# This 到底指向什么?

# 在全局环境中……

```javascript
console.log(this.document === document); // true

// In web browsers, the window object is also the
global object:
console.log(this === window); // true

this.a = 37;
console.log(window.a); // 37
```

# 简单函数中......

```javascript
// Simple Call
function f1(){
  return this;
}

f1() === window; // global object

// Strict Mode
function f2(){
  "use strict"; // see strict mode
  return this;
}

f2() === undefined;
```

# 作为对象方法被调用......

```
var o = {
  prop: 37,
  f: function() {
    return this.prop;
  }
};
console.log(o.f()); // logs 37
```

```
var o2 = {prop: 37};
function independent() {
  return this.prop;
}
o2.f = independent;
console.log(o2.f()); // logs 37
```

# 在构造函数中……

```
function C(){
  this.a = 37;
}

var o = new C();
console.log(o.a); // logs 37


function C2(){
  this.a = 37; // 实际上相当于作废了
  return {a:38};
}

o = new C2();
console.log(o.a); // logs 38
```

# 在事件处理器中……

```javascript
// When called as a listener, turns the related element blue
function bluify(e){
  // Always true
  console.log(this === e.currentTarget);
  // true when currentTarget and target are the same object
  console.log(this === e.target);
  this.style.backgroundColor = '#A5D9F3';
}

// Get a list of every element in the document
var elements = document.getElementsByTagName('*');

// Add bluify as a click listener so when the
// element is clicked on, it turns blue
for(var i=0 ; i<elements.length ; i++){
  elements[i].addEventListener('click', bluify, false);
}
```

# e.target vs e.currentTarget

- target = element that triggered event.

- currentTarget = element that listens to event.

# 练习

```
if (true) {
    // this
}
```

```
var obj = {
    someData: "a string"
};

function myFun() {
    // this
}

obj.staticFunction = myFun;

obj.staticFunction();
```

```
var obj = {
    myMethod : function () {
        // this
    }
};
var myFun = obj.myMethod;
myFun();
```