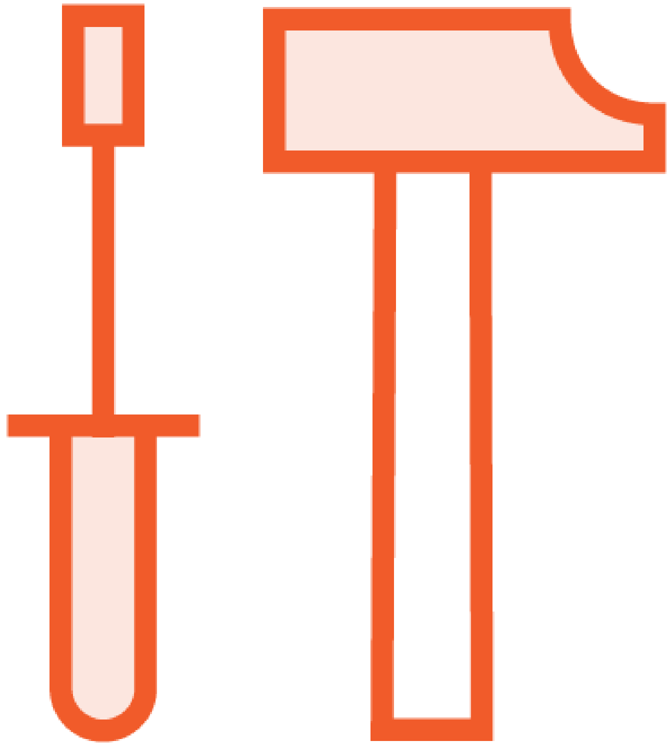
Creational Design Patterns

Jonathan Mills

@jonathanfmills www.jonathanfmills.com



Creational Design Patterns

Used to Construct New Objects

Adapting Creation to the Situation

Constructor Pattern

Use to create new objects with their own object scope.

The new keyword

Creates a brand new object

Links to an object prototype

Binds ‘this’ to the new object scope

Implicitly returns this

```
function ObjectName(param1, param2){
    this.param1 = param1;
    this.param2 = param2;
    this.toString = function () {
        return this.param1 + ' ' + this.param2;
    }
}
```

Constructor Pattern

Create objects from functions.

Demo

Lets Try This Out

```
function ObjectName(param1, param2){
    this.param1 = param1;
    this.param2 = param2;
    this.toString = function () {
        return this.param1 + ' ' + this.param2;
    }
}
```

Constructor Pattern

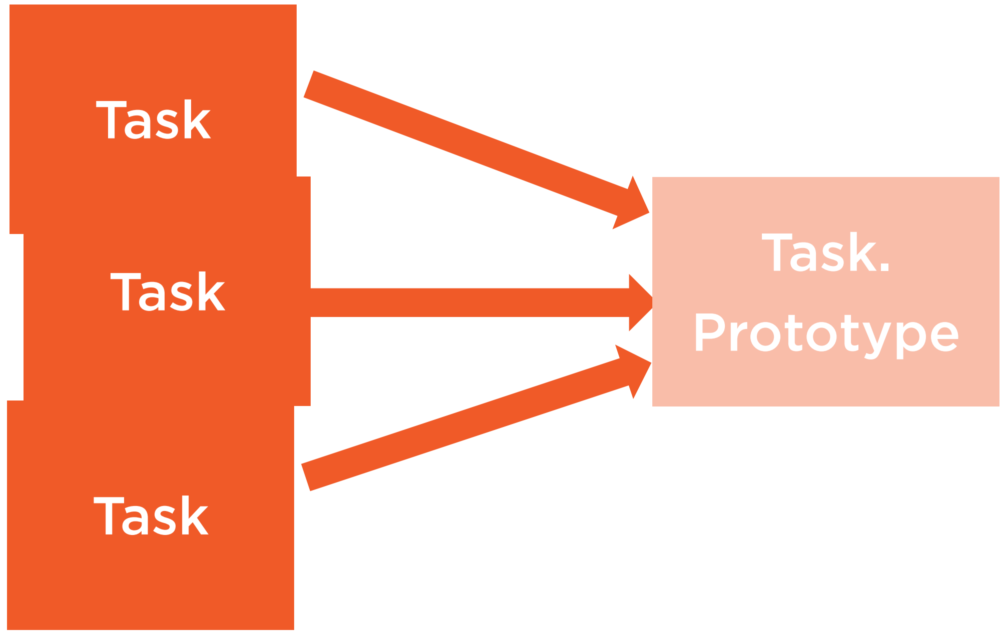
Drawbacks

Prototypes

Prototype

An encapsulation of properties that an object links to.

Prototypes



Demo

Prototypes

Demo

In the Node environment

Demo

In the angular environment

Demo

In EcmaScript2015

Module Pattern

Module Pattern

Simple Way to Encapsulate Methods

Creates a “Toolbox” of functions to use.

```
var Module = {
    method: function(){...},
    nextMethod: function(){...}
}
```

Module Pattern

Object Literal

```
var Module = function () {
    var privateVar = 'I am private...';
    return {
        method: function () {...
        },
        nextMethod: function () {...
        }
    }
}
```

Module Pattern

Wrap it in a function!

Demo

Module Pattern Demo

Demo

Module Pattern in Angular

Factory Pattern

A pattern used to simplify object creation.

Simplifies object creation

Creating different objects based on need

Repository Creation

Demo

Factory Pattern in our Node App

Singleton

Used to restrict an object to one instance of that object across the application.

Remembers the last time you used it

Hands the same instance back

Node.js uses CommonJS

Demo

Singletons in Node

Demo

Singletons in Angular

Summary

Creational Design Patterns

Constructor Pattern

Module Pattern

Factories

Singletons