JavaScript Master Seminar
Module Pattern

Sirma Gjorgievska
Johannes Fischer

Technische Universität München

September 07, 2015

Fakultät für Informatik

Agenda

- Introduction
- The Basics
- Augmentation
- Shared Private State
- Submodules
- Inheritance
- Demonstration
- Conclusion

Introduction

What is Module?

- Integral piece of robust application's architecture
- Keeps the units of code separated and organized

Introduction

Implementation of modules

- The Module pattern
- Object literal notation
- AMD modules
- CommonJS modules
- ECMAScript Harmony modules

Introduction

What is Module pattern?

- JavaScript design pattern
- Developed in 2003
- Private and public encapsulation
- Mimic classes in software engineering

Introduction

Advantages

- Cleaner approach for developers
- Supports private data
- Less clutter in global namespace
- Localization of functions and variables

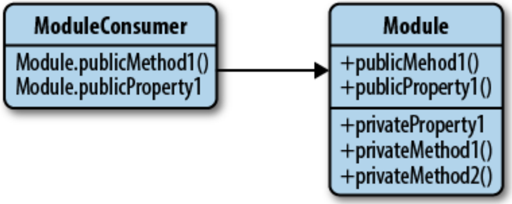
Introduction

Disadvantages

- Inability to create automated unit tests
- Lose of extendibility
- Problems when changing visibility of public/private members

The Basics

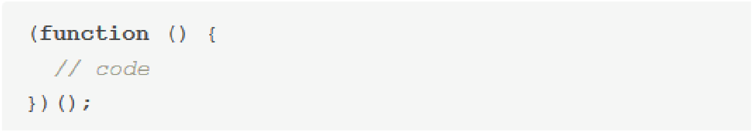
- Anonymous Closures
- Private methods
- Global Import
- Module Export



The Basics

Anonymous Closures

- Defined function is executed immediately
- Code inside the function lives in a **closure**
- It provides **privacy** and **state**
- Maintains access to all globals



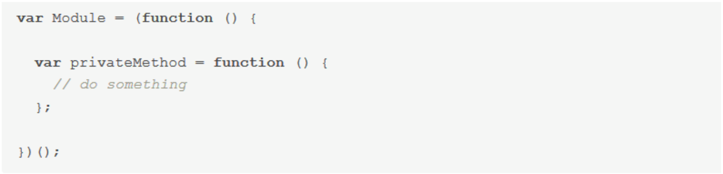
```
(function () {
  // code
})();
```

Figure: A simple anonymous closure

The Basics

Private methods

- Methods locally declared in modules
- Inaccessible outside of the scope defined



```
var Module = (function () {

  var privateMethod = function () {
    // do something
  };

})();
```

Figure: Private scope of a function

The Basics

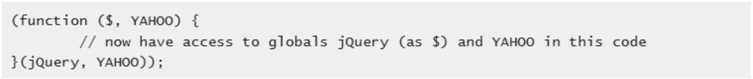
Implied Globals

- Hard-to-manage code
- Not obvious (to humans) which variables are global

The Basics

Global Import

- Better alternative
- Passing globals as parameters to anonymous function
- Clearer and faster approach
- Better efficiency and readability



```
(function ($, YAHOO) {
        // now have access to globals jQuery (as $) and YAHOO in this code
}(jQuery, YAHOO));
```

Figure: Importing of globals

The Basics

Module Export

- Declare globals for further use
- Return value of anonymous function
- Module variables readable afterwards
- Namespacing (avoids varname conflicts)

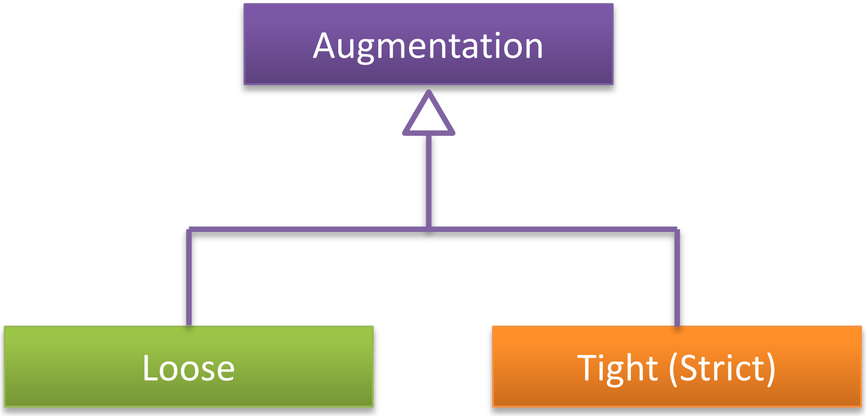
Augmentation

Problems:

- Entire module must be in one file
- Not extendable

Solution: **Augment modules**

Augmentation



Augmentation

Tight (Strict) Augmentation

Augmentation

Tight (Strict) Augmentation

- Parameter: MODULE
- Loading order has to be fixed
- Properties of earlier modules usable reliably
- Properties overwritable

Augmentation

Loose Augmentation

Augmentation

Loose Augmentation

- Parameter: MODULE || {}
- Loading order irrelevant
- Module files can be loaded in parallel

Augmentation

Tight Augmentation vs Loose Augmentation

Tight Augmentation	Loose Augmentation
- Allows overrides	- Cannot override safely
- Loading order fixed	- Loading order not fixed
- Parameter: MODULE	- Parameter: MODULE || {}

Disadvantage

- Inability to share private variables between files

Shared Private State

Shared Private State

- Variable _*private* identical for all modules
- Only accessible from old module files
- Unlocks _private for later module file loading

Submodules

- Creation is same like regular modules
- All the advanced capabilities of normal modules

Inheritance

- Parent module has to exist
- New module now has parent and parent's properties
- m.parent needs to be set last

Live Demonstration

Conclusion

- Conclusion
  - We just showed the tip of the iceberg, there are much more details and many more interesting features.
  - The AngularJS official website is a great place to start learning: https://docs.angularjs.org/
  - Angular is a complete client-side solution which is opinionated about how a CRUD (Create, Read, Update, Delete) application should be built.
  - Angular and its mentality is the future of the web, so if you are dealing with JavaScript in your career, sooner or later, you will end up learning such a framework.
  - Finally we think Angular is the coolest thing happened to client side JavaScript since jQuery!

Thank You!