

# JavaScript 命名空間

---

向 皓 田

2018/11/2

- 
- ECMAScript 5 並無命名空間
  - 本文內容係針對以下專案的程式碼做說明  
<https://github.com/leoshiang/dolphin-js5/blob/master/Source/Core/Namespace.js>

# 命名空間

---

- 可以義多個命名空間樹
- 可以定義階層式的物件結構
- 可以使用 Unicode
- 支援 Reflection

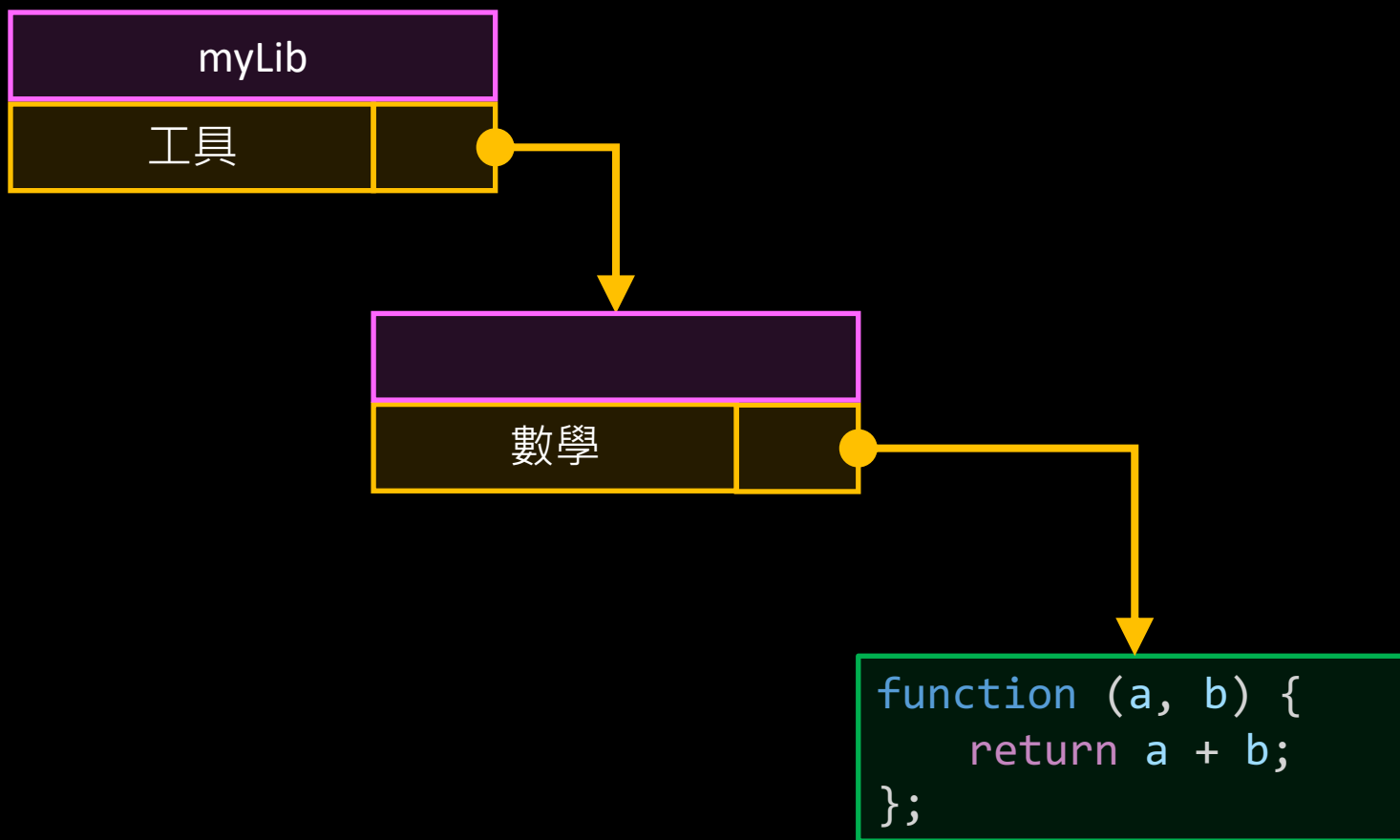
# 命名空間範例

---

```
var myLib = new Namespace();  
  
myLib.namespace('工具.數學').相加 = function (a, b) {  
    return a + b;  
};  
  
var result = myLib.工具.數學.相加(1, 2);
```

# 資料結構

---



# Show me the code

---

```
function Namespace() {}
```

```
Namespace.prototype = {  
  constructor: Namespace,  
  
  traverse: function(namespace, callback) { },  
  
  namespace: function(namespace) { },  
  
  reflection: function(namespace) { }  
};
```

```
var Dolphin = new Namespace();
```

# Namespace.traverse

---

```
traverse: function(namespace, callback) {
    if (namespace.trim() === '') { return this; }
    var parts = namespace.split('.');
    for (var i = 0; i < parts.length; i++) {
        var part = parts[i].trim();
        if (part === '')
            throw new Dolphin.InvalidNamespaceException();
        parts[i] = part;
    }

    var node = this;
    parts.forEach(function(name) {
        node = callback(node, name);
    });

    return node;
},
```

# Namespace.traverse

---

myLib.namespace('工具.數學')



```
var parts = namespace.split('.');
for (var i = 0; i < parts.length; i++) {
    var part = parts[i].trim();
    if (part === '')
        throw new Dolphin.InvalidNamespaceException();
    parts[i] = part;
}
```



# Namespace.traverse

```
var node = this;  
parts.forEach(function(name) {  
  node = callback(node, name);  
});
```

```
return node;
```

```
namespace: function(namespace) {  
  return this.traverse(namespace, function(node, item) {  
    node[item] = node[item] || {};  
    return node[item];  
  });  
},
```

變數	值
parts	['工具', '數學']
node	{}

變數	值
parts	['工具', '數學']
node	{'工具': {}}