进阶–补环境自吐 3

书接上集携程（进度到 HTMLElement 阶段）

补环境自吐 2:https://workspace.dingtalk.com/pj1VVxiKVziEKn6q3nPBfF

补环境自吐 1:https://workspace.dingtalk.com/oCeRMbeikBM7vNjxuXFbgv

推荐看的重点：
3+6+12+13

1

HTMLCollection 本质是一个数组，但是也有对象自己的函数。common_proxy 函数可以加一个 opt 对象，存储 identifier，这样就可以使得 proxy 有自己的名字。这样我返回了一个对象之后，后续再对这个对象做了什么操作，也可以继续监控

JavaScript

```
get children(){
  return common_proxy([],{
     identifier:'document.body.children'
  }
}
```

2

随后，我们发现后续调用了一个 fbejkbakrbadskfe 对象，猜测可能是在携程页面上的东西，先不去管他

3

随后，发现系统报错：

JavaScript

```
[regexp /vm|repl|bootstrapNodeJSCore|tryModeleLoad|evalmachine|runInContext/g].text [Argu
'0':TypeError: Cannot read propertied of undefined(reading 'apply') \n"
......
```

regexp 代表是正则，后面这些都是一些关键字。代码是故意抛出异常，用关键字看调用堆栈里是否存在关键字。如果存在就代表你不是个正常的浏览器（比如 node 主动报错，堆栈中就有 node:replj 就会击中风控）

4

window.Image 同理，按照公式添加 Image

```JavaScript
window.Image=Image
```

5

随后，发现对 HTMLElement 调用了方法 getAttribute 方法，在浏览器上看属性描述符，同理进行定义

```JavaScript
#在代码里构造
Object.defineProperty(HTMLhtmlElement.prototype , 'getAttribute',{
  "writable": true,
  "enumerable": true,
  "configurable": true,
  "value": common_proxy(getAttribute)
})
```

这里有个问题，因为在浏览器上查看原型链，其实 getAttribute 是在 Element 上的。所以正规的写法应该是写在 Element 上，然后让 HTMLhtmlElement 继承它。这里因为没有检测，无所谓放在哪里了。

6

navigtor.ownKeys 是获取 navigtor 所有属性，用 indexOf 查看是否有 webdriver 关键字，所以可以说，发现了 getAttribute、ownKeys 或者 indexOf。大多都是检测自动化工具。

因此，我们构造 getAttribute()方法可以先返回 null，以通过这些检验

7

看日志中，使用 createElement 创建了我们定义的对象，然后想取 tagName 未果。因此我们需要补充 tagName 对象。

```JavaScript
get tagName(){
  return 'DIV'
}
```

8

对新构造的对象又调用了 appendChild(),我们把他保护起来定义一下。（跟上面那个情况相同，最好写在基类里,我们在网页上看，最基类是 node 对象）

JavaScript

```
func_set_natvie(createElement)
Object.defineProperty(Node.prototype,'appendChild',{
  "writable": true,
  "enumerable":true,
  "configurable":true,
  "value": common_proxy(getAttribute)
})
```

9

然后又取了 style 标签，style 先补上

JavaScript

```
get style(){
  return common_proxy(new CSSStyleDeclaration)
}
```

10

发现 createElement 了一个 tagName 为 a 的对象，过程非常雷同继续添加即可

JavaScript

```
function createElement(tagName){
  switch(tagName){
    case 'div':
      return common_proxy(new HTMLDivElement)
    case 'a':
      return common_proxy(new HTMLAnchorElement)
    default:
      console,log(`创建 ${tagName}`)
  }
}
```

11

尝试取了 navigator.ownKeys 返回了 0，实际上是对的。因为在浏览器上，Object.keys[navigator] 的返回是空[]，因此返回 0 就是对的。

12

JavaScript

```
[function appendChild].apply ⇒ HTMLDivElement {} [ HTMLDivElement {} ]
```

这句话代表了先连续创建了两个 Div 标签，然后把第二个创建的追加到第一个创建下面去，根据官方文档的定义 https://developer.mozilla.org/zh-CN/docs/Web/API/Node/appendChild

纯文本

Node.appendChild() 方法将一个节点附加到指定父节点的子节点列表的末尾处。
如果将被插入的节点已经存在于当前文档的文档树中，
那么 appendChild() 只会将它从原先的位置移动到新的位置（不需要事先移除要移动的节点）。

特别的，风控使用了 A.append(B) 然后 B.append(A)，如果在浏览器上会发现报错，但是补环境补的不合适不会报错。因此我们应该思考一下，如何抽象这个逻辑。我们只需要加一个判断，先给每个对象增加一个a.childnood=[],随后添加的时候维护这个字段，正确的添加即可

JavaScript

```
function appendChild(ele){
  if(ele._childNodes.includes(this)){
    throw Error(`Failed to execute 'appendChild' on 'Node': The new child element contain
  }
  this._childNodes.push(ele)
}
```

13

对 divElement 进行了 set height →20,人后对 divElement 查询 offsetHeight 观察浏览器的情况，把没有赋值的进行赋值。【这堂课只是提及，智海并没有完整实现（也有可能是视频录漏了，或者智海以前已经补好了直接跳过了），我在这里添加了我实操的结果】

可以在浏览器看到，通过测试，我把一个有高度的子标签添加到父标签后，父标签被撑大了

JavaScript

```
// 创建一个父 div 元素
const parentDiv = document.createElement('div');
document.body.appendChild(parentDiv); // 将父 div 添加到文档中
// 查看 div 的 offsetHeight
console.log('offsetHeight:', parentDiv.offsetHeight);
// 创建一个新的 div 元素
const childDiv = document.createElement('div');
childDiv.style.height = '20px'; // 设置子 div 高度
parentDiv.appendChild(childDiv); // 将子 div 添加到父 div 下方
// 查看 div 的 offsetHeight
console.log('offsetHeight:', parentDiv.offsetHeight);

结果：
offsetHeight: 0
offsetHeight: 20
```

JavaScript

```
#代码
class CSSStyleDeclaration {
    constructor() {
        this._height = ''
    }
    get[Symbol.toStringTag]() {
        return "CSSStyleDeclaration"
    }

    set height(v) {
        return this._height = v
```

```
    }
    get height() {
        return this._height
    }
}
```

重新写获取高度的offsetHeight方法，如果父标签没有设置height且有子标签，取子标签中最高的height

```
#正确的日志
[object HTMLDivElement 0.4718440245312423].get =>  style CSSStyleDeclaration { _height: '
[object CSSStyleDeclaration].set =>  height 20px
[object HTMLDivElement 0.4718440245312423].get =>  offsetHeight 0
[object HTMLDocument].get =>  body HTMLBodyElement { _childNodes: [] }
[object HTMLBodyElement].get =>  appendChild [Function: appendChild]
[function appendChild].get =>  apply [Function: apply]
[function appendChild].apply =>  HTMLBodyElement {
  _childNodes: [
    HTMLDivElement [HTMLDivElement 0.4718440245312423] {
      _id: 0.4718440245312423,
      _childNodes: []
    }
  ]
} [
  HTMLDivElement [HTMLDivElement 0.4718440245312423] {
    _id: 0.4718440245312423,
    _childNodes: []
  }
] undefined
[object HTMLDivElement 0.4718440245312423].get =>  style CSSStyleDeclaration { _height: '
[object CSSStyleDeclaration].get =>  height 20px
[object HTMLDivElement 0.4718440245312423].get =>  offsetHeight 20
```