# DoAndIfThenElse

山本和彦
(株)インターネットイニシアティブ
kazu@iij.ad.jp

# 自己紹介

- 山本和彦
  - この会の主催者です
- 趣味
  - ジョギング、ロッククライミング、サッカー、ダイビング
  - 吟醸酒、ワイン
  - 科学全般
  - 今は子育てで何もできていません！

# 問題

- do の中で以下のように書きたいけどエラーになる

```
main = do
    if True
    then putStrLn "Boo"
    else putStrLn "Foo"

main = do
    if True then
       putStrLn "Boo"
    else
       putStrLn "Foo"
```

# 素朴な疑問

- 以下は正しい

```
a = if True
    then "Boo"
    else "Foo"

a = if True then
      "Boo"
    else
      "Foo"
```

- 疑問）do の中では NG なのに
  どうして普通の式では OK なの？

# 驚愕の事実

- 驚くべきことに以下も正しい

```
a = if True
      then "Boo"
        else "Foo"

a = if True then
       "Boo"
      else
    "Foo"
```

- 疑問）何故だー？？？

# if の秘密

- Haskell 98: if は 1 つの式である

```
exp -> if exp1 then exp2 else exp3
```

- if には endif がない
  - if 式の終わりは、次の式が始まるまで
  - 次の式は、if と同じ行頭揃えから始まる
  - なので、if を複数行に書く場合は
    2行目以降の行頭を下げておけばいい
  - 下がっている部分は、レイアウト規則により、
    一行と見なされる

# 先ほどの例を再び

- オフサイドラインに注意

```
|a = if True
|       then "Boo"
|         else "Foo" -- 式1
|b = ...             -- 式2

|a = if True then
|         "Boo"
|       else
|     "Foo"          -- 式1
|b = ...             -- 式2
```

# do の例も再び見る

- 3つの式と解釈される

```
main = do
     |if True                -- 式1
     |then putStrLn "Boo" -- 式2
     |else putStrLn "Foo" -- 式3
```

- 2つの式と解釈される

```
main = do
     |if True then
     |   putStrLn "Boo"   -- 式1
     |else
     |   putStrLn "Foo"   -- 式2
```

# DoAndIfThenElse での提案

- Haskell 2010: if を 3 つの式にする

```
exp -> if exp1 [;] then exp2 [;] else exp3
```

- すると以下は、3式なのでOK

```
main = do
    |if True                -- 式1
    |then putStrLn "Boo" -- 式2
    |else putStrLn "Foo" -- 式3
```

- 以下は、2式 + 1式 = 3 式で OK

```
main = do
    |if True then
    |   putStrLn "Boo"   -- 式1 + 式2
    |else
    |   putStrLn "Foo"   -- 式3
```

# 問題点

- 明示的に；を書いてしまう人が出てくるかも
- そうなると、規則を再び変えたときに、
  互換性がなくなる

# 実装状況

- GHC
  - 未対応
  - 提案では、GHC も対応したと書いてあったけれど。。。
- Hugs
  - 対応済み
  - 明示的に ; を書くとエラーになる。。。