Feature-Oriented Programming

特色導向程式開發

Fred Chien

錢逢祥

cfsghost

@ gmail.com

cfsghost

At Github

HanGee

蕃薯國民機

寫了本 Node.js 的書

不一樣的

LXDE, NPK… etc

Open Source Developer

Stem OS

JavaScript OS

創業 Startup

AvengerGear, Mandice, PearNature...etc

不過

Well

這些通通不是最重要的事

Nothing Important

立志

立志

以 JavaScript 統一天下

立志

以 JavaScript 統一天下

這才重要！

然後

And then

從此 JSDC
成為第一大語言 Conf

Biggest Conference

從佔中開始

佔領中研院

還有

And

種下希望的種子

話說

Well

最近東搞西搞

OS, Application, Web….

發現件驚人的事實

Found the truth!?

自己重覆做過很多事

Repeat again and again

會員系統 x N

會員系統 x N

還記得那神秘的 OAuth 嗎？

管理系統 x N

管理系統 x N

客戶好喜歡超複雜的權限控管呢！

各種 API x N

各種 API x N

若是要再考慮開放第三方 API …. 呃

如果不是純網站的專案

架構複雜、軟硬體大整合相關

Hardware, System integration….

同個 Library
還要**重寫**好幾次

在同個專案中

同個 Library
還要**複製**好幾次

在同個專案中

客戶追著要大便

Requirement Sucks

但我總是便秘

I suck

更重要的是

The Most Important Thing

N x N = 死阿宅

某天不小心讓 Google 小姐唸了一下 ZxN

連 Google 小姐都嘲笑我

工程師的悲哀

想少一點宅味？

xN 勢必得減少一些

能否像堆積木一樣？

開發工作

Feature-Oriented Programming

特色導向程式開發

FOP

特色導向程式開發

FOSD

特色導向軟體開發

簡單來說

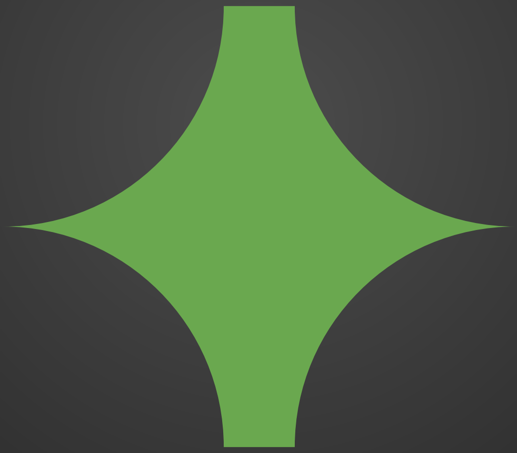
Well

Do you want to build a snowman?

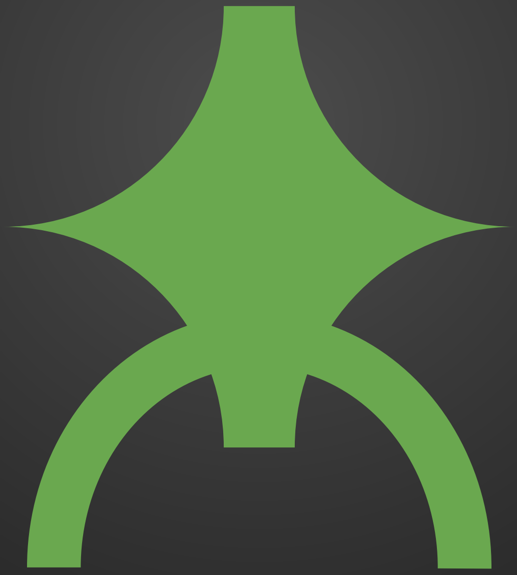
你想做個雪人嗎？

Do you want to build a snowman?
littlemermaidluna
BBC AMERICA

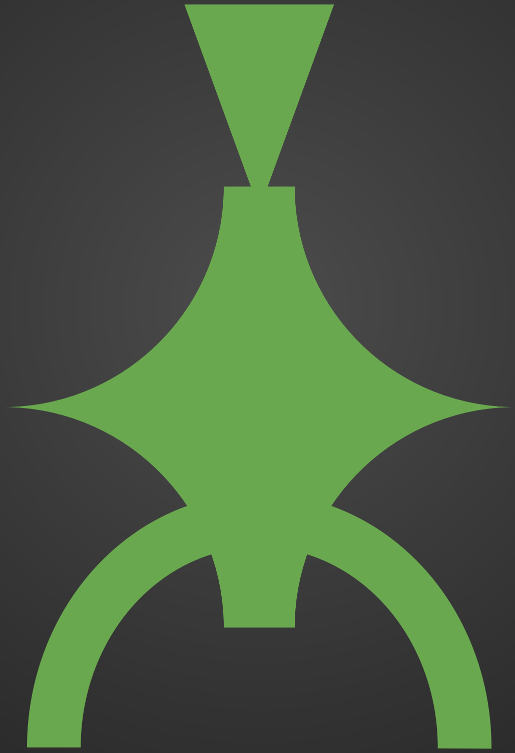
xN 的做法



xN 的做法



xN 的做法



xN 的做法

xN 的做法

一般人的做法

一般人的做法

一般人的做法

一般人的做法

一般人的做法

以 Feature 為本

FOP 的精神

FOP

f -- base program with feature f

FOP

f   -- base program with feature f

h   -- base program with feature h

FOP

f -- base program with feature f

h -- base program with feature h

f • x -- adds feature f to program x

h • x -- adds feature h to program x

FOP

f -- base program with feature f

h -- base program with feature h

f • x -- adds feature f to program x

h • x -- adds feature h to program x

f • h • x -- adds feature f and h to x

層層疊加而成的生產線

Software Product Line

標準的血汗工廠

還好電腦不會跳樓

Sven Apel · Don Batory
Christian Kästner · Gunter Saake

Feature-Oriented Software Product Lines

Concepts and Implementation

Springer

Languages Support

ex, FeatureC++, Java...

JavaScript + FOP ?

有可能嗎？

wag.js

cfsghost At Github

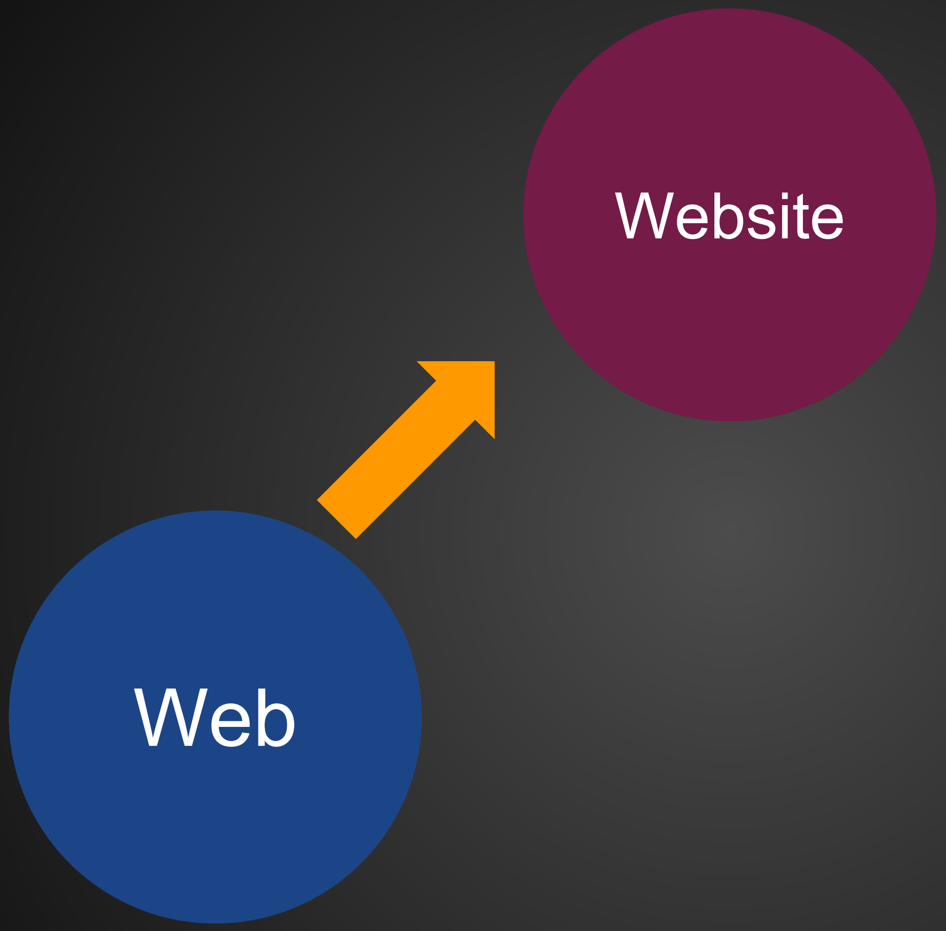
wag.js

搖擺.js

這是我們的希望

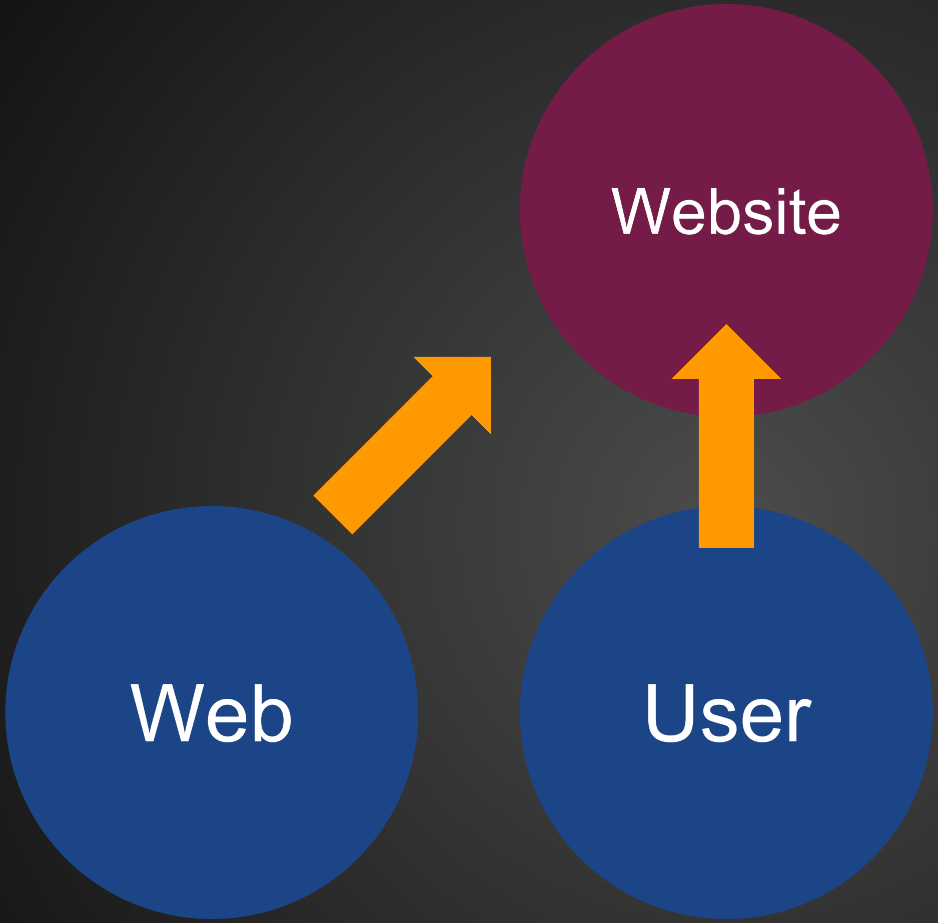
如果可以這樣就做完網站

會員系統・管理系統・前端 = 完成

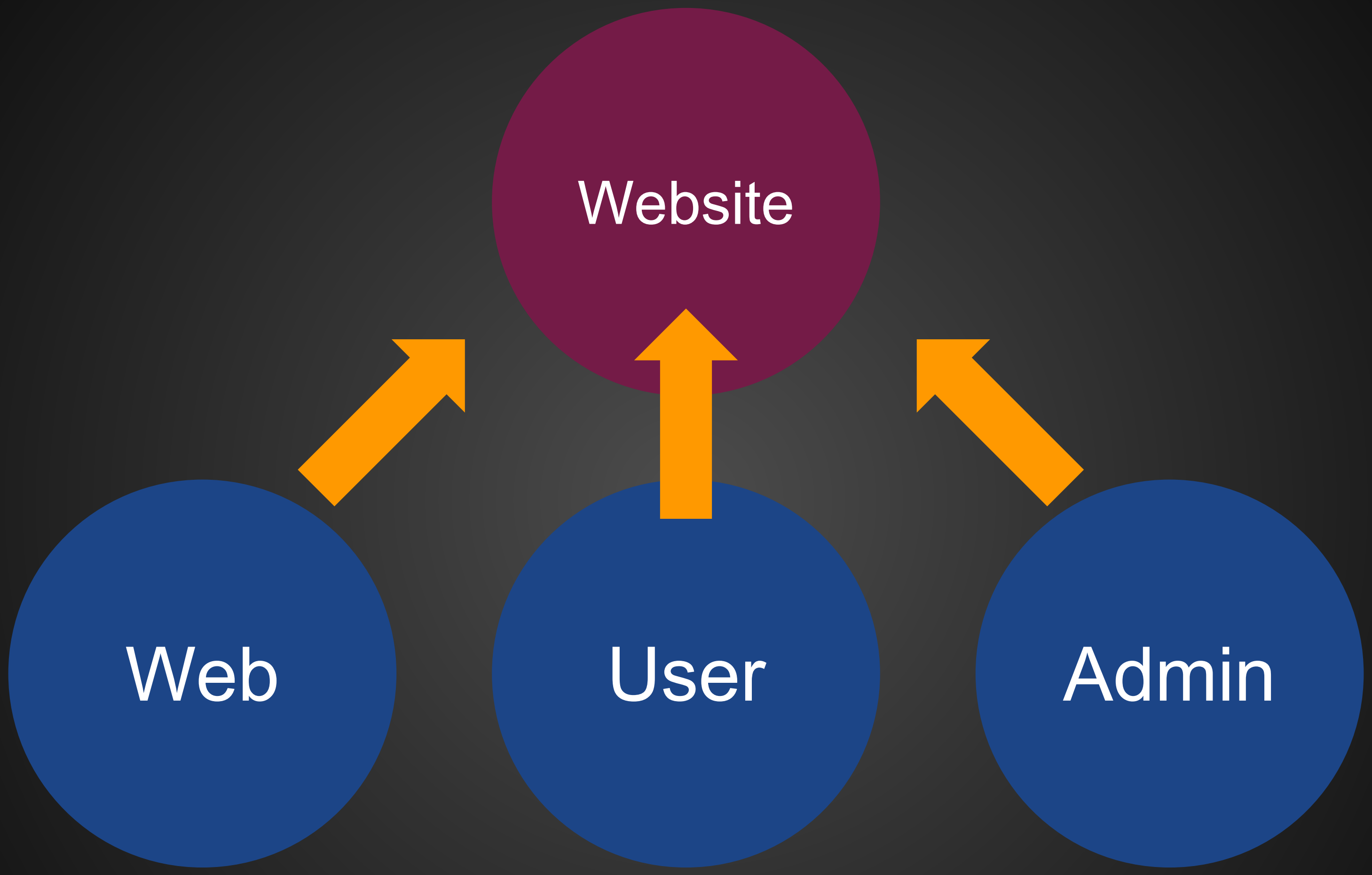


Website

Web



Website
Web
User



Website

Web

User

Admin

FOP in JavaScript

```
var wag = new Wag();

var app = wag.mixin([
    ‘Web’,
    'User',
    'UserAdmin'
]);

app.run(function(feature) {
    feature.listen(8000);
});
```

FOP in JavaScript

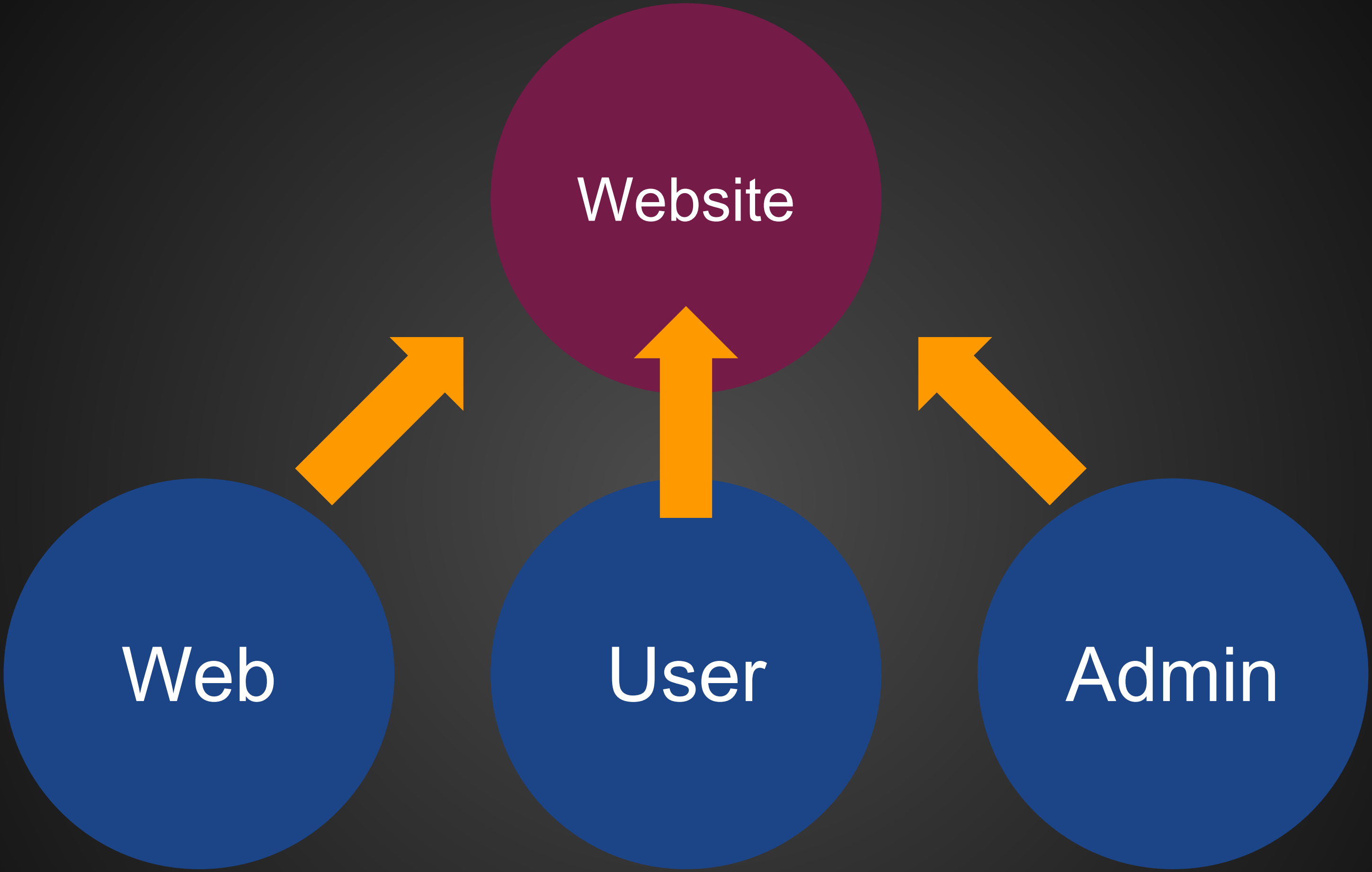
```
var wag = new Wag();

var app = wag.mixin([
    ‘Web’,
    'User',
    'UserAdmin'
]);

app.run(function(feature) {
    feature.listen(8000);
});
```

完成

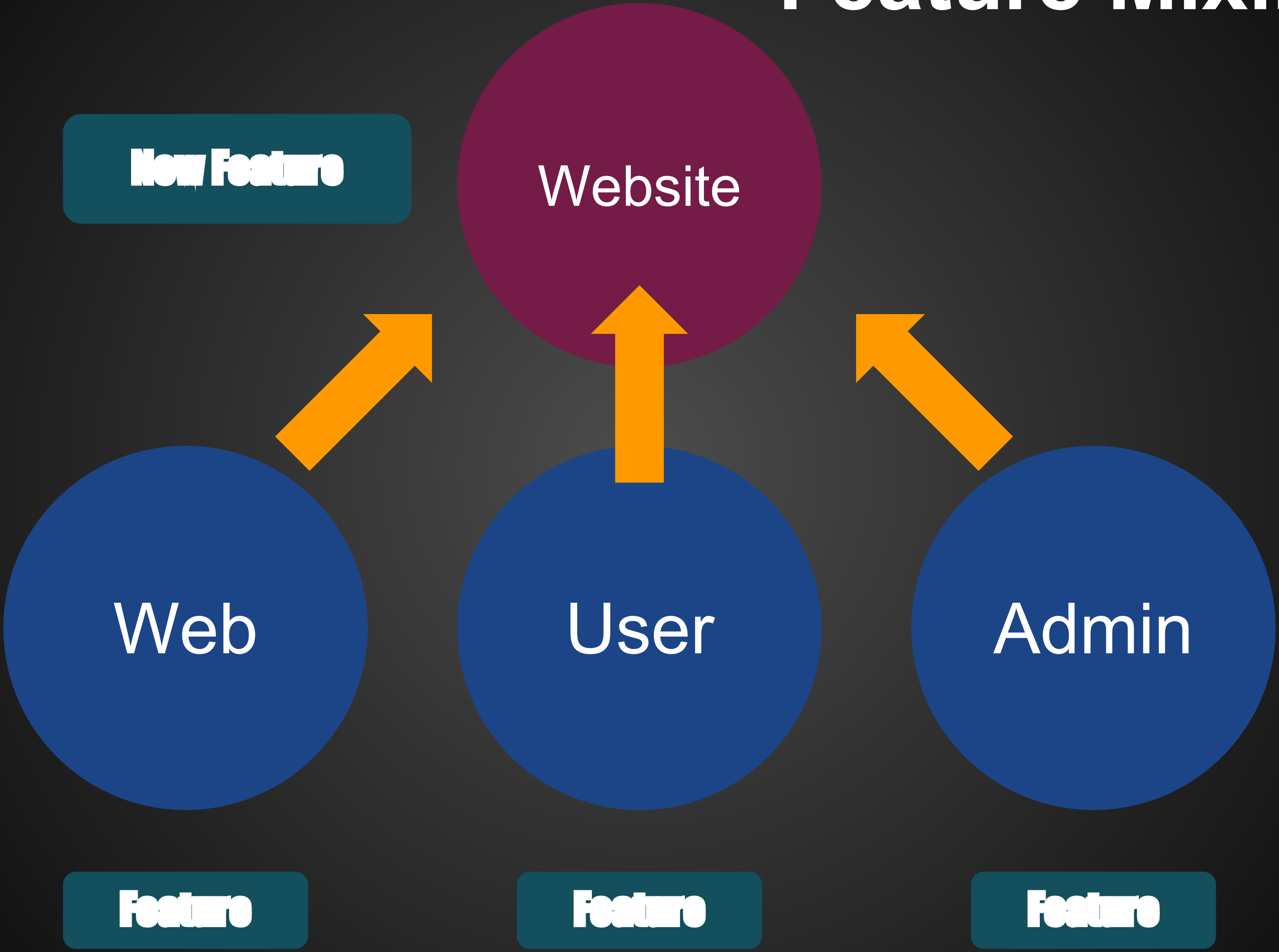
走路都**搖擺**！

Awesome!

How it works?


Website
Web
User
Admin

Feature Mixing



New Feature

Website

Web

User

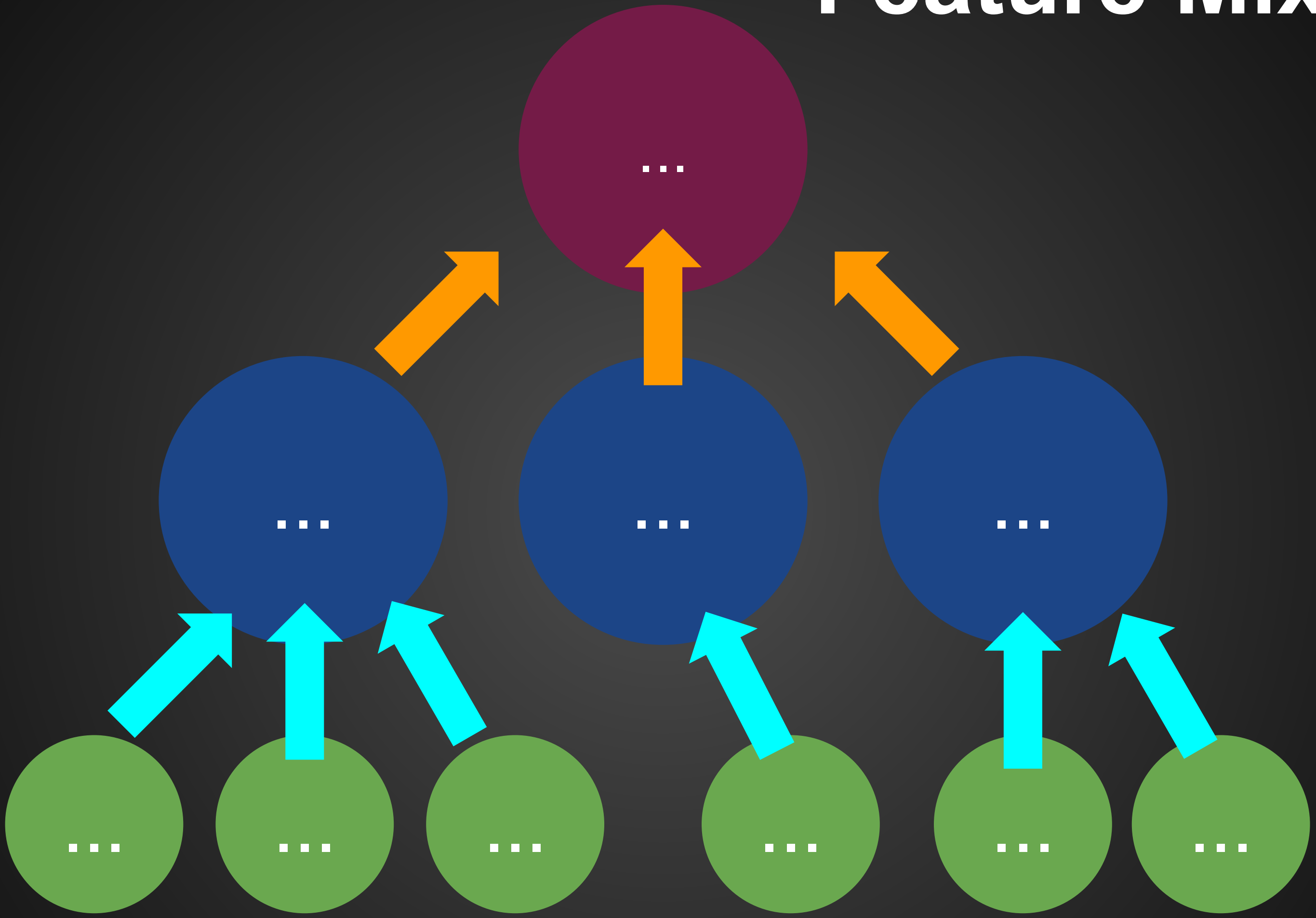
Admin

Feature

Feature

Feature

Feature Mixing



First wag.js Program

開始動手

Loading local Features

```
var wag = new Wag();
var app = wag.mixin([
    ‘./mom.js’,
    './dad.js'
]);

app.run(function(feature) {
    feature.helloMom();
    feature.helloDad();
});
```

mom.js

```
var Mom = module.exports = function(scope) {
        scope.$super(Mom);
};

Mom.prototype.helloMom = function() {
        console.log('hello mom!');
};
```

dad.js

```
var Dad = module.exports = function(scope) {
    scope.$super(Dad);
};

Dad.prototype.helloDad = function() {
    console.log('hello dad!');
};
```

Result

```
$ node app.js
hello mom!
hello dad!
```

Prototype 合成

Mixin Prototypes Up

Mom + Dad

兩個原型的合成

Loading local Features

```
var wag = new Wag();
var app = wag.mixin([
    ‘./mom.js’,
    './dad.js'
]);

app.run(function(feature) {
    feature.helloMom();
    feature.helloDad();
});
```

注意順序

Mom + Dad

實際上是繼承關係

Dad 繼承了 Mom

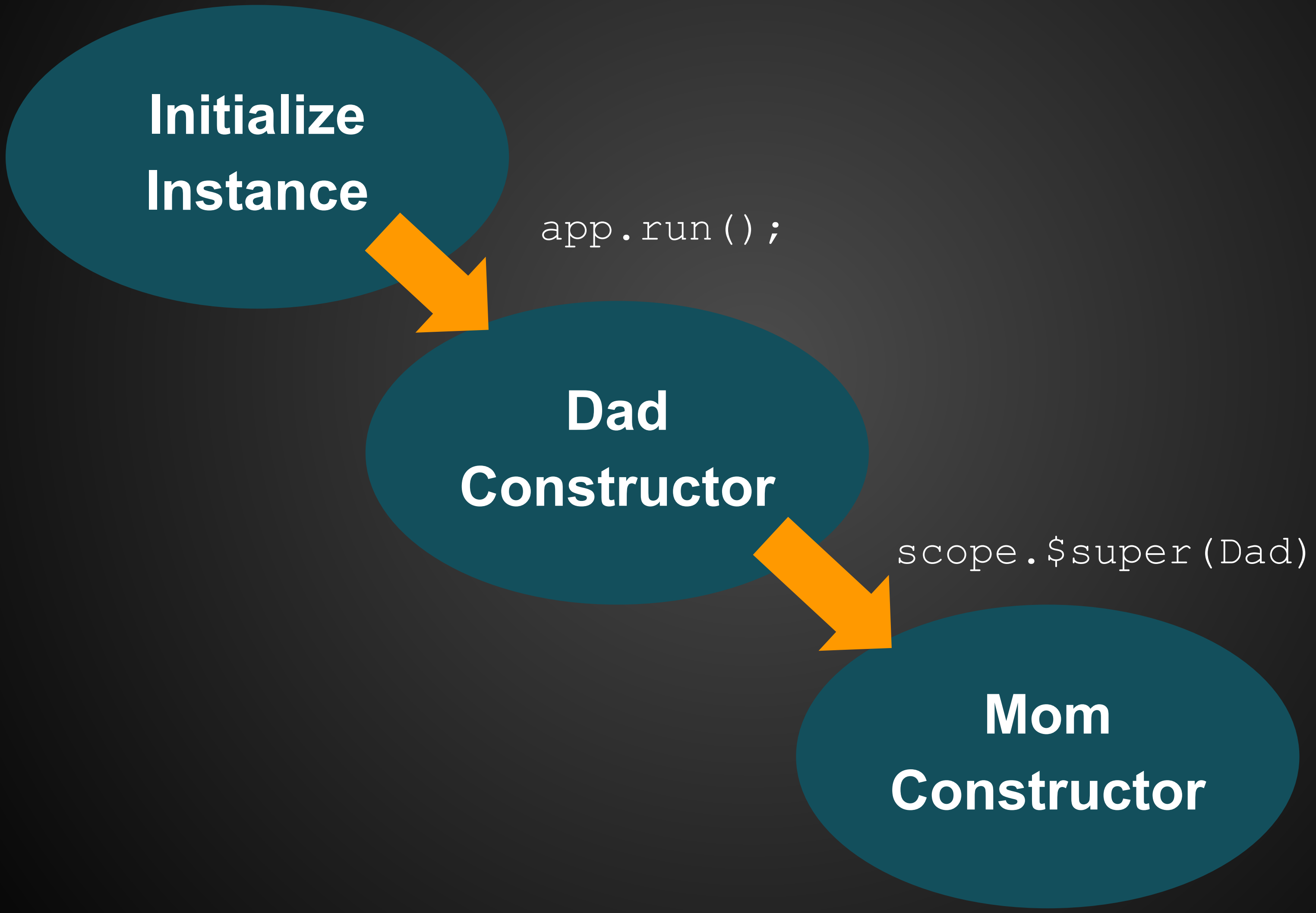
Dad inherits from Mom

dad.js

```
var Dad = module.exports = function(scope) {
        scope.$super(Dad);
};

Dad.prototype.helloDad = function() {
        console.log('hello dad!');
};
```

呼叫前一個
Constructor

新 Prototype 的初始化流程



Initialize Instance

`app.run();`

Dad Constructor

`scope.$super(Dad)`

Mom Constructor

scope.$super()

運用 super

取代 helloMom() 方法

How to extend

Added lines to dad.js

```
Dad.prototype.helloMom = function() {
	console.log('I Killed mom!');
};
```

Result

```
$ node app.js
I killed mom!
hello dad!
```

擴充 helloMom() 方法

How to extend

Added lines to dad.js

```
Dad.prototype.helloMom = function() {
    console.log('before mom');
    this.scope.$super(Dad, 'helloMom')();
};
```

Result

```
$ node app.js
before mom
hello mom!
hello dad!
```

來寫個網站吧

用 wag.js

Web Example

```
var wag = new Wag();
var app = wag.mixin([
    ‘web’
]);

app.run(function(feature) {
    feature.listen(8000);
});
```

wag.js 內建了一個 Engine

Web Feature with Express

web engine in wag.js

```
var Web = module.exports = function(scope) {
        scope.$super(Web);
        this.express = express();
        this.server = null;
};

Web.prototype.configure = function(callback) {
        callback();
};
```

Routers in web engine

```
Web.prototype.routers = function(callback) {
        this.express.get('/', function(req, res) {
                res.send('Hello! Wag Web!');
        });

        callback();
};
```

『/』頁面回傳字串

Run server in web engine

```
Web.prototype.listen = function(port) {
        var self = this;

        var _port = port || Web.Wag.settings.web.port;

        self.configure(function() {
                self.routers(function() {
                        self.server = self.express.listen(_port);
                });
        });
};
```

管理 Feature

Manage Your Features

來寫一個留言板吧

Let’s write a guestbook :-)

Guestbook

```
var wag = new Wag();
app.addPath('engineDirs', __dirname + '/engines');
var app = wag.mixin([
	‘web’,
	'guestbook'
]);

app.run(function(feature) {
	feature.listen(8000);
});
```

讓 wag.js 在此路徑下

尋找 feature engine

Guestbook Directory Structure

- engines/
  - web
  - guestbook
- public/
  - css
  - fonts
  - images
  - js
- views/
- app.js

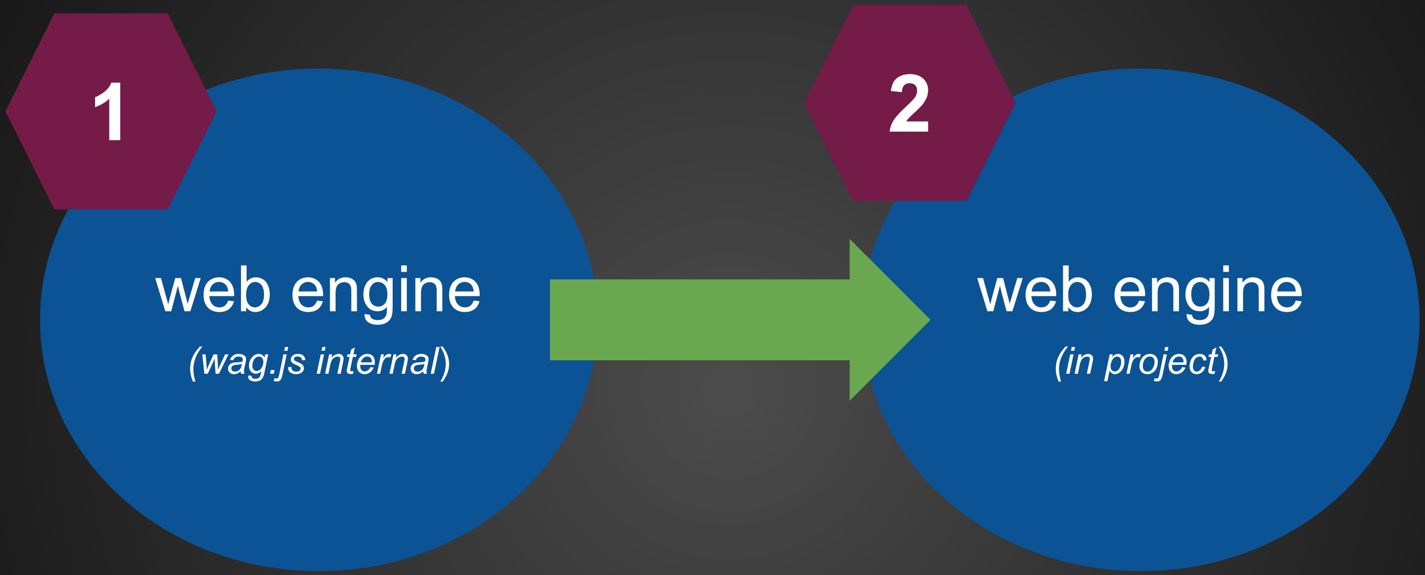
web engine

```
var path = require('path');
var express = require('express');
var bodyParser = require('body-parser');
var Web = module.exports = function(scope) {
	scope.$super(Web);
};
Web.prototype.configure = function(callback) {
	this.express.set('view engine', 'jade');
	this.express.set('views', path.join(__dirname, '..', '..', 'views'));
	this.express.use(express.static(path.join(__dirname, '..', '..', 'public')));
	this.express.use(bodyParser());
	callback();
};
Web.prototype.routers = function(callback) {
	callback();
};
```

增加 Views 和 Public 的支援

移除所有 Routes

web engine 的合成順序



```
app.addPath('engineDirs', path);
```

guestbook Feature - Constructor

```
var Guestbook = module.exports = function(scope) {
    scope.$super(Guestbook);
    this.articles = [
        {
            title: 'What\'s Wag.js',
            text: 'Wag.js is a feature-oriented based …..'
        },
        {
            title: 'What\'s News',
            text: 'Guestbook example was added.'
        }
    ];
};
```

guestbook Feature - Add Routes

```
Guestbook.prototype.routers = function(callback) {
    var self = this;
    // Call super to inherit
    self.scope.$super(Guestbook, 'routers')(function() {
        // Added route rules ...
        self.express.get('/', function(req, res) {
            res.render('index', {
                articles: self.articles
            });
        });
        // Added more route rules ...
};
```

理論上 Feature 都可重複合成使用

輪子不必重造

建立個 WPM 是可能的

wag.js package management is possible

wag.js 更多支援？

Settings support

這跟 FOP 無關

It’s not about FOP

```
app.addPath('settingDirs', __dirname + '/settings');
```

Settings Path

Settings 會在啟動時載入

Settings will be loaded on start up application

Get Settings in Wag.js

```
var password = scope.settings.test.password;
```

歡迎！如果你有興趣

Welcome to join wag.js

cfsghost

At Github

Thanks

Any Questions?