Parsing F# with PetitParser

F# indentation sensitive lightweight syntax

Milan Kubicek
SCG Seminar
06.01.2015

some F# code..

```
1   #light "off"
2
3   let rec fibonacci n = (if (n = 1 || n = 2) then ( 1 )
4   else (let result = fibonacci(n-1)
5   + fibonacci(n-2) in
6    result
7   )
8   );;
9
10  for i in 1 .. 10 do ( printfn "%d: %d" i (fibonacci i) ) done
```

..some nicer F# code

```
1   #light "off"
2
3   let rec fibonacci n =
4   (
5       if (n = 1 || n = 2) then
6       (
7           1
8       )
9       else
10      (
11          let result = fibonacci(n-1)
12                     + fibonacci(n-2) in
13          result
14      )
15  );;
16
17  for i in 1 .. 10 do
18  (
19      printfn "%d: %d" i (fibonacci i)
20  ) done
```

```
Output:


1:    1
2:    1
3:    2
4:    3
5:    5
6:    8
7:   13
8:   21
9:   34
10:  55
```

indentation as secondary notation

```
#light "off"

let rec fibonacci n =
    if (n = 1 || n = 2) then
        1
    else
        let result = fibonacci(n-1)
                   + fibonacci(n-2)
        result


for i in 1 .. 10 do
    printfn "%d: %d" i (fibonacci i)
done
```

indentation as formal notation

```
#light "on"

let rec fibonacci n =
    if (n = 1 || n = 2) then
        1
    else
        let result = fibonacci(n-1)
                   + fibonacci(n-2)
        result

for i in 1 .. 10 do
    printfn "%d: %d" i (fibonacci i)
```

indentation in programming

“curly bracket programming languages”

secondary notation
/ syntactic sugar

```
else
{
    if (stuff is true)
    {
        Do stuff
    }
    else
    {
        if (other stuff is true)
        {
            Do other stuff
        }
        else
        {
            if (stuff is still not true)
            {
                Do even more other stuff
                }
            }
        }
}
```

Python

Haskell

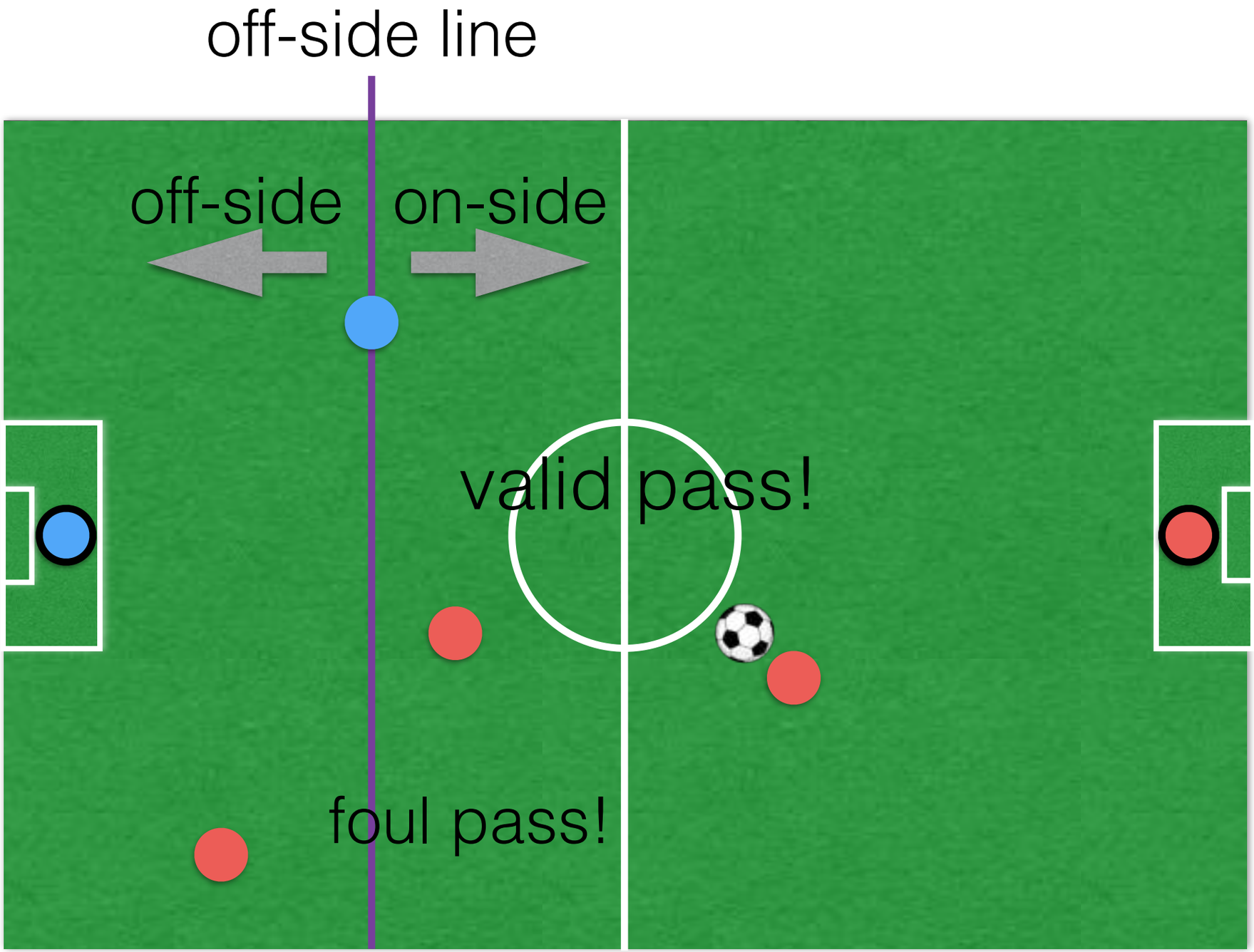
“indentation sensitive
programming languages”

define structure

F#

```
else
    if (stuff is true)
        Do stuff
    else
        if (other stuff is true)
            Do other stuff
        else
            if (stuff is still not true)
                Do even more other stuff
```

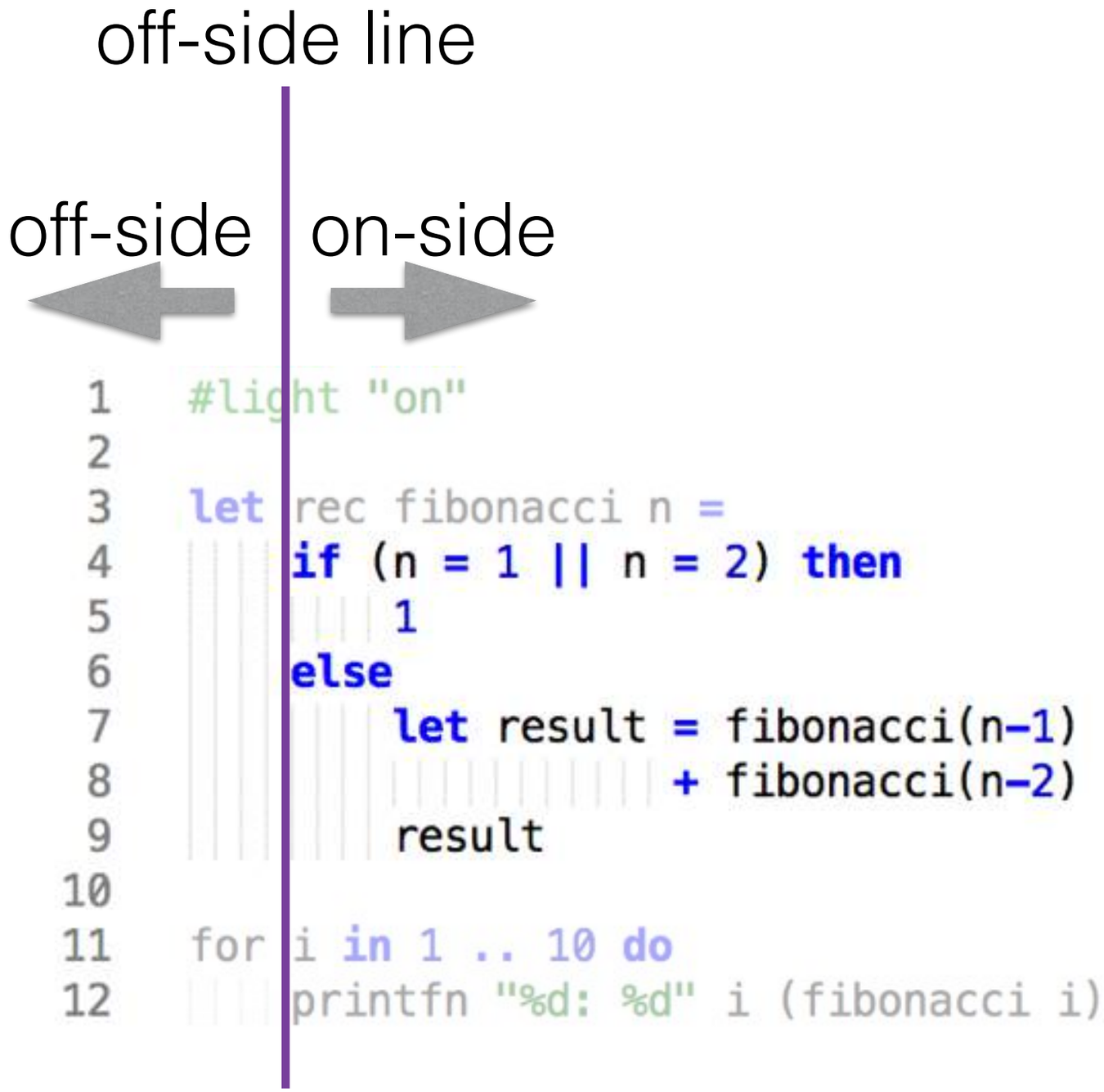
off-side rule



off-side rule

```
1   #light "on"
2
3   let rec fibonacci n =
4       if (n = 1 || n = 2) then
5           1
6       else
7           let result = fibonacci(n-1)
8                      + fibonacci(n-2)
9           result
10
11  for i in 1 .. 10 do
12      printfn "%d: %d" i (fibonacci i)
```

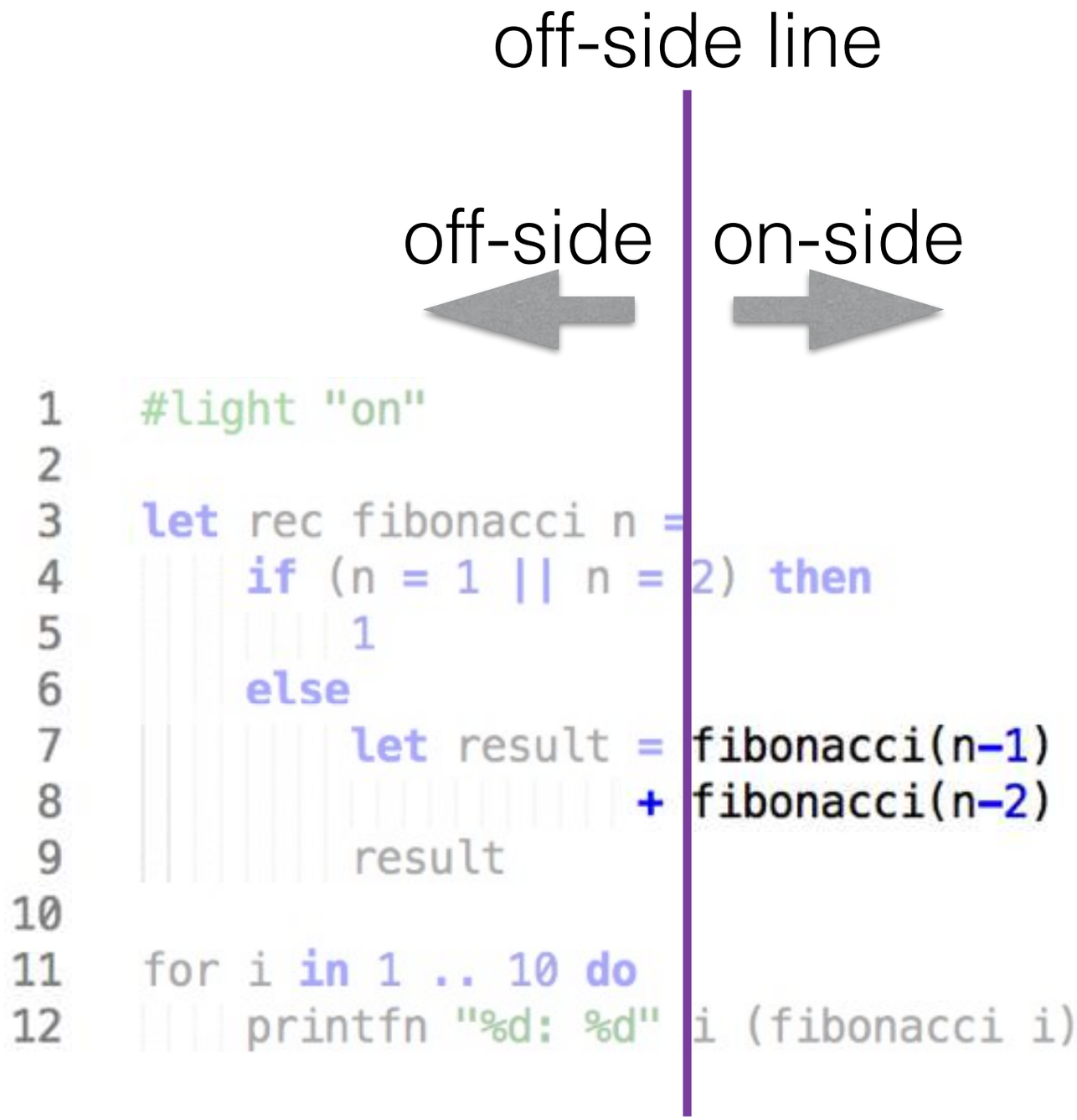
off-side rule



off-side rule

```
1   #light "on"
2
3   let rec fibonacci n =
4       if (n = 1 || n = 2) then
5           1
6       else
7           let result = fibonacci(n-1)
8                      + fibonacci(n-2)
9           result
10
11  for i in 1 .. 10 do
12      printfn "%d: %d" i (fibonacci i)
```

off-side rule exceptions




```
1   #light "on"
2
3   let rec fibonacci n =
4       if (n = 1 || n = 2) then
5           1
6       else
7           let result = fibonacci(n-1)
8                      + fibonacci(n-2)
9           result
10
11  for i in 1 .. 10 do
12      printfn "%d: %d" i (fibonacci i)
```

off-side rule exceptions

“then-&else-body on-side aligned”

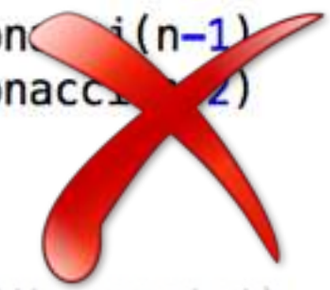
```
#light "on"

let rec fibonacci n =
    if (n = 1 || n = 2) then
        1
    else
        let result = fibonacci(n-1)
                   + fibonacci(n-2)
        result

for i in 1 .. 10 do
    printfn "%d: %d" i (fibonacci i)
```

“else-body aligned”

```
#light "on"

let rec fibonacci n =
    if (n = 1 || n = 2) then
        1
    else
    let result = fibonacci(n-1)
               + fibonacci(n-2)
    result

for i in 1 .. 10 do
    printfn "%d: %d" i (fibonacci i)
```

“then-body aligned”

```
#light "on"

let rec fibonacci n =
    if (n = 1 || n = 2) then
    1
    else
        let result = fibonacci(n-1)
                   + fibonacci(n-2)
        result

for i in 1 .. 10 do
    printfn "%d: %d" i (fibonacci i)
```



challanges

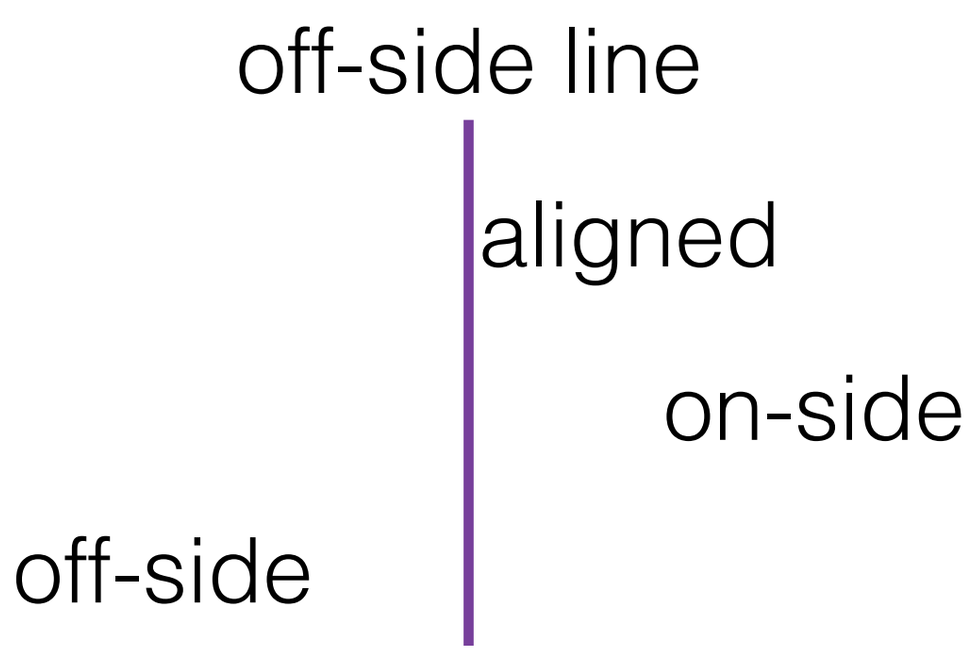
(Petit)Parsing knowledge

mini F# PetitParser grammar
(using PetitIndent)

F# language definition

F# lightweight syntax

indentation as primary notation:
..code position changes semantics!




WHITESPACE
IT'S NOT JUST FOR
FORMATTING ANYMORE
memegenerator.net

questions?

secondary notation

"…An example of secondary notation is in computer programming, where code is often displayed with positioning, indentation, color and symmetry. This does not affect the behaviour of the program, but it makes it easier to read and understand the code…"

– http://en.wikipedia.org/wiki/Secondary_notation 05.01.2015

Off-side rule

"Any non-whitespace token to the left of the first such token on the previous line is taken to be the start of a new declaration."

–Peter J. Landin, "The Next 700 Programming Languages"