9      return valor + b
10 }
11
12 function lazy(a) {
13     // TRECHO apenas para simular um processamento demorado
14     // ignorem os detalhes de implementação
15     const fim = Date.now() + 2500
16     while(Date.now() < fim) {}
17     // FIM DO TRECHO
18
19     const valor = Math.pow(a, 3)
20     return function(b) {
21         return valor + b
22     }
23 }
24
25 console.time('#1')
26 console.log(eager(3, 100))
27 console.log(eager(3, 200))
28 console.log(eager(3, 300))
29 console.timeEnd('#1')
30
31 console.time('#2')
32 const lazy3 = lazy(3)
33 console.log(lazy3(100))
34 console.log(lazy3(200))
35 console.log(lazy3(300))
36 console.timeEnd('#2')
```

