Replacing Iteration with Recursion



Gerald Britton
IT Solutions Designer

@GeraldBritton www.linkedin.com/in/geraldbritton

Foundations



Mathematical foundation

E.W. Dijkstra – *Recursive Programming*

- http://oai.cwi.nl/oai/asset/9253/9253A.pdf
- *"Go To Statement Considered Harmful"*

$$F_n = \begin{cases} 0 & if\ n = 0 \\ 1 & if\ n = 1 \\ F_{n-1} + F_{n-2} & if\ n > 1 \end{cases}$$

Recurrence

$$F_{10} = F_9 + F_8 = F_8 + F_7 + F_7 + F_6 = \ \ldots$$

$$S_n = \begin{cases} 0 & if\ n = 0 \\ n + S_{n-1} & if\ n > 0 \end{cases}$$

Demo



Summing numbers using recursion

Tail Recursion



Avoid stack exhaustion

Function f calls function g at the end

If function g leads to f ⇒ f is tail recursive

Function f and g may be the same

Turn recursion into iteration

Only one stack frame per function

Demo



Summing using tail recursion

Trampolining



Python does not optimize tail calls

Need workaround

Trampolining!

~~GO TO~~

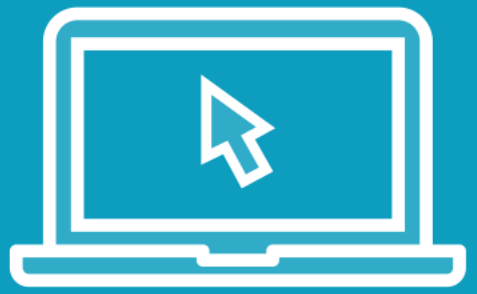
Generators!

Demo



Alex Beal's `tramp` function

Demo



- **Add Employee class**
- **Populate with names**
- **Find nth-over manager**
- **Use `tramp` function**

Summary



Recursion in Python

- Tail call recursion
- Trampolining

Costs

- Extra object creation and destruction
- Loss of traceback information

Should you use recursion?

- It depends!