What is FsReveal?

- Generates reveal.js presentation from markdown
- Utilizes FSharp.Formatting for markdown parsing
- Get it from http://fsprojects.github.io/FsReveal/

Reveal.js

- A framework for easily creating beautiful presentations using HTML.

> ***Atwood's Law****: any application that can be written in JavaScript, will eventually be written in JavaScript.*

FSharp.Formatting

- F# tools for generating documentation (Markdown processor and F# code formatter).
- It parses markdown and F# script file and generates HTML or PDF.
- Code syntax highlighting support.
- It also evaluates your F# code and produce tooltips.

Syntax Highlighting

F# (with tooltips)

```fsharp
let a = 5
let factorial x = [1..x] |> List.reduce (*)
let c = factorial a
```

C#

```csharp
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Hello, world!");
    }
}
```

JavaScript

```javascript
function copyWithEvaluation(iElem, elem) {
    return function (obj) {
        var newObj = {};
        for (var p in obj) {
            var v = obj[p];
            if (typeof v === "function") {
                v = v(iElem, elem);
            }
            newObj[p] = v;
        }
        if (!newObj.exactTiming) {
            newObj.delay += exports._libraryDelay;
        }
        return newObj;
    };
}
```

Haskell

```haskell
recur_count k = 1 : 1 :
    zipWith recurAdd (recur_count k) (tail (recur_count k))
        where recurAdd x y = k * x + y

main = do
  argv <- getArgs
  inputFile <- openFile (head argv) ReadMode
  line <- hGetLine inputFile
  let [n,k] = map read (words line)
  printf "%d\n" ((recur_count k) !! (n-1))
```

code from NashFP/rosalind

SQL

```sql
select *
from
(select 1 as Id union all select 2 union all select 3) as X
where Id in (@Ids1, @Ids2, @Ids3)
```

sql from Dapper

Paket

```
source https://nuget.org/api/v2

nuget Castle.Windsor-log4net >= 3.2
nuget NUnit

github forki/FsUnit FsUnit.fs
```

C/AL

```
PROCEDURE FizzBuzz(n : Integer) r_Text : Text[1024];
VAR
  l_Text : Text[1024];
BEGIN
  r_Text := '';
  l_Text := FORMAT(n);

  IF (n MOD 3 = 0) OR (STRPOS(l_Text,'3') > 0) THEN
    r_Text := 'Fizz';
  IF (n MOD 5 = 0) OR (STRPOS(l_Text,'5') > 0) THEN
    r_Text := r_Text + 'Buzz';
  IF r_Text = '' THEN
    r_Text := l_Text;
END;
```

Bayes' Rule in LaTeX

$$\Pr(A|B) = \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B|A)\,\Pr(A)+\Pr(B|\neg A)\,\Pr(\neg A)}$$

The Reality of a Developer's Life

When I show my boss that I've fixed a bug:

When your regular expression returns what you expect:

from The Reality of a Developer's Life - in GIFs, Of Course