JavaScript 核心觀念（下）

JavaScript 各種特別的地方

是誰住在六角學院裡~OH~~~

萬惡的 This

請問 console.log 結果為何？

```
console.log(this);
```

```
» this
← ▼ Window
    ▶ 0: Restricted about:srcdoc
    ▶ _: function ur(e) ↱≡
    ▶ browser: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
    ▶ chrome: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
    ▶ <default properties>
      Symbol(Symbol.toStringTag): "Window"
    ▶ <prototype>: WindowPrototype { … }
```

請問 console.log 結果為何？

```
function a (){
  console.log(this);
}

a();
```

```
» function a (){
    console.log(this);
  }

  a();
```

```
▼ Window
  ▶ _: function ur(e)
  ▶ a: function a()
  ▶ browser: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
  ▶ chrome: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
  ▶ <default properties>
    Symbol(Symbol.toStringTag): "Window"
  ▶ <prototype>: WindowPrototype { … }
```

請問 console.log 結果為何？

```
var b = function () {
  console.log(this);
};

b();
```

```
var b = function () {
  console.log(this);
};

b();
```

```
▼ Window
  ▶ _: function ur(e) ↱≡
  ▶ b: function b()
  ▶ browser: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
  ▶ chrome: Object { manifest: Getter & Setter, events: Getter & Setter, types: Getter & Setter, … }
  ▶ <default properties>
    Symbol(Symbol.toStringTag): "Window"
  ▶ <prototype>: WindowPrototype { … }
```

具名陳述式

```
function a (){
  console.log(this);
}

a();
```

匿名表達式

```
var b = function () {
  console.log(this);
};

b();
```

陳述式與表達式 補充文章

以下是兩篇對於陳述式與表達式的說明與描述

前端，沒有極限 – 卡斯柏

Welcome.Web.World – Ray

請問 console.log 結果為何？

```
function a (){
  this.myName = 'Ray';
}

a();

setTimeout(function() {
  console.log(myName);
}, 5000);
```

```
function a (){
  this.myName = 'Ray';
}

a();

setTimeout(function() {
    console.log(myName);
}, 5000);
```

```
2
```

```
Ray
```

請問 console.log 結果為何？

```
function a (){
  var myName = 'Ray';
}

a();

setTimeout(function() {
  console.log(myName);
}, 5000);
```

```
function a (){
  var myName = 'Ray';
}

a();

setTimeout(function() {
  console.log(myName);
}, 5000);
2
Uncaught ReferenceError: myName is not defined
    <anonymous>          debugger eval code:9
    setTimeout handler*  debugger eval code:8
  [了解更多]
```

簡易呼叫 (Simple Call)

直接呼叫這個 this

this
指向重點

This 的指向與怎麼宣告定義它無關

而是與你怎麼呼叫它有關。

```
var myName = '小明';

var app = {
  myName: 'Ray',
  hello: function() {
    console.log(this.myName);
  },
};

app.hello();
```

1. 小明

2. Ray

```
» var myName = '小明';

  var app = {
    myName: 'Ray',
    hello: function() {
      console.log(this.myName);
    },
  };

  app.hello();
Ray
```

```
var myName = '小明';

var app = {
  myName: 'Ray',
  hello: function() {
    console.log(this.myName);
  },
};

var hello = app.hello;

hello();
```

1. 小明

2. Ray

```
var myName = '小明';

  var app = {
    myName: 'Ray',
    hello: function() {
      console.log(this.myName);
    },
  };

  var hello = app.hello;

  hello();
```

小明

```
var myName = '小明';

var app = {
  myName: 'Ray',
  hello: function() {
    setTimeout(function() {
      console.log(this.myName);
    }, 5000);
  },
};

app.hello();
```

1. 小明

2. Ray

```
var myName = '小明';

  var app = {
    myName: 'Ray',
    hello: function() {
      setTimeout(function() {
        console.log(this.myName);
      }, 5000);
    },
  };

  app.hello();
```

```
undefined
```

```
小明
```

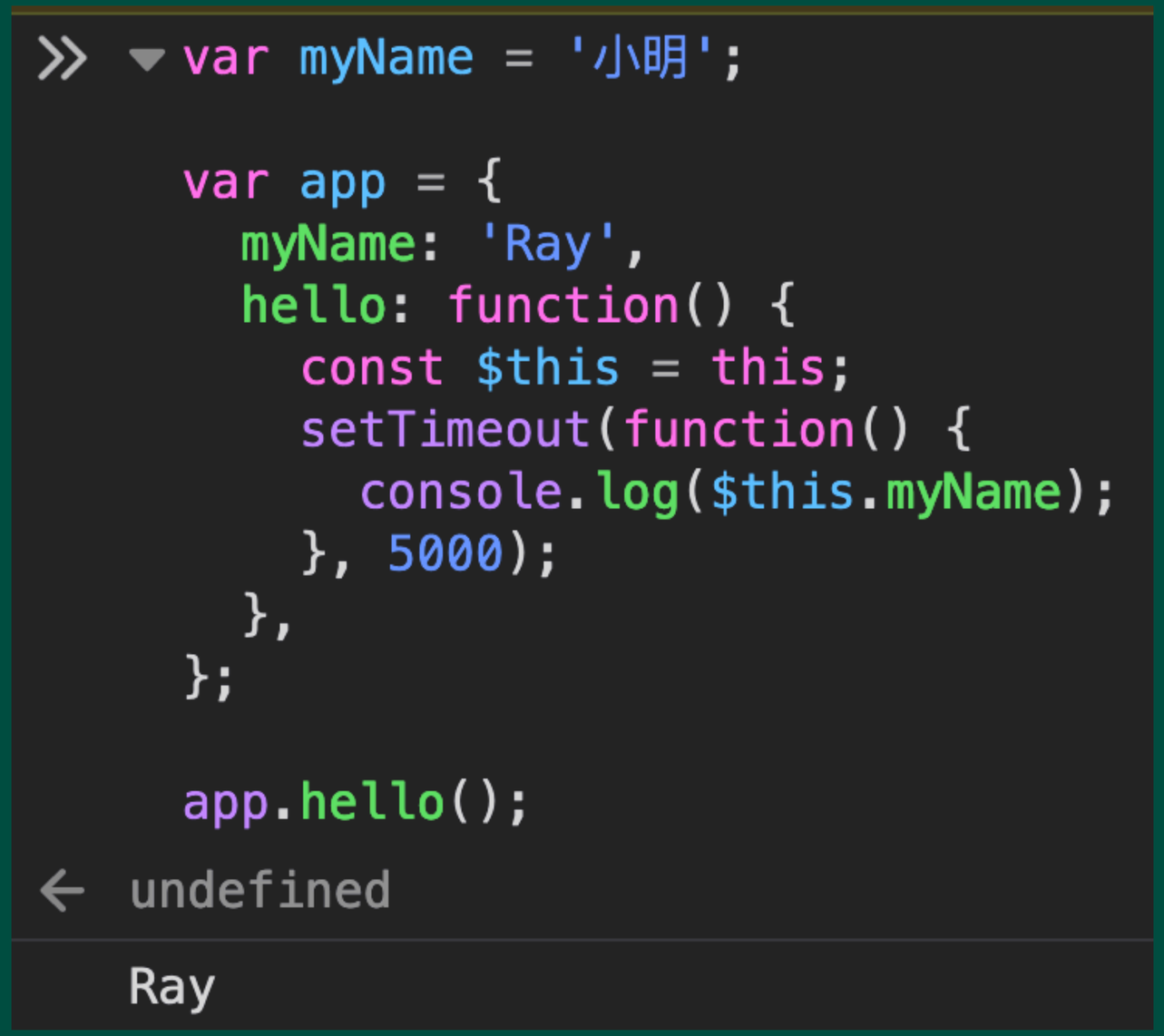
```
var myName = '小明';

var app = {
  myName: 'Ray',
  hello: function() {
    const $this = this;
    setTimeout(function() {
      console.log($this.myName);
    }, 5000);
  },
};

app.hello();
```

1. 小明

2. Ray


var myName = '小明';
var app = {
myName: 'Ray',
hello: function() {
const $this = this;
setTimeout(function() {
console.log($this.myName);
}, 5000);
},
};
app.hello();
undefined
Ray

```
var myName = '小明';

var app = {
  myName: 'Ray',
  hello: function() {
    var myName = 'QQ';
    var sayHi = function () {
      console.log(this.myName);
    }
    sayHi();
  },
};

app.hello();
```

1. 小明　　2. Ray　　3. QQ

```
var myName = '小明';

  var app = {
    myName: 'Ray',
    hello: function() {
      var myName = 'QQ';
      var sayHi = function () {
        console.log(this.myName);
      }
      sayHi();
    },
  };

  app.hello();
```

```
小明
```

```
function sayHi() {
  console.log(this.myName);
}

var myName = '小明';

var obj = {
  myName: 'Ray',
  hello: sayHi,
}

obj.hello();
```

1. 小明

2. Ray

```
» ▼ function sayHi() {
      console.log(this.myName);
  }

  var myName = '小明';

  var obj = {
    myName: 'Ray',
    hello: sayHi,
  }

  obj.hello();
```

```
Ray
```

```
function sayHi() {
  console.log(this.myName);
}

var myName = '小明';

var obj = {
  myName: 'Ray',
  hello: sayHi(),
}

obj.hello;
```

1. 小明

2. Ray

```
» ▼ function sayHi() {
      console.log(this.myName);
  }

  var myName = '小明';

  var obj = {
    myName: 'Ray',
    hello: sayHi(),
  }

  obj.hello;
```

小明

優先性
與相依性

```
var a = 2 + 2 * 2 / 2;

console.log(a); // 4
```

```
console.log(1 < 2 < 3); // true
```

```
console.log(3 > 2 > 1); // ?
```

```
» console.log(3 > 2 > 1); // ?
  false
```

運算子優先序

11	Less Than	從左至右	… < …
	Less Than Or Equal		… <= …
	Greater Than		… > …
	Greater Than Or Equal		… >= …
	in		… in …
	instanceof		… instanceof …

課後學習資源

優先序與相依性

表達式與陳述式

函式與 this 的運作

六角學院

是 Ray 不是 Array

六角學院