15.2.4.4 Object.prototype.valueOf ( )

When the valueOf method is called, the following steps are taken:

1. Let O be the result of calling ToObject passing the this value as the argument.

2. If O is the result of calling the Object constructor with a host object (15.2.2.1), then

a. Return either O or another value such as the host object originally passed to the constructor. The specific result that is returned is implementation-defined.

3. Return O.

**针对调用ToObject时传入的不同参数（this），返回值分别如**下：   
  
1、this为宿主对象时，返回值取决于浏览器的实现，即不同浏览器的返回可能不同  
2、this不是宿主对象，则返回ToObject(this)的值

|  |  |
| --- | --- |
| **参数类型** | **返回结果** |
| Undefined | 抛出TypeError异常 |
| Null | 抛出TypeError异常 |
| Number | 创建一个Number对象，它内部的初始值为传入的参数值 |
| String | 创建一个String对象，它内部的初始值为传入的参数值 |
| Boolean | 创建一个Boolean对象，它内部的初始值为传入的参数值 |
| Object | 返回传入的参数（无转换） |

根据Object.prototype.valueOf的定义，以及抽象方法ToObject的描述，可得下表

|  |  |
| --- | --- |
| **obj类型** | **Object.prototype.valueOf.call(obj)返回结果** |
| Undefined | 抛出TypeError异常 |
| Null | 抛出TypeError异常 |
| Number | Number类型的对象，值等于obj |
| String | String类型的对象，值等于obj |
| Boolean | Boolean类型的对象，值等于obj |
| Object | obj对象本身 |

示例：

var num = 123;

console.log(num.valueOf()); //输出：123

console.log(num.valueOf()); //输出：'number'

var unde = undefined;

console.log(Object.prototype.valueOf.call(unde)); //输出：'TypeError: Cannot convert null to object'

var obj = {name:'casper'};

var linkObj = obj.valueOf();

linkObj.name = 'change';

console.log(linkObj.name); //输出：'change' ...说明obj.valueOf()返回的是对象自身

当Object.prototype.valueOf调用时，参数为Array、Function类型的对象时，返回的结果也为对象自身。

var num = 123; //通过对象字面量声明console.log(typeof num); //输出：'number'   
var num = new Number(123); //通过构造方法声明console.log(typeof num); //输出：'object'

原来Number有属于自身的原型valueOf方法，不是直接从Object.prototype上继承下来，类似的，Boolean、String也有自己的原型valueOf方法，归纳如下：

|  |  |
| --- | --- |
| **类型** | **是否有属于自己的原型valueOf方法** |
| Undefined | 无 |
| Null | 无 |
| Number | 有，Number.prototype.valueOf |
| String | 有，String.prototype.valueOf |
| Boolean | 有，Boolean.prototype.valueOf |
| Object | - |

此处之外，Array、Function并没有自己的原型valueOf方法，见规范说明：