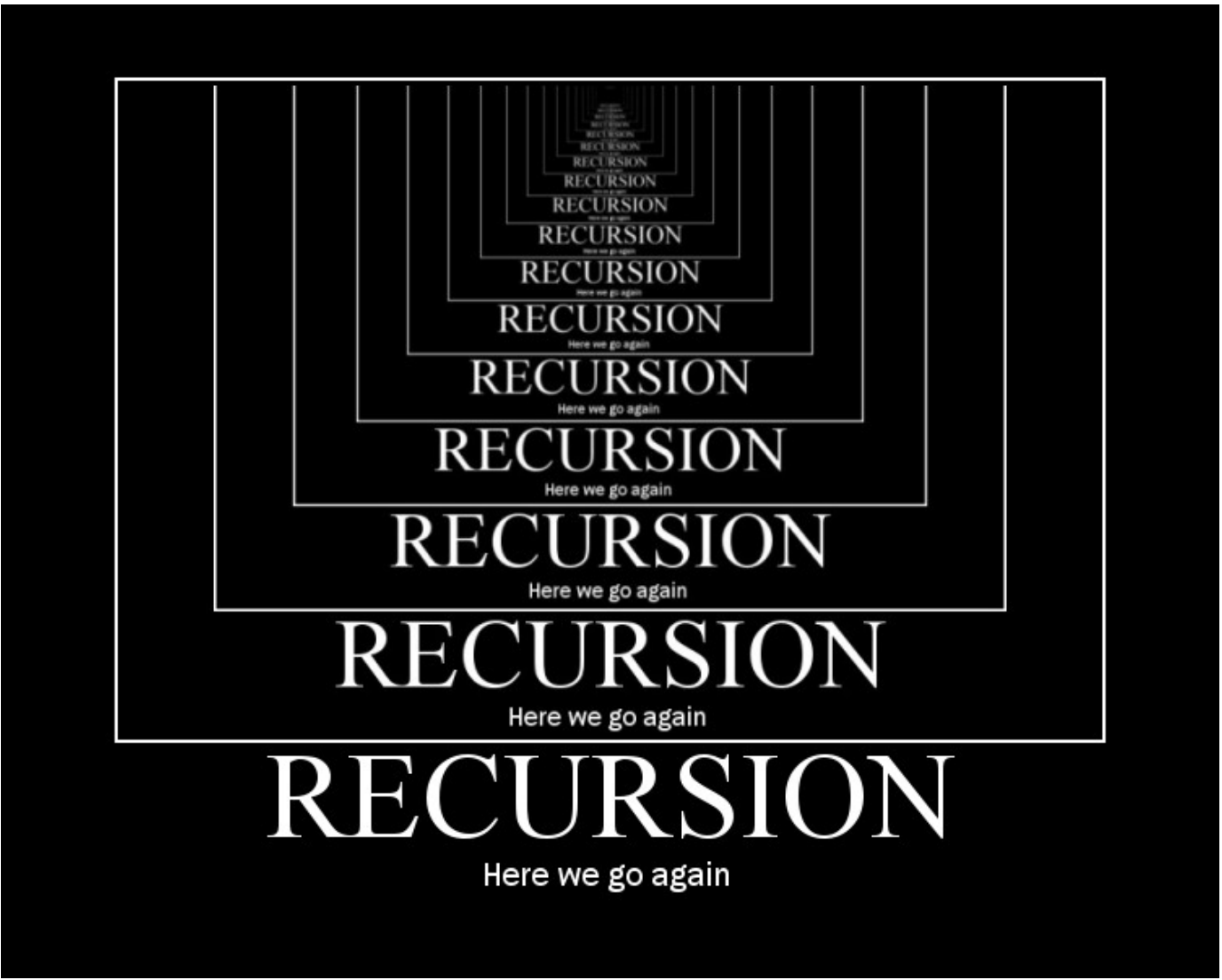


Recursion EXPLAINED!

Katakanlah kita melihat satu method yang rekursif

```
function recurse_me(times) {
  if (times == 0) {
    console.log("Base Case 0!")
  } else {
    console.log(`Before: ${times}`)
    recurse_me(times-1)
    console.log(`After: ${times}`)
  }
}
```



Gimana cara kerjanya?

Coba kita panggil method recurse me: recurse_me(3)

Ketika times = 3

```
if (times == 0) {
  console.log("Base Case 0!")
} else {
  console.log(`Before: 3`)
  recurse_me(3-1)
  console.log(`After: 3`)
}
```

Ketika times = 2

```
if (times == 0) {
  console.log("Base Case 0!")
} else {
  console.log(`Before: 2`)
  recurse_me(2-1)
  console.log(`After: 2`)
}
```

Ketika times = 1

```
if (times == 0) {
  console.log("Base Case 0!")
} else {
  console.log(`Before: 1`)
  recurse_me(1-1)
  console.log(`After: 1`)
}
```

Ketika times = 0

```
if (times == 0) {
  console.log("Base Case 0!")
} else {
  console.log(`Before: ${times}`)
  recurse_me(times-1)
  console.log(`After: ${times}`)
}
```

Ketika kita memanggil recurse_me, sebelum mencapai 0 (times == 0), kita kembali melakukan panggilan ke recurse_me dengan nilai yang dikurangi satu. Dan seterusnya hingga times mencapai angka 0 sebagai treshold-nya.

Cala lain untuk menggambarkan eksekusi dari recurse_me(3)

