Truth, Equality and JavaScript

- [1] == true // true
- [0] == false // true
- [2] == true // false
- [0,1] == true // false
- [‘’] == false // true
- "potato" == true // false
- "potato" == false // false

条件运算符

ToBoolean算法

Argument Type	*Result*
Undefined	**false**
Null	**false**
Boolean	The result equals the input argument (no conversion).
Number	The result is **false if** the argument is **+0**, **−0**, or **NaN**; otherwise the result is **true**.
String	The result is **false if** the argument is the empty String (its length is zero); otherwise the result is **true**.
Object	**true**.

- new Boolean(false) // true
- new String(“”) // true
- new Number(NaN) // true

Type(x)	*Values*	*Result*
Type(x) different from Type(y)		**false**
Undefined or Null		**true**
Number	x same value as y (but not `NaN`)	**true**
String	x and y are identical characters	**true**
Boolean	x and y are both true or both false	**true**
Object	x and y reference same object	**true**
otherwise…		**false**

= =

Type(x)	*Type(y)*	*Result*
x and y are the same type		**See Strict Equality (===) Algorithm**
null	Undefined	**true**
Undefined	null	**true**
Number	String	x == toNumber(y)
String	Number	toNumber(x) == y
Boolean	(any)	toNumber(x) == y
(any)	Boolean	x == toNumber(y)
String or Number	Object	x == toPrimitive(y)
Object	String or Number	toPrimitive(x) == y
otherwise…		**false**

ToNumber

Argument Type	*Result*
Undefined	**NaN**
Null	**+0**
Boolean	The result is **1** if the argument is **true**. The result is **+0** if the argument is false.
Number	The result equals the input argument (no conversion).
String	In effect evaluates Number(*string*) “abc” -> NaN “123” -> 123
Object	Apply the following steps: 1. Let *primValue* be ToPrimitive(*input argument*, hint Number). 2. Return ToNumber(*primValue*).

ToPrimitive

Argument Type	***Result***
Object	(in the case of equality operator coercion) if `valueOf` returns a primitive, return it. Otherwise if `toString` returns a primitive return it. Otherwise throw an error
otherwise…	The result equals the input argument (no conversion).

- "potato" == true;
- "potato" == 1;
- NaN == 1;

- crazyNumeric = new Number(1);
- crazyNumeric.toString = function() {return "2"};
- crazyNumeric == 1;
- crazyNumeric.valueOf() // 1
- 1 == 1 //true

- ‘’ == 0 // true
- 0 == ‘0’ // true
- ‘’ == ‘0’ // false

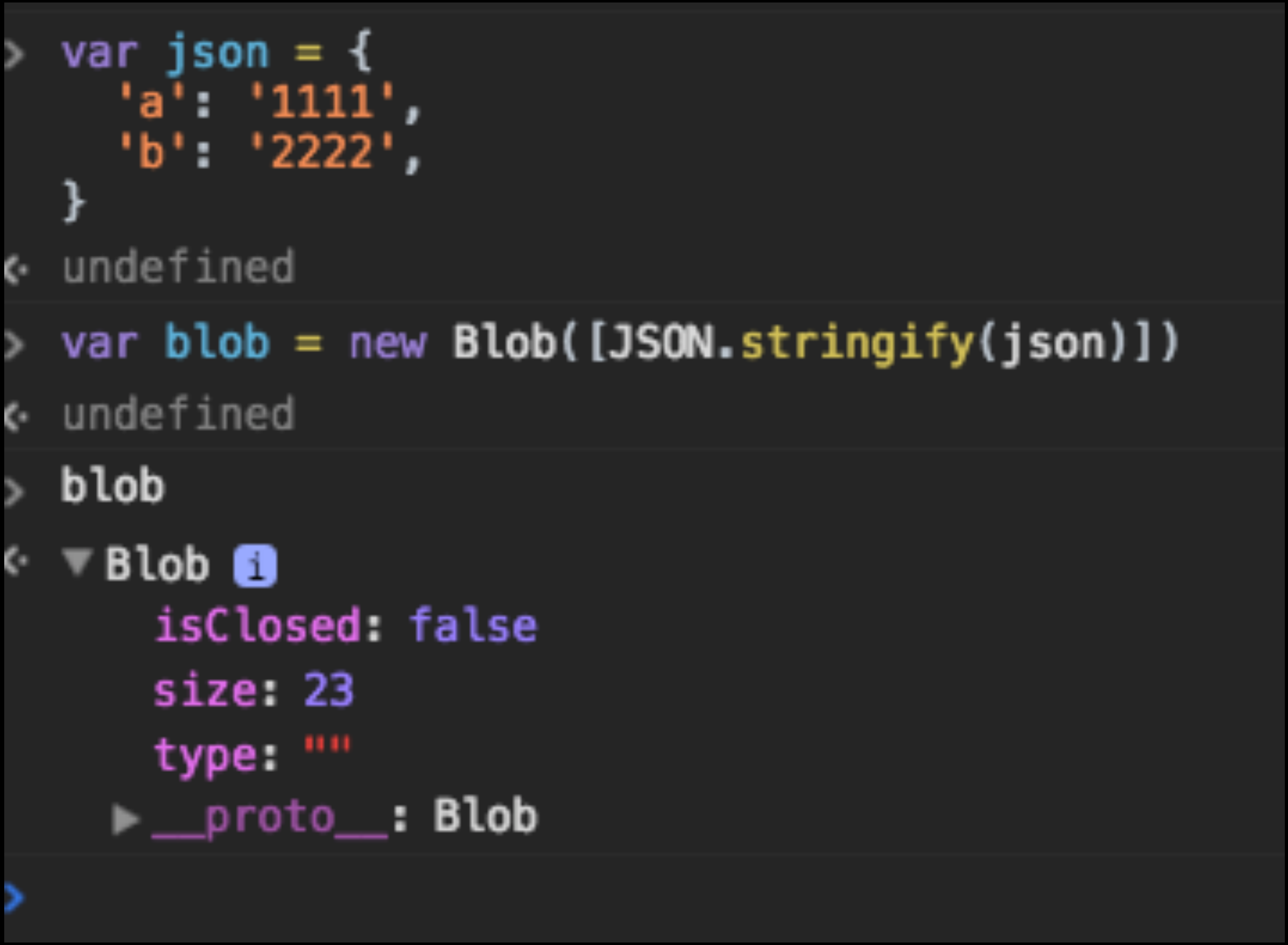
Unnecessary

- if (typeof myVar === “function")
- if (myArray.length == 3)

Blob类型

```
var blob = new Blob([“Hello World”], {type: “text/html”})
```

sizeof ?



```
> var json = {
    'a': '1111',
    'b': '2222',
  }
<· undefined
> var blob = new Blob([JSON.stringify(json)])
<· undefined
> blob
<· ▼Blob i
      isClosed: false
      size: 23
      type: ""
    ▶__proto__: Blob
>
```

utf-8

- 000000 – 00007F 1
- 000080 – 0007FF 2
- 000800 – 00D7FF 3
- 00E000 – 00FFFF 3
- 010000 – 10FFFF 4

utf-16

- 000000 – 00FFFF 2
- 010000 – 10FFFF 4

```
str.charCodeAt(i)
```