

# NewtonScriptに見る もうひとつのプロトタイプ指向

Yet Another Prototype Oriented

LL Future Lightning Talk 発表資料

GNUE(鵜)

30 Aug 2008



# NewtonScript

- Apple の PDA “Newton”のために生まれた
- 文法は Pascal + JSON
- 動的オブジェクト指向言語？
- プロト、ペアレントのデュアル継承
- クロージャ、レキシカルスコープ





オブジェクト指向というと

Class ?

いいえ  
プロトタイプベースもあります！



# プロトタイプ指向

- クラスは不要
- すべてがインスタンス
- プロトタイプチェーンで継承



# 代表的な言語

- Self
- NewtonScript
- JavaScript
- Io



# オブジェクトの作成(1)

```
NewtonScript := {  
  Paradigm: "Prototype-based",  
  AppearedIn: 1993,  
  DesignedBy: "Walter Smith",  
  TypingDiscipline: 'dynamic',  
  Influencedby: ["Self", "Dylan"],  
  Influenced: ["Io"]  
};
```



# オブジェクトの作成(2)

```
language := {  
  TypingDiscipline: 'dynamic',  
  age: func() 2008 - AppearedIn  
};
```

```
NewtonScript := {  
  _proto: language,  
  Paradigm: "Prototype-based",  
  AppearedIn: 1993,  
  DesignedBy: "Walter Smith",  
  Influencedby: ["Self", "Dylan"],  
  Influenced: ["Io"]  
};
```



# 参照

- スロットへのアクセス

```
p(NewtonScript.TypingDiscipline);  
=> 'dynamic'
```

- メソッドの呼出し

```
p(NewtonScript:age());  
=> 15
```



# 継承(1)

- スロットへのアクセス

```
p(NewtonScript.TypingDiscipline);  
=> 'dynamic
```

- スロットの値を上書きする

```
NewtonScript.TypingDiscipline := 'static;  
p(NewtonScript.TypingDiscipline);  
=> 'static
```

- プロトは書換えられていない

```
p(NewtonScript._proto.TypingDiscipline);  
=> 'dynamic
```



# 継承(2)

- 継承されたメソッドを呼出す

`inherited:メソッド名(引数)`



# プロトタイプチェーン

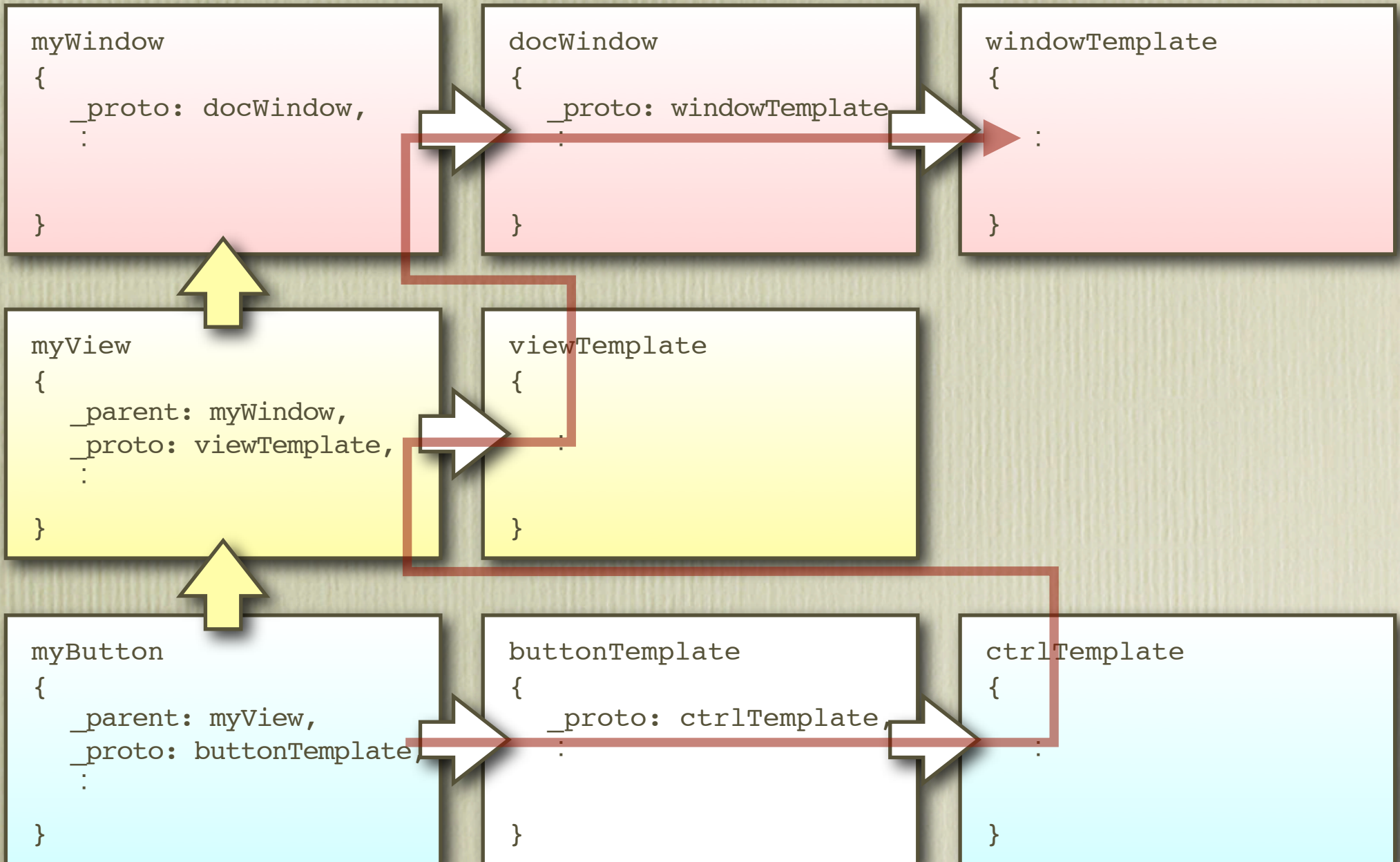
```
myPage
{
  _proto: pageTemplate,
  :
}
```

```
pageTemplate
{
  _proto: someFrame,
  :
}
```

```
someFrame
{
  :
}
```



# プロト・ペアレント継承





# 結論

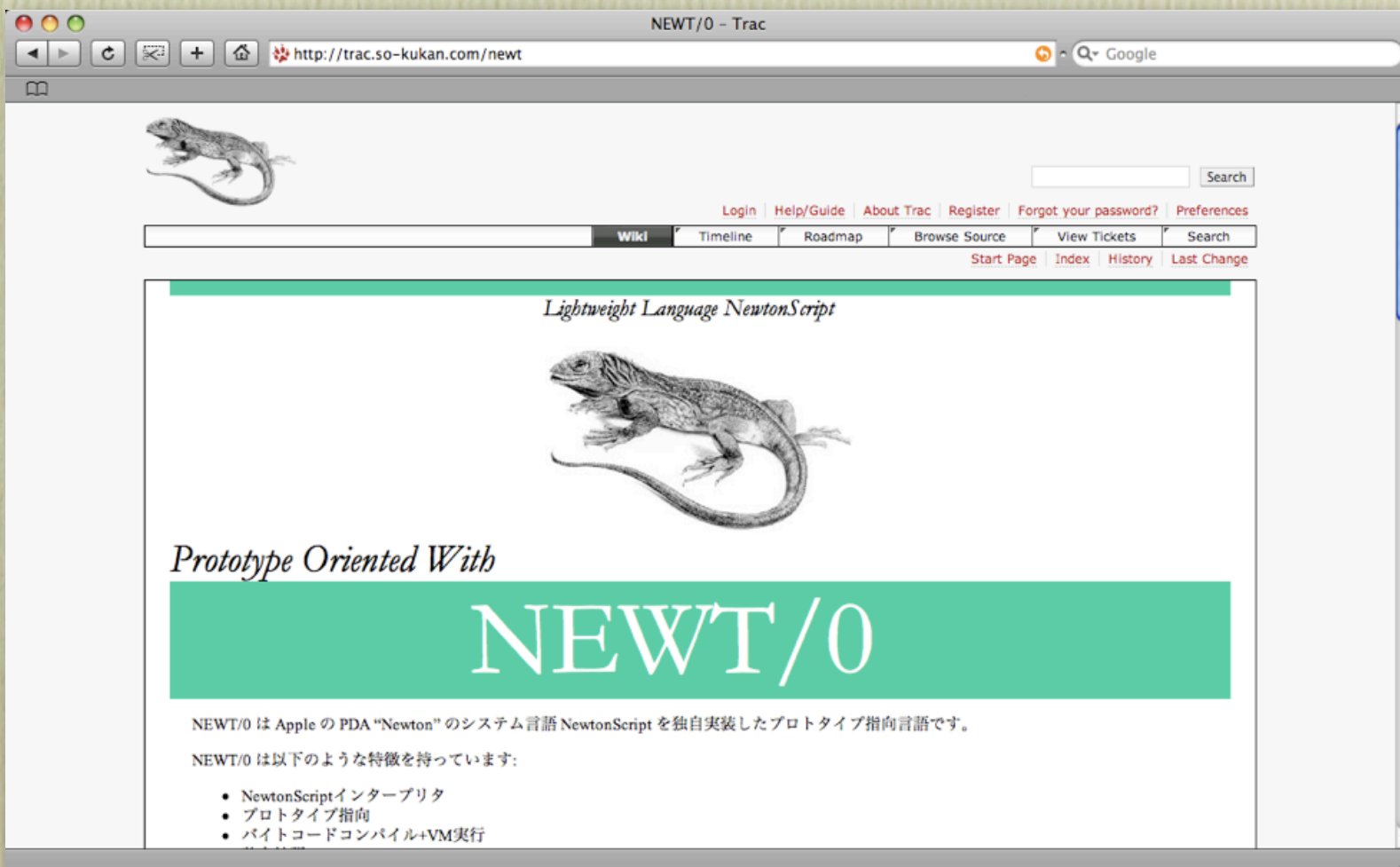
プロトタイプ指向は怖くない！







# NewtonScriptの処理系



<http://trac.so-kukan.com/newt>