

Swyft UI の実験

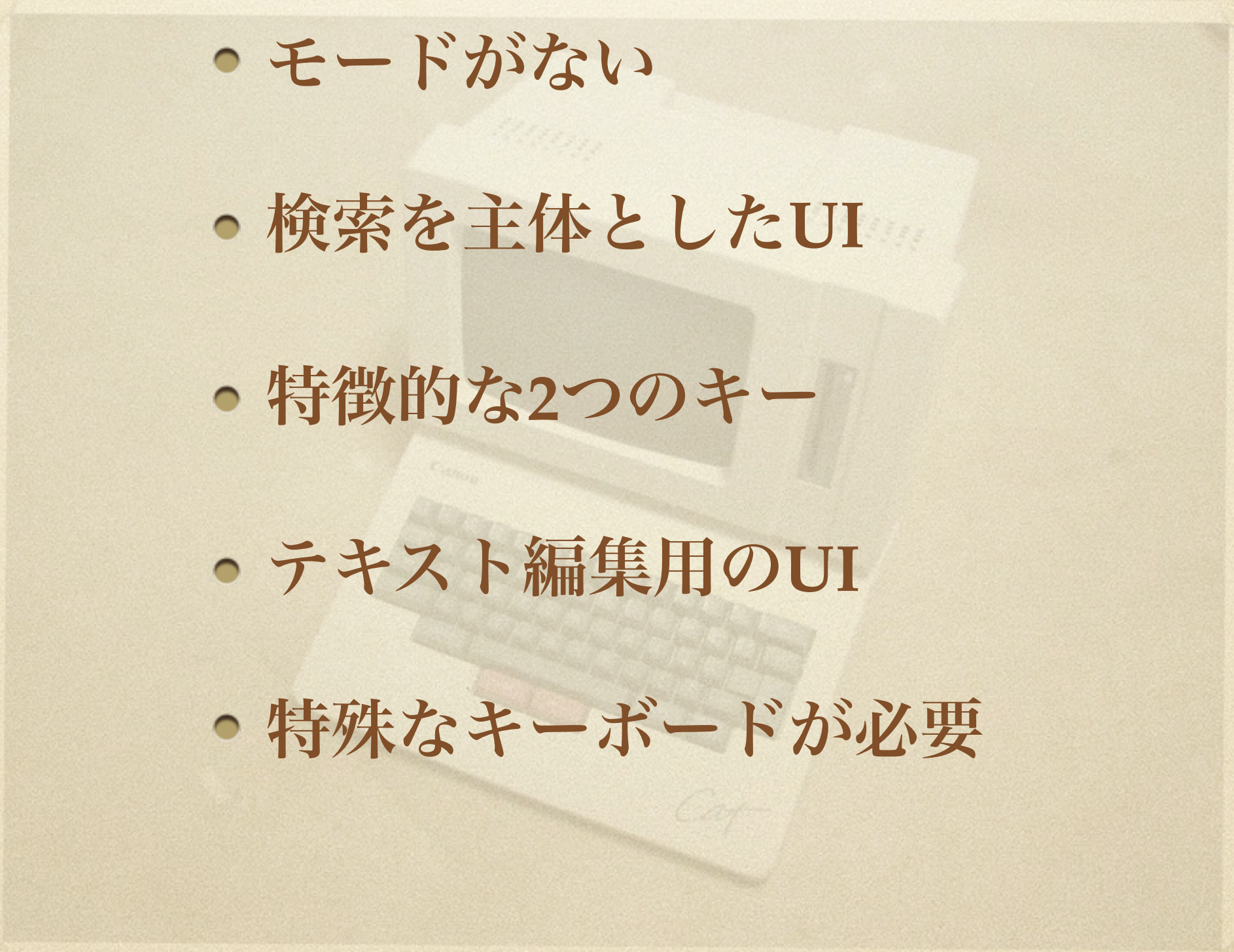
2016/01/08

成田丞(mindtools@mac.com)

Swyft UI とは何か



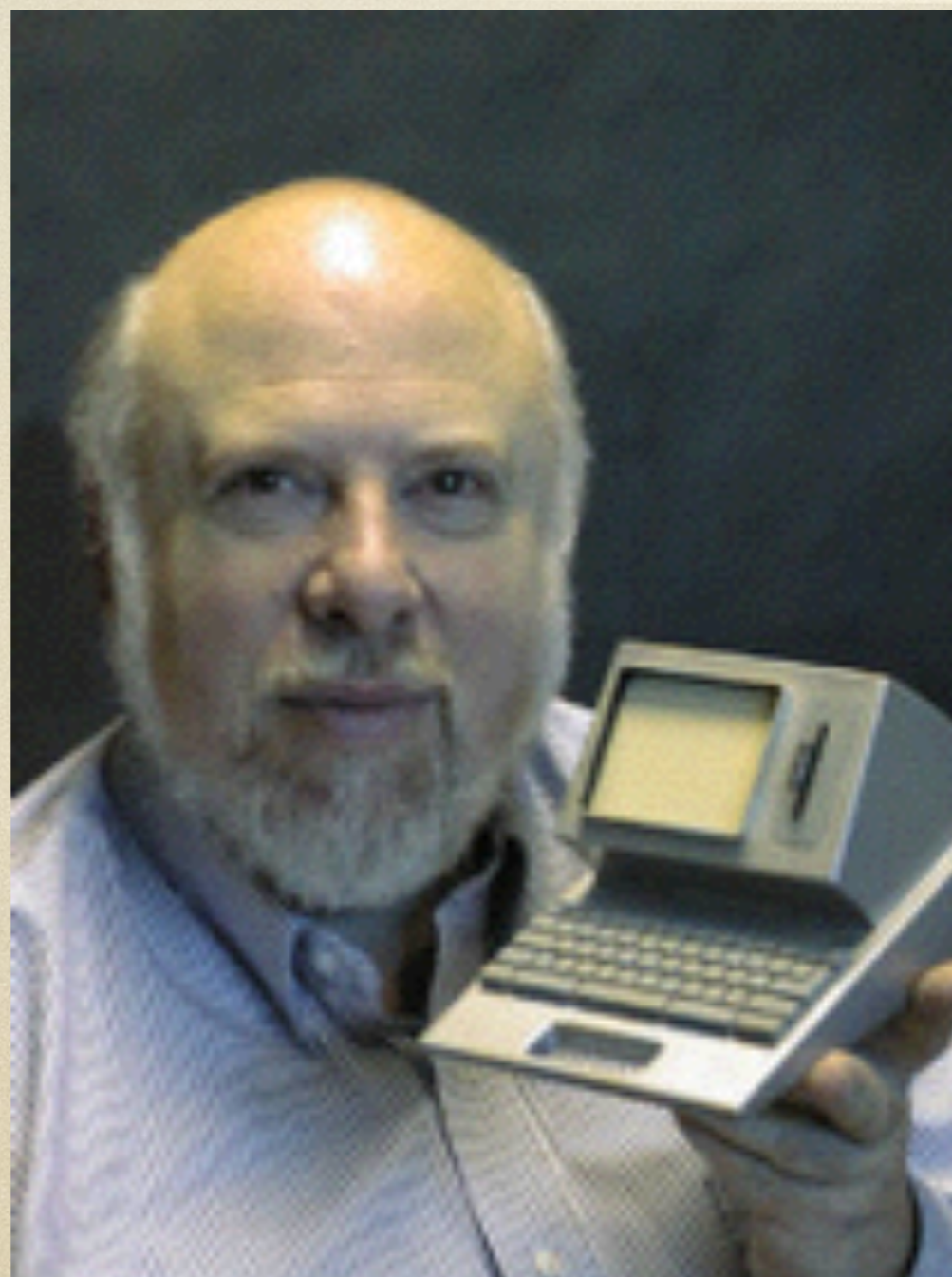
Swyft UIとは何か

- モードがない
 - 検索を主体としたUI
 - 特徴的な2つのキー
 - テキスト編集用のUI
 - 特殊なキーボードが必要
- 

Swyft UI とは何か

- 伝説の男 Jef Raskin が設計した
テキスト編集のUI
- ワープロ機能を中心に検索機能
を応用した住所録機能や計算機能が
入っていた。

Swyft UI とは何か



Swift UI とは何か

- 計算式をテキストで入れて、評価すると、答えが出てくる。

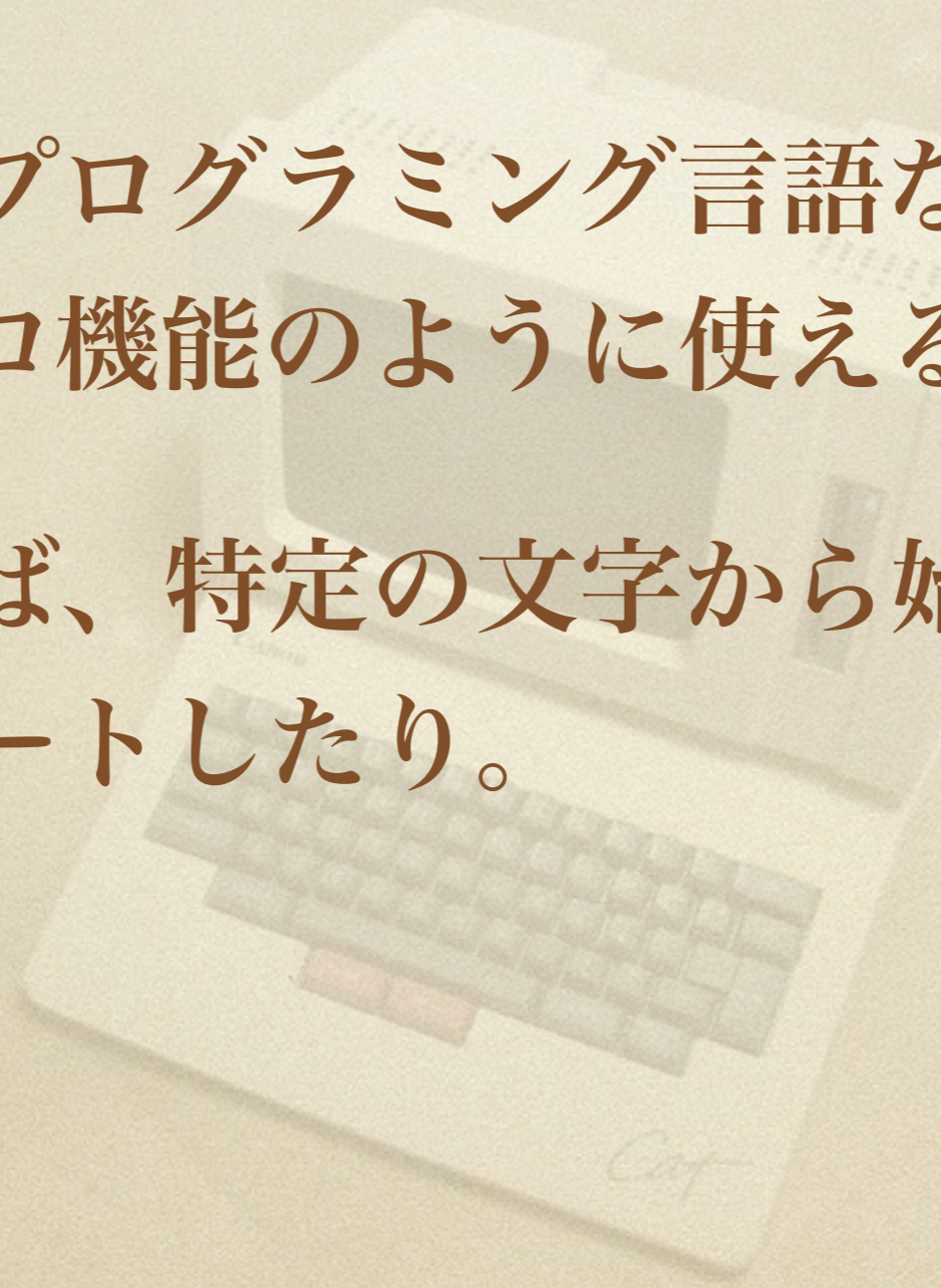
1+1

1+1

1+1 2

Swyft UIとは何か

- 実はプログラミング言語なので、マクロ機能のように使える。
- 例えば、特定の文字から始まる行をソートしたり。



Swyft UIとは何か

- YYYY/MM/DDの文字列を書いておけば、検索機能を使ってスケジュール帳になる。

2016/01/09

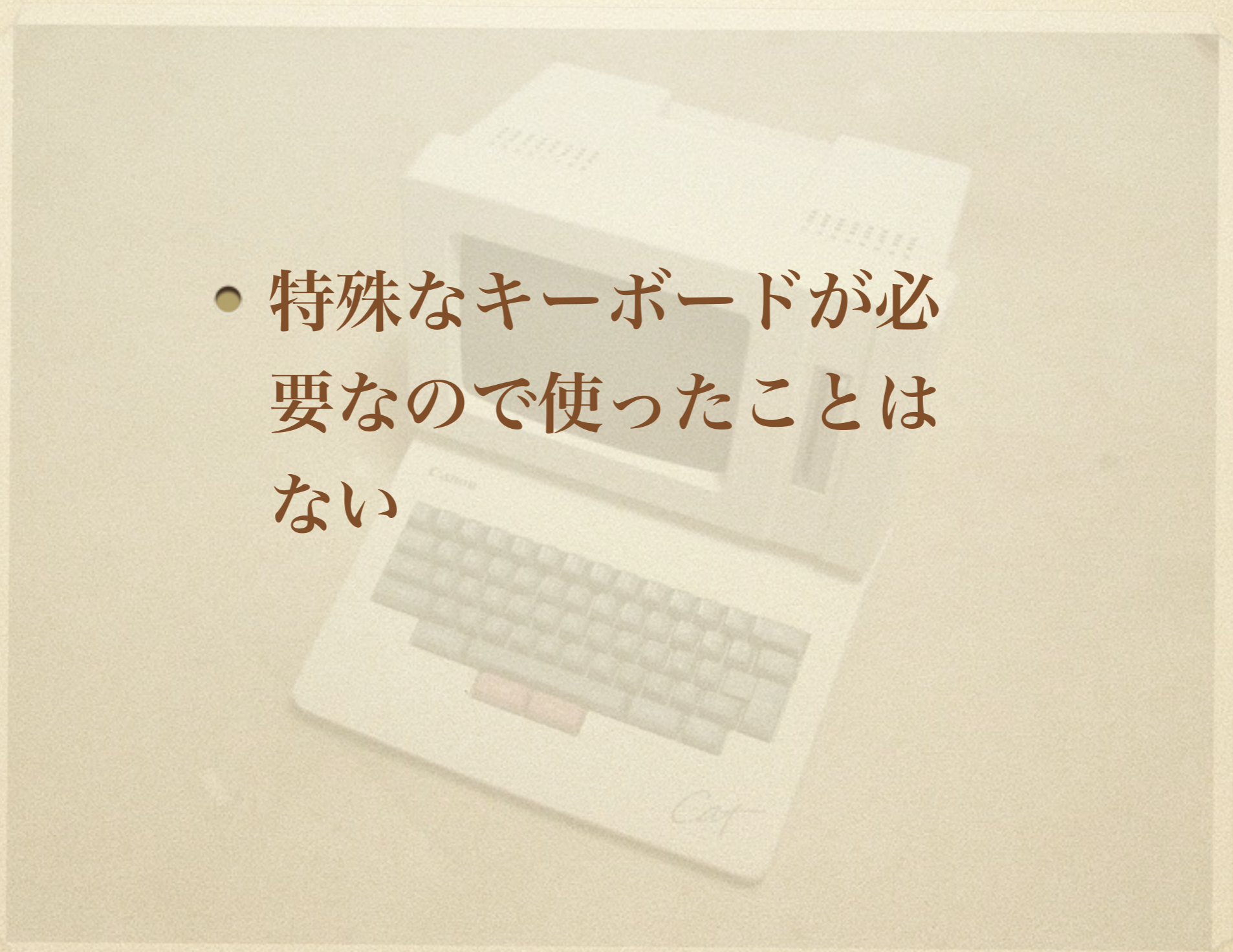
Cocoa勉強会

2016/01/11

成人式

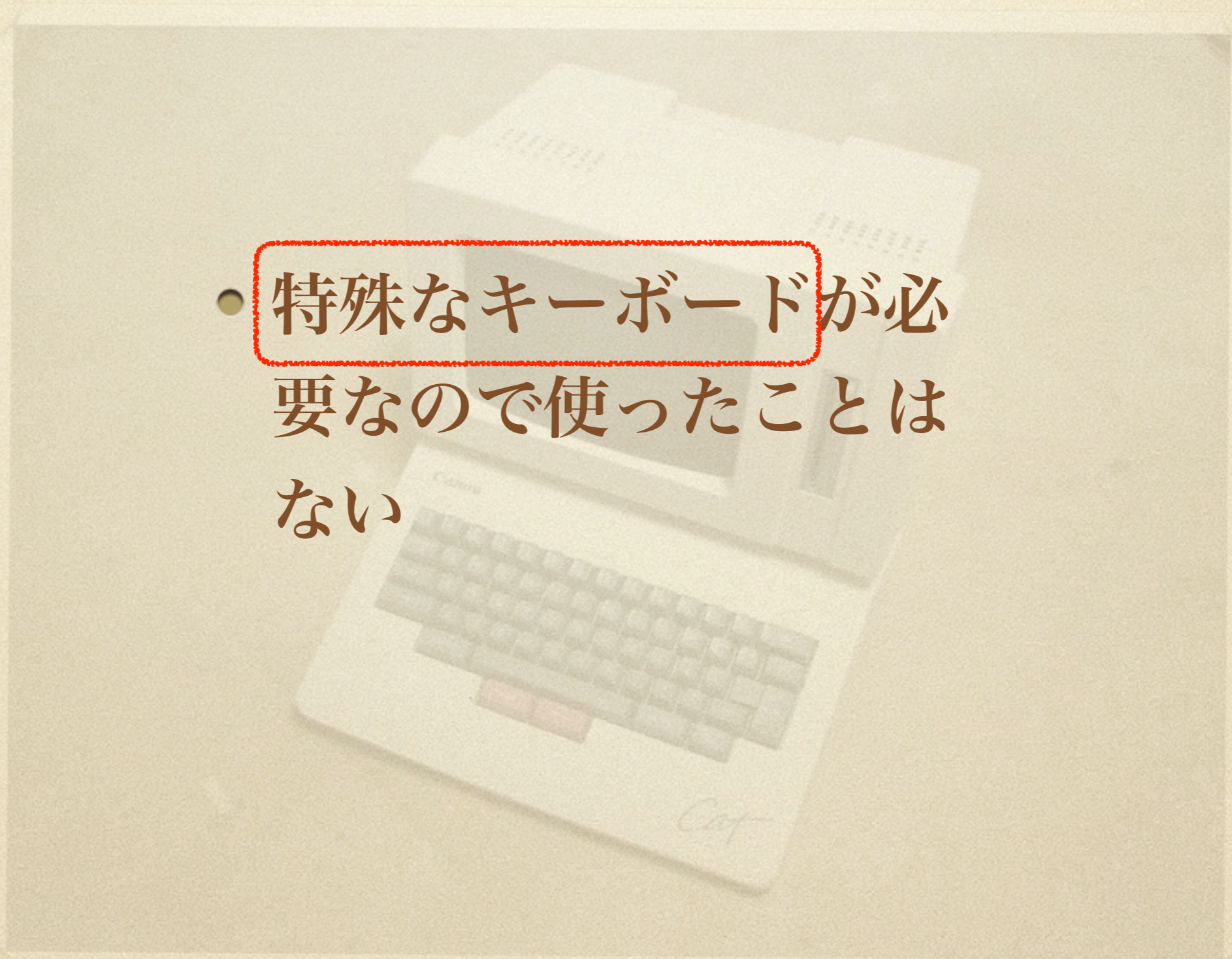
Swyft UIとは何か

- 特殊なキーボードが必要なので使ったことはない



Swyft UI とは何か

- 特殊なキーボードが必要なので使ったことはない



特殊なキーボードが必要



作った

特徴的な2つのキー



ソフトウェアも作った

```
TestTextEdit  File  Edit  Format  View  Window  Help
Untitled
TE
#pragma mark action & canAction methods

- (void) updateTextEdit:(id)sender
{
    NSRegularExpression* theRep;

    // ローマ字入力から正規表現へ変換
    theRep = [self romaziToRegularExpression:self.textField.stringValue];

    // TextViewのコンテンツで上記の正規表現オブジェクトの一致する範囲を
    // 全てリストアップする
    NSArray* theSelectedRanges;

    theSelectedRanges = [[theRep matchesInString:self.textView.string
                        options:0
                        range:NSMakeRange(0,
                                    self.textView.string.length)]
                        valueForKey:@"range"];
}
```


どーやって使うか？



作った機能は4つ

Leap	Leapキーを押しながら文字を打ち込む
Leap Again	Leapキーを押しながらTabキーを押して離す
Creap	Leapキーを単体で押して離す
Mark	Emacsと同じだが、Leap開始した地点がMark地点。

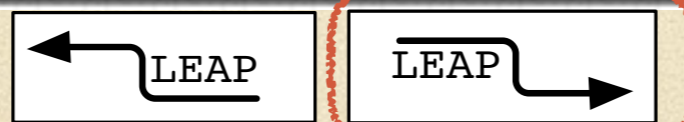
SwyftWareのサブセット

Leapキー単体の動作

動作名：Creap

Leapキー単体の動作

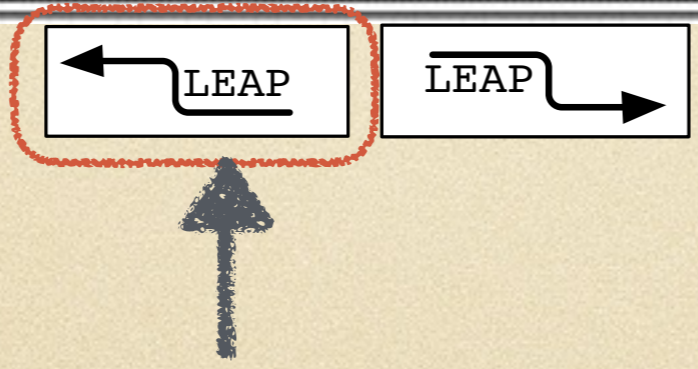
テキスト 1234 **A**BCDEFG 本日は晴天なり



一回押して離すとカーソルが右に移動

SwyftWareのサブセット

テキスト 1234 ABCDEFG 本日は晴天なり



一回押して離すとカーソルが左に移動

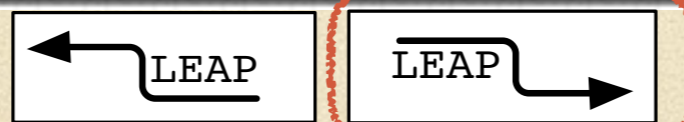
SwyftWareのサブセット

Leapキーを押しながら逐次検索

動作名：Leap

Leapキーを押しながら逐次検索

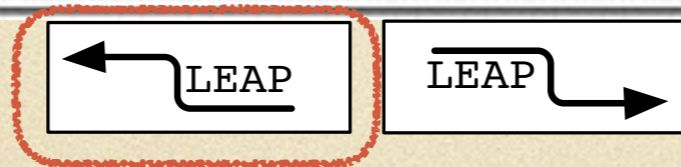
テキスト 1234 ABCDEFG 本日は晴天なり



押し続けて、“ABCDE”と文字を打ち込む

Leapキーを押しながら逐次検索

テキスト **1234** ABCDEFG 本日は晴天なり



押し続けて、“1234”と文字を打ち込む

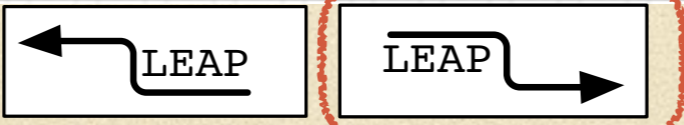
SwyftWareのサブセット

Leap + Tabキーで次の検索結果へ移動

動作名：Leap Again

Leap + Tabキーで次の検索結果へ移動

.....▶
abc ab abc ab abcdef ab ab **abc** abcdef

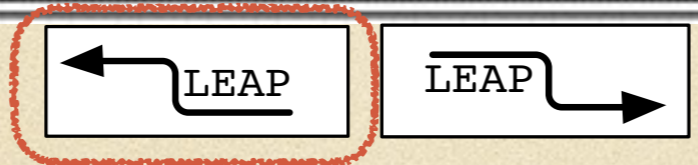


一度"abc"をLeap後にLeapキーを押し続けて、Tabキーを連打

Leap + Tabキーで前の検索結果へ移動



abc ab abc ab abcdef ab ab abc abcdef



一度"abc"をLeap後にLeapキーを押し続けて、Tabキーを連打

SwyftWareのサブセット

Leapキーの同時押しで選択

動作名：選択

Leapキーを押しながら逐次検索

⌘テキスト 1234 ABCDEFG 本日は晴天なり

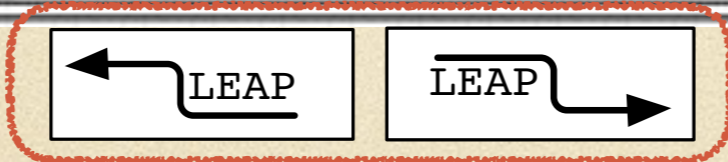


Leap前にカーソルがある位置がMark位置

Emacsのマークに似ているが、明示的に指定しない

Leapキーを押しながら逐次検索

テキスト 1234 ABCDEFG 本日は晴天なり



“ABCDE”のLeap後に同時押し

マークからカーソル位置までが選択される

デモを行う

- LeapキーでCreap
- LeapキーでLeap
- TabキーでLeap Again
- Leapキーで選択

Q：逐次検索は日本語では使いにく
いのではないか？

A:C/Migemoを使う

<https://github.com/koron/cmigemo>

<https://github.com/laiso/OCMigemo>

Q：押しているだけ有効なのはUIと
しておかしくないか？

A:SwyftWareは擬似モード

- 特定キーを機能に結びつける。
- キーを押してる間だけのモード
- キーを離したら元に戻る
- 意外に身近なUI



擬似モード

スポイト

定規



複製ツール

ハンドツール

忘れた

作って使った感想

主な機能が親指で使えるのは便利

Emacsのctrlとmetaが両方とも親指で押せる感じ

明示的なMark機能がないのは不便

既存のOSの機能と整合性を考えるのが面倒。

これはEmacsも同じ。

作って使った感想

消費するためのUIと生産するためのUIの2種類がある。

世の中の主流は、消費させるためのUIだが、これは生産するためのUI。