

DOMラッパーについて聞いてください

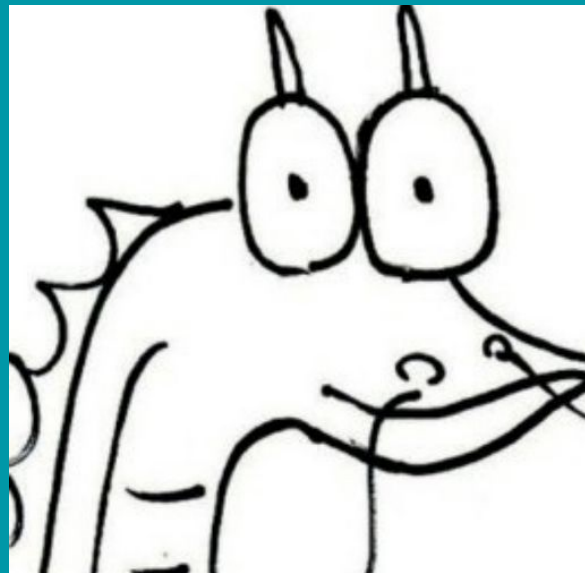
DOMラッパーはじまるよ!!! 聞いてください

DOMラッパーはじまるよ!!! 聞いてください



自己紹介

- canalun (@i_am_canalun)
- フロントエンド
- DOM、ブラウザ



趣味でやっていること

- DOMDOMタイムス
(zennで週刊連載)



- TAPL.ts
(TAPLの輪読会)



趣味でやっていること

- DOMDOMタイムス
(zennで週刊連載)



- TAPL.ts
(TAPLの輪読会)

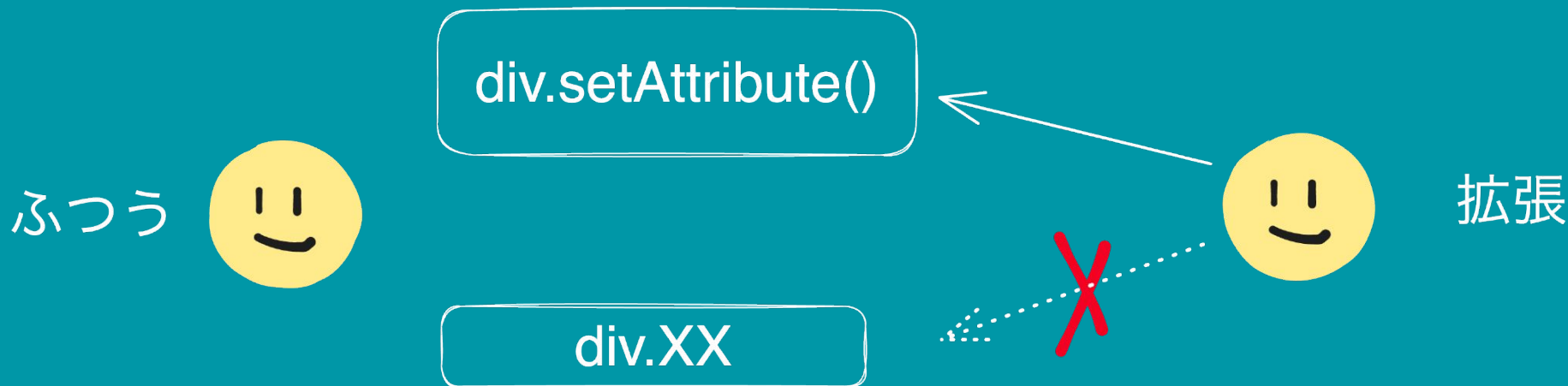
来週から型推論の章！
TVの前みんなもチェック！



DOMに関する実験をしてみましょう

実験タイム

同じDOMを見ているはずなのに……？



同じDOMを見ているはずなのに……？

ふつう

今日はこの現象を深掘りしてみよう！

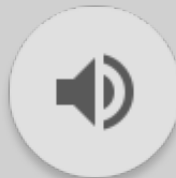
拡張

同じDOMを見ているはずなのに……？

ふつう

今日はこの現象を深掘りしてみよう！

拡張



今日のまとめ

- JSで扱っているのはDOMの実体ではなくDOMラッパーである
- 各worldはDOMを共有するが、ラッパーは共有されない

結論

- JSで扱っているのはDOMの実体ではなくDOMラッパーである
- 各worldはDOMを共有するが、ラッパーは共有されない



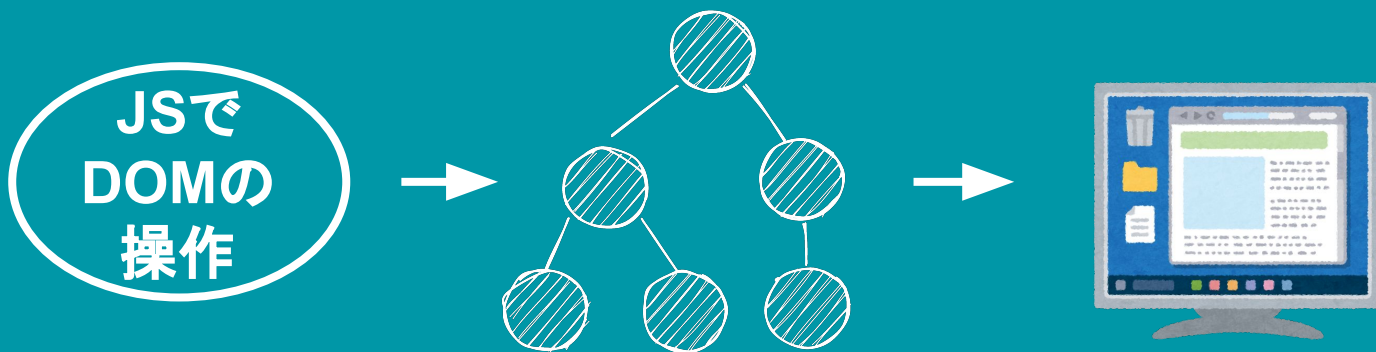
DOMラッパー??



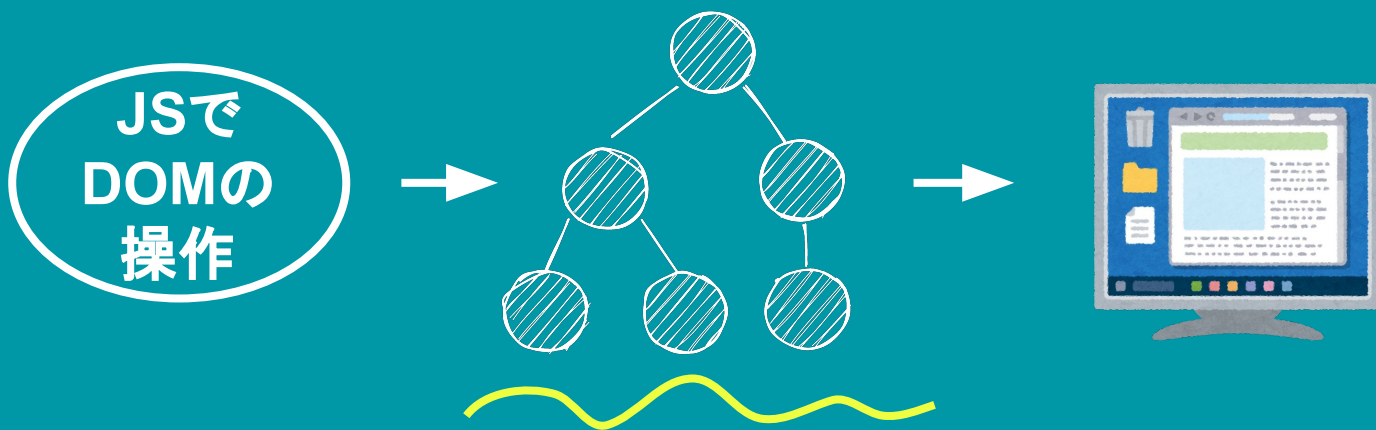
Own work, CC BY-SA 3.0, via Wikimedia Commons



JSでDOMを操作して画面を変更するときのイメージ

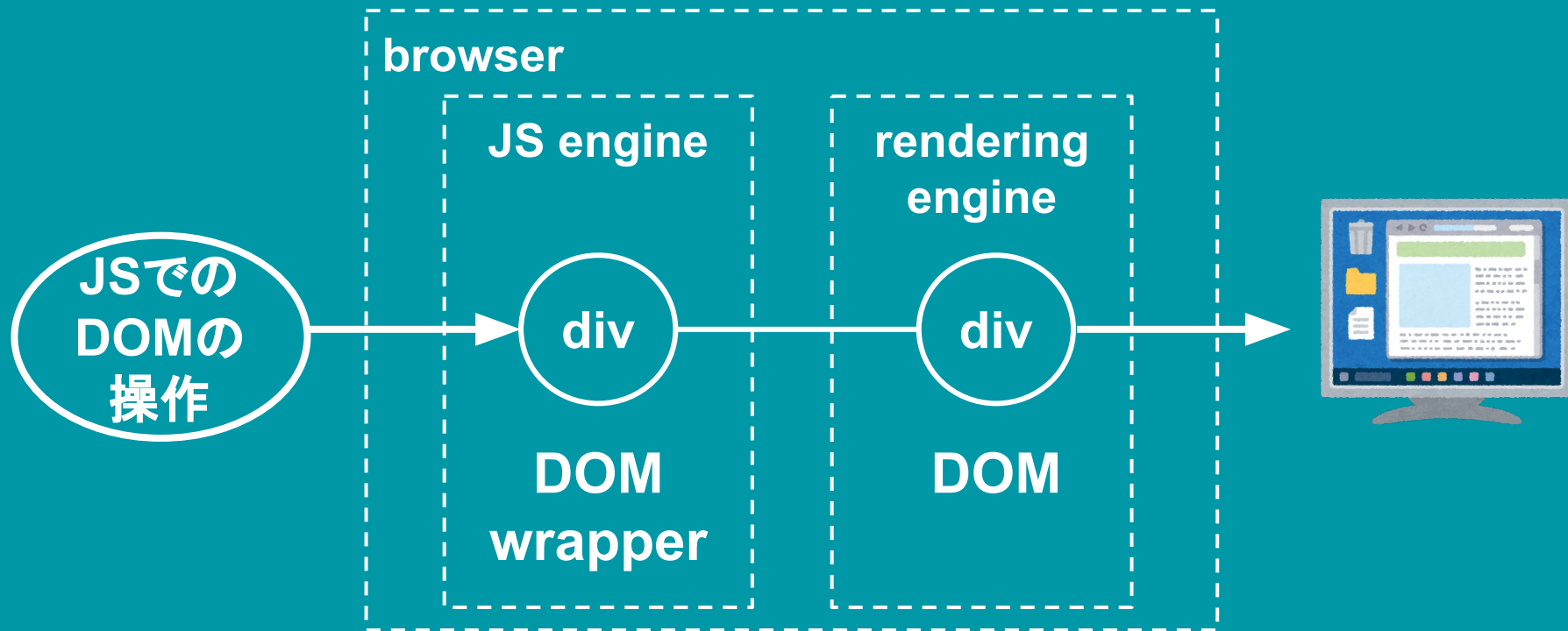


DOMは誰が持っているんだ？

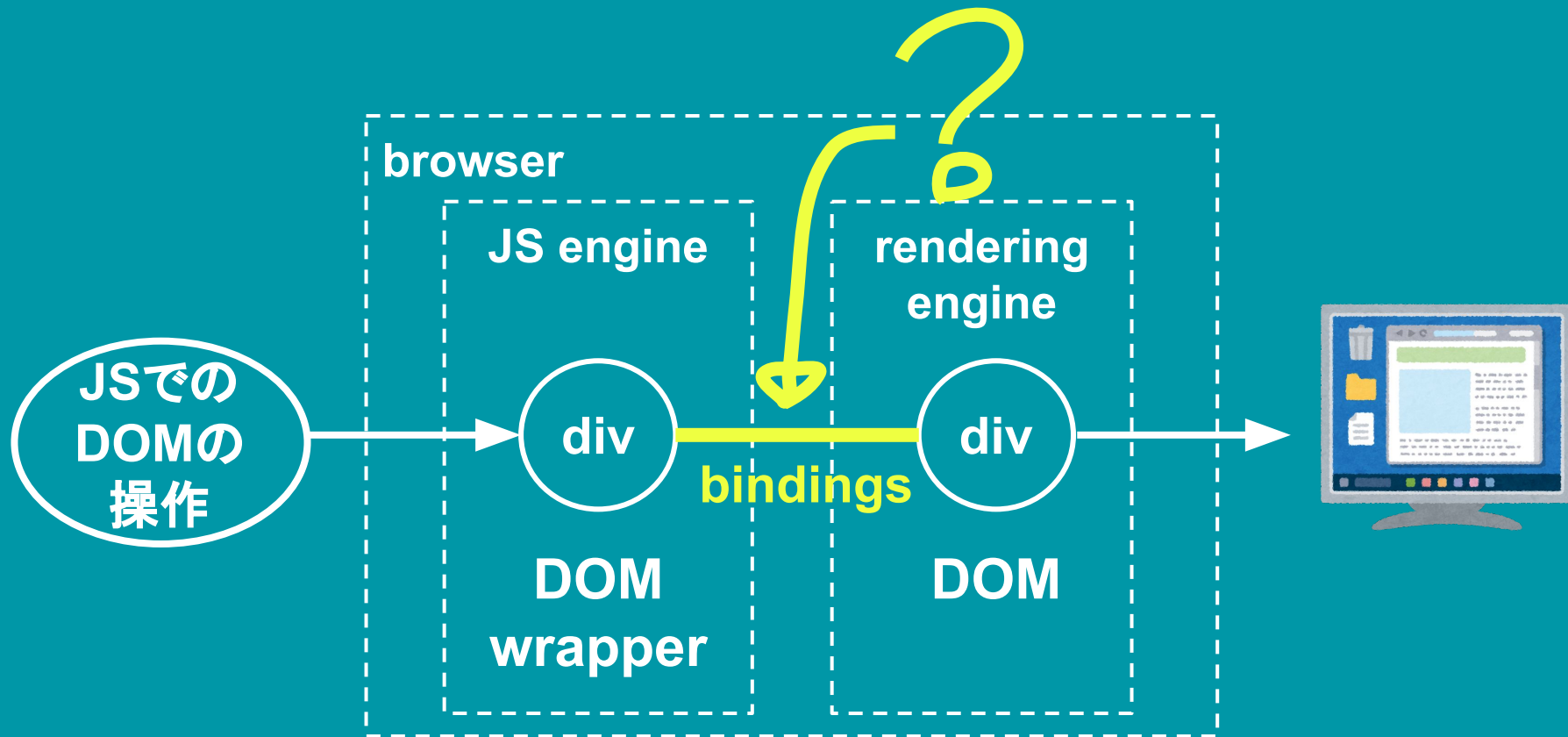


レンダリングエンジンにある？
でもJSエンジンにないとさわれない？

DOMの本体はレンダリングエンジン側にあり、ラッパーを通じてJSに露出



どうなっているんだ??



例えばdiv.firstChildをやると……

```
void FirstChildAttributeGetCallbackForMainWorld(  
    const v8::FunctionCallbackInfo<v8::Value>& info) {  
  
    /* 省略 */  
  
    v8::Isolate* isolate = info.GetIsolate();  
    v8::Local<v8::Object> v8_receiver = info.This();  
    Node* blink_receiver = V8Node::ToWrappableUnsafe(isolate, v8_receiver);  
  
    auto&& return_value = blink_receiver->firstChild();  
  
    bindings::V8SetReturnValue(info, return_value,  
                               bindings::V8ReturnValue::kMainWorld);  
}
```

結論

- JSで扱っているのはDOMの実体ではなくDOMラッパーである
- 各worldはDOMを共有するが、ラッパーは共有されない



DOMラッパーをsandboxする“world”

“A world is a concept to sandbox DOM wrappers among content scripts of Chrome extensions.

“All worlds in one isolate share underlying C++ DOM objects, but each world has its own DOM wrappers.

from “Design of V8 bindings”

worldは3種類あり、コンテキストを持っている

“ Also each world has its own context.

“ There are three kinds of worlds: a main world, an isolated world and a worker world.

A main world is a world where a normal JavaScript downloaded from the web is executed.

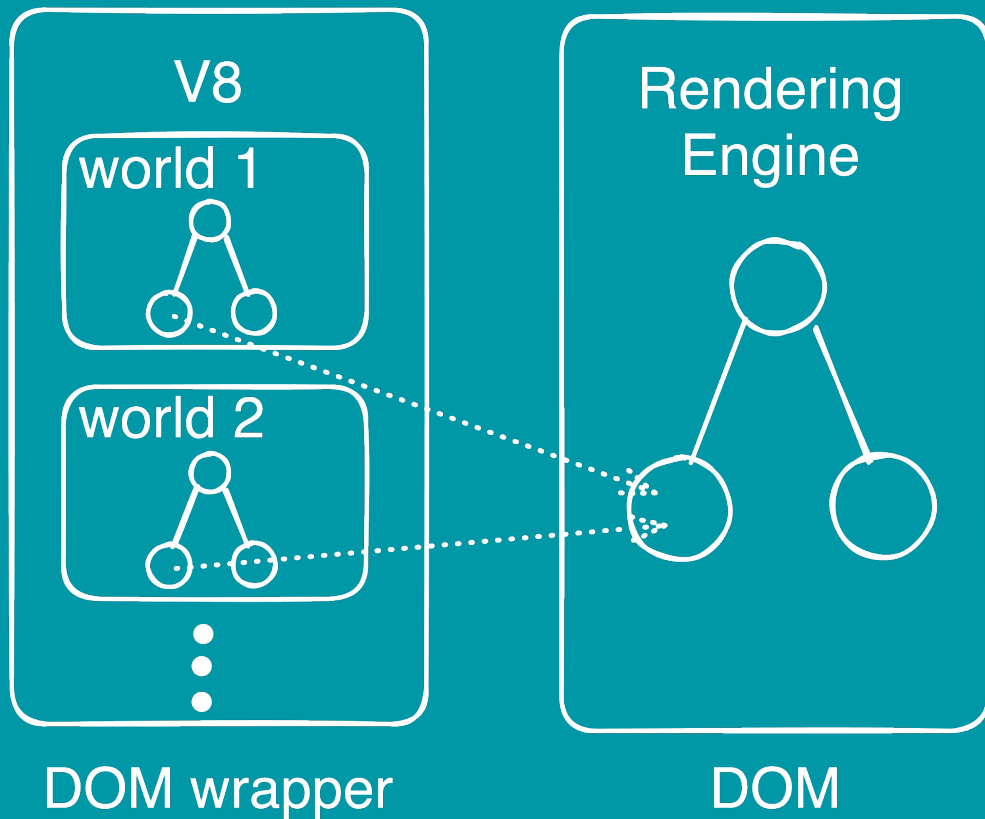
An isolated world is a world where a content script of a Chrome extension is executed.

from “Design of V8 bindings”

結論

- JSで扱っているのはDOMの実体ではなくDOMラッパーである
つまり.....
- 各worldはDOMを共有するが、ラッパーは共有されない

こういうこと



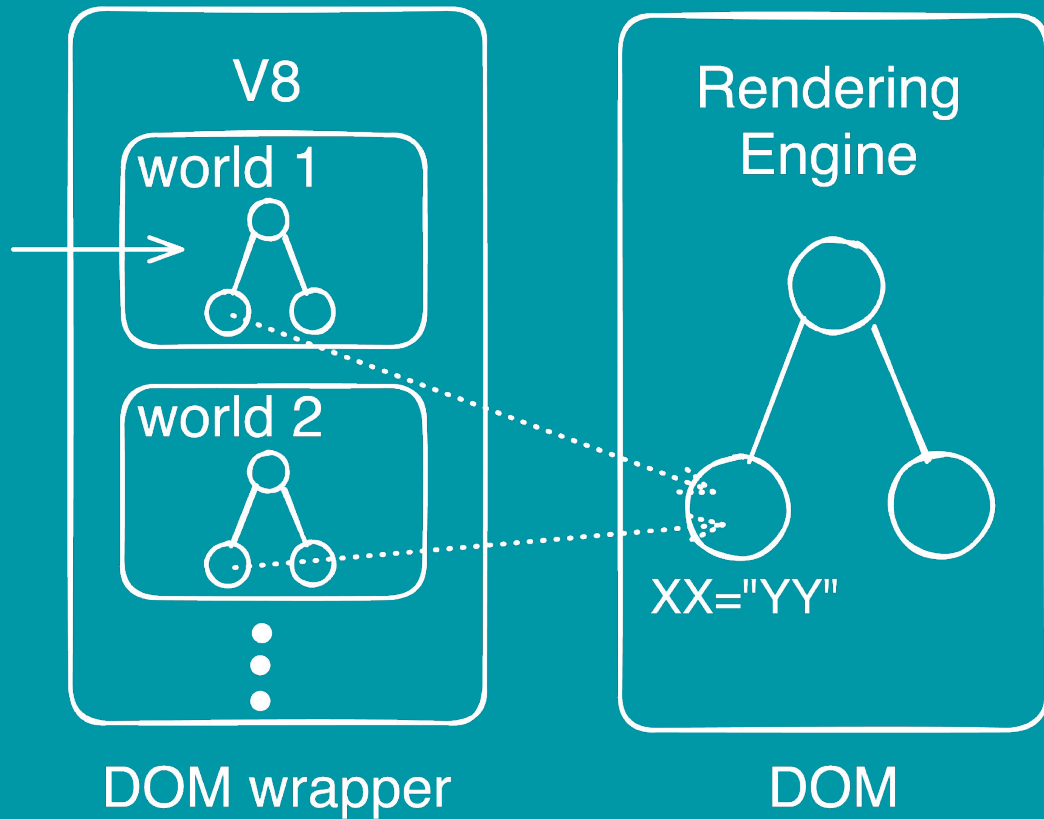
setAttribute()はDOMの操作であり……



div.setAttr(XX, YY)



div.getAttr(XX)
=> YY



div.XXはDOMラッパー、要するにJSオブジェクトの操作



div.XX = YY

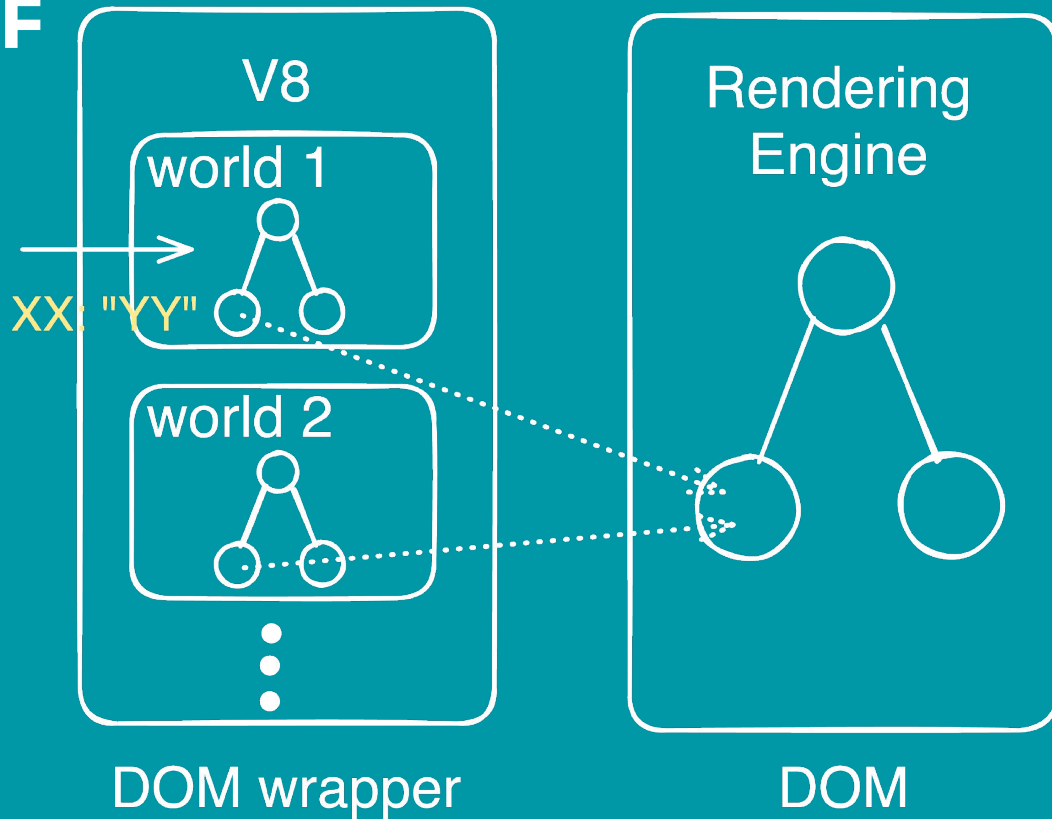


div.getAttribute(XX)

=> null

div.XX

=> undefined



まとめ

- JSで扱っているのはDOMの実体ではなくDOMラッパー(JS Object)である
- 各worldはDOMを共有するが、ラッパーは共有されない



本当はさあ……

GC

