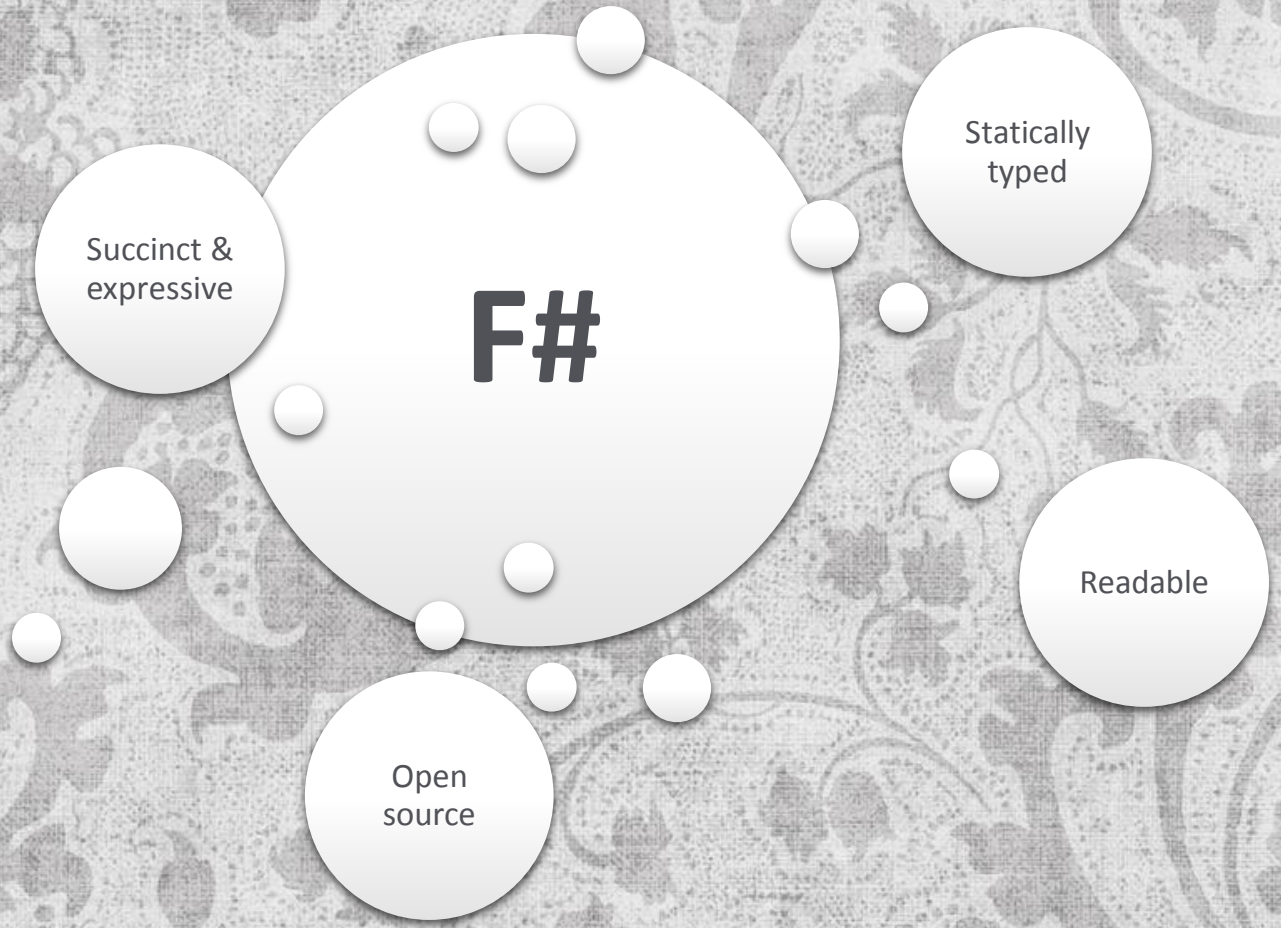
# Try F#:
# From Zero to Data Science

Rachel Reese | @rachelreese | rachelree.se | github: rachelreese

# Agenda

| | |
|---|---|
| **1:00** | Intro and Setup |
| **1:15** | **Set 1**: Getting Started in F# |
| **1:50** | Intro to Type Providers |
| **2:10** | **Set 2**: Data Visualization |
| **2:45** | Break |
| **2:55** | **Set 3**: Data Science |
| **3:30** | Challenge! |
| **4:05** | Wrap up |
| **4:15** | Fin! |

# Why F#?


F#
Statically typed
Readable
Open source
Succinct & expressive

# Set up

- Pair Up!
- Recommend Chrome or IE
- Explore Try F# site.

# Set 1: Getting Started

**Bindings and values and REPLs! Oh my!**

- Print a statement with a float.

**Fun with Functional Functions**

- Add to the toHackerTalk function by changing e -> 3 and l -> 1.

**Chaining Functions with the Forward Pipe Operator**

- Find the difference between the max high and the min low for all the days.
- Find the difference between open and close on the day with the highest volume.

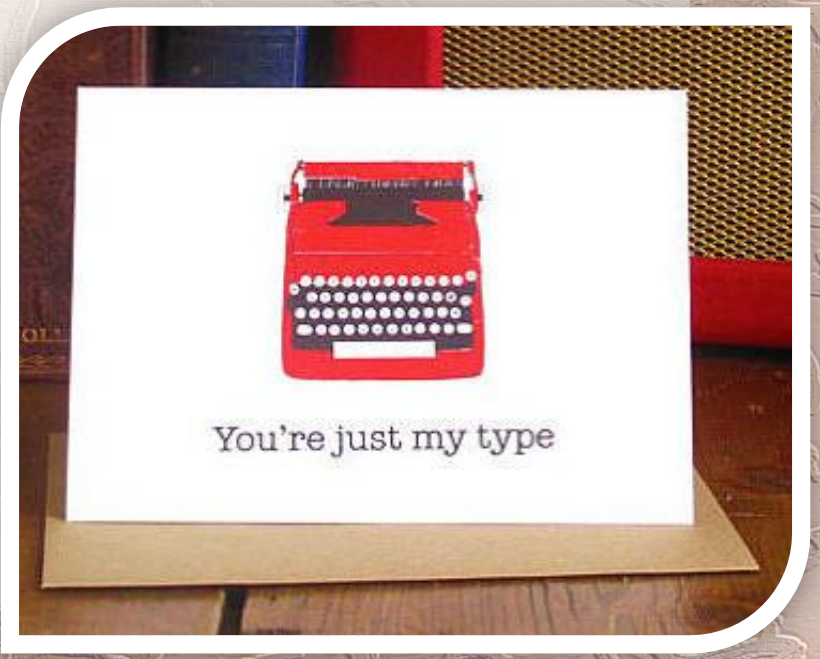
**Using Data Structures to Create Larger Programs**

- Add a SuperLeaf to powerUp; test handlePowerUp.


Type Providers
TYPE ALL THE DATA

# What are type providers?

A mechanism to provide types to the compiler.



For me, part of the role of F# is about proving that statically-typed languages can play fully in the modern world of connected programming, without losing the **simplicity, elegance or tooling** that come with strong types.

Type providers are an essential part of tackling this, because we can no longer ignore the **information-richness of external data sources**, and have to change language and compiler architecture to adapt.

- Don Syme

# Why do we need them?

# Benefits

Intellisense, tooltips, & other tooling available

No code generation

- Always in sync with the source
- No extra bloated code
- Scalable to millions of types (e.g. - freebase)

More natural with REPL

"Type Providers are about replacing our conventional notion of a "library" with a provider model. This allows a type provider to project an external information source into F# and makes it easier to access diverse sources of data."

# Demo


WSDL Mash-up

Using F# to visualize World Bank data in R

# Set 2: Data Visualization

**Charting Quick-Start**

- Add a vertical line to the “Combining Charts” example to make an asterisk, then center the figure on the chart.

**Introduction to Data Visualization**

- Add a title to the Column chart.

**Sampling Functions and Performance**

- Using DateTime.Now.Ticks, find the difference in processing time for the final samples.

# Break

# Agenda

| | |
|---|---|
| **1:00** | Intro and Setup |
| **1:15** | **Set 1**: Getting Started in F# |
| **1:50** | Intro to Type Providers |
| **2:10** | **Set 2**: Data Visualization |
| **2:45** | Break |
| **2:55** | **Set 3**: Data Science |
| **3:30** | Challenge! |
| **4:05** | Wrap up |
| **4:15** | Fin! |

# Set 3: Data Science

# Challenge!

Use tryfsharp.org/Create to find the name of the country with the max gasoline price in 2010.

Bonus: Graph the top 10 countries in that list with a bar or column chart.

# Additional Resources

- MSDN F# home: http://fsharp.net
- F# 3 Sample Pack: F# 3.0 Sample Pack
- F# Koans: https://github.com/ChrisMarinos/FSharpKoans
- Try F#: http://www.tryfsharp.org or http://tryfs.net/
- Skills Matter: http://skillsmatter.com/go/scala
- Community contributed samples: http://fssnip.net/
- Wikibook: http://en.wikibooks.org/wiki/Programming:F_Sharp
- Chat rooms: http://jabbr.net/ or http://irc.freenode.net
- FPish: http://fpish.net/home/1/f~23

# Try F#:
# From Zero to Data Science

Rachel Reese | @rachelreese | rachelree.se | github: rachelreese