alvin777 16 марта 2016 в 14:06

Стандарт ECMA-262 (JavaScript) в картинках, часть 1

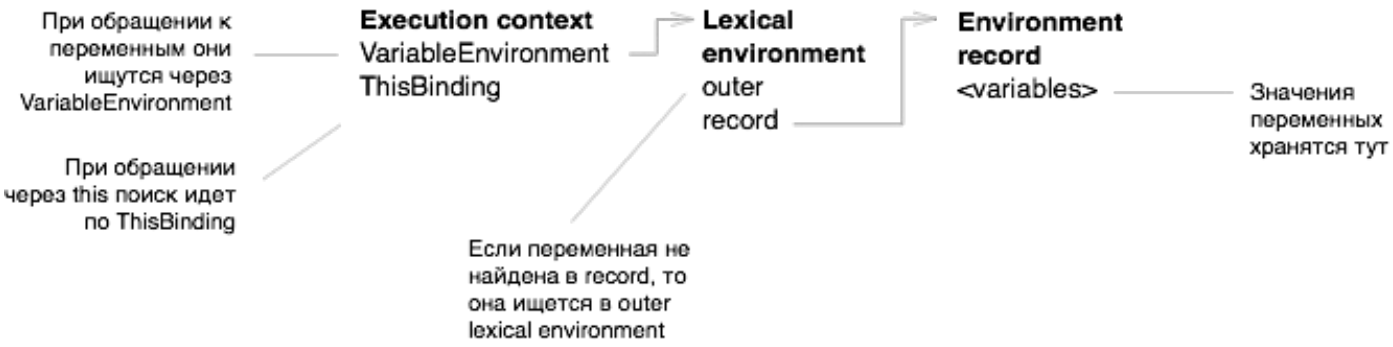
JavaScript

Из песочницы Tutorial

Про устройство JavaScript написано много статей. В первую очередь, это "JavaScript. Ядро." Дмитрия Сошникова, перевод статьи Ричарда Корнфорда и поста Дмитрия Франка. Но для того чтобы хорошо разобраться в какой-либо технологии лучше обратиться к первоисточникам. В данном случае к стандарту ECMA-262 ECMAScript Language Specification. Я рассматриваю этот пост как облегченный способ начать изучение стандарта. Рекомендую переходить по ссылкам, вчитываться в текст спецификации и составлять собственные схемы.

Как работают замыкания в JavaScript

Основные структуры ECMAScript это *execution context*, *lexical environment* и *environment record*. Они связаны следующим образом:



> Кроме *VariableEnvironment* в *execution context* есть еще *LexicalEnvironment*, подробнее про различия можно прочитать в ECMAScript 5 spec: LexicalEnvironment versus VariableEnvironment.
> *ThisBinding* будет рассмотрен в следующей части статьи

В *environment record* хранятся значения переменных. Если объявить `var a = 1`, то в текущем *record* появится {a: 1}. В *lexical environment* кроме *record* есть еще поле *outer*. *Outer* ссылается на внешний *lexical environment*, например, в случае вложенных функций.

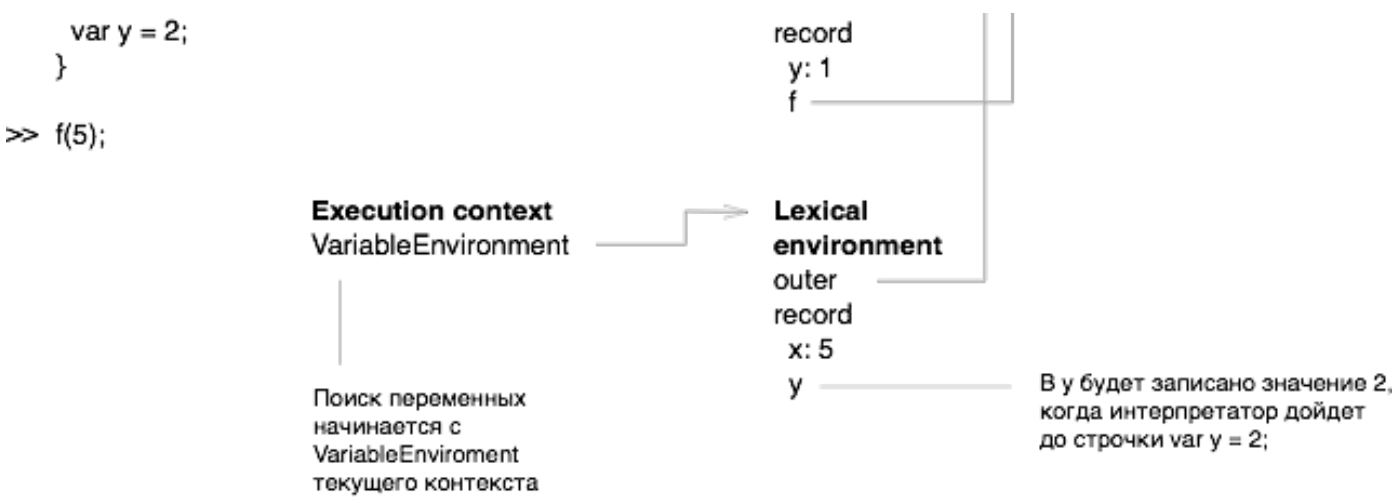
Поиск переменных начинается с *VariableEnvironment* текущего контекста. Если переменная с таким именем не найдена в *record*, то она ищется в *outer environment* по цепочке.

При запуске программы создаются глобальные *context* и *environment*. В качестве *record* используется *global object*.

Когда интерпретатор встречает ключевое слово *function* он создает *FunctionObject*. В свойство *scope* созданного *FunctionObject* записывается ссылка на текущий *lexical environment*.

Запись `function f(){}` практически эквивалентна записи `var f = function(){}` за исключением того, что в первом случае *FunctionObject* будет создан при входе в блок, содержащий `function f()`, а во втором при выполнении конкретной строчки.

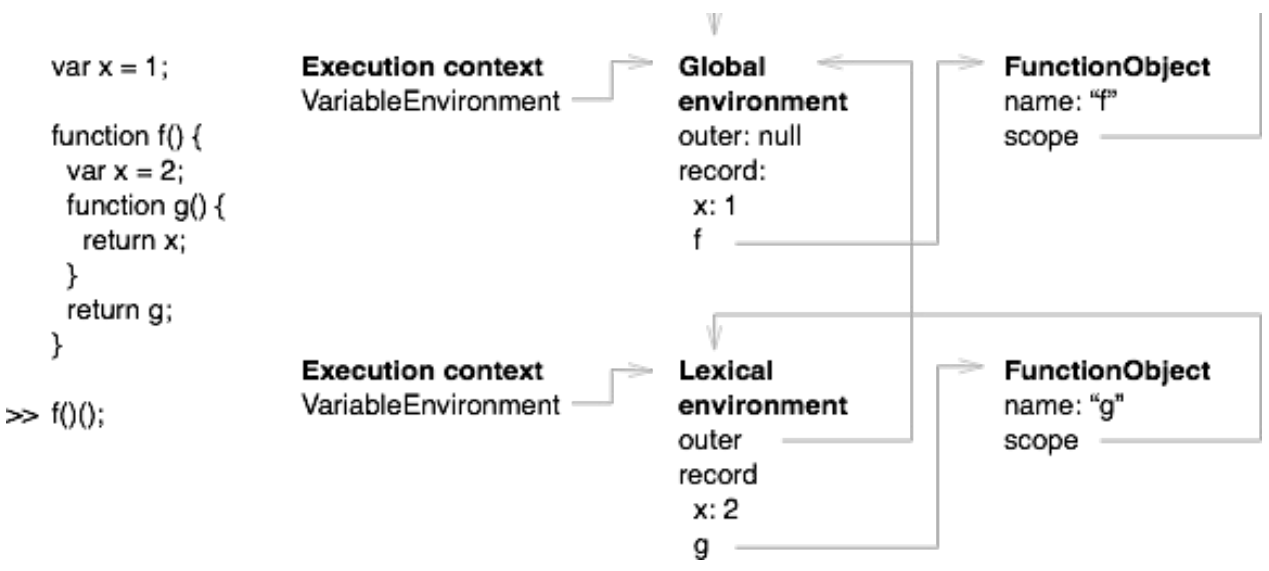
При каждом вызове функции создаются новые *context*, *environment* и *record*. Контекст кладется в стек, при выходе из функции он уничтожается. В *outer* созданного *environment* записывается *scope* вызываемого *FunctionObject*. Если функция была объявлена в глобальном контексте, то *outer* будет указывать на *global environment*.



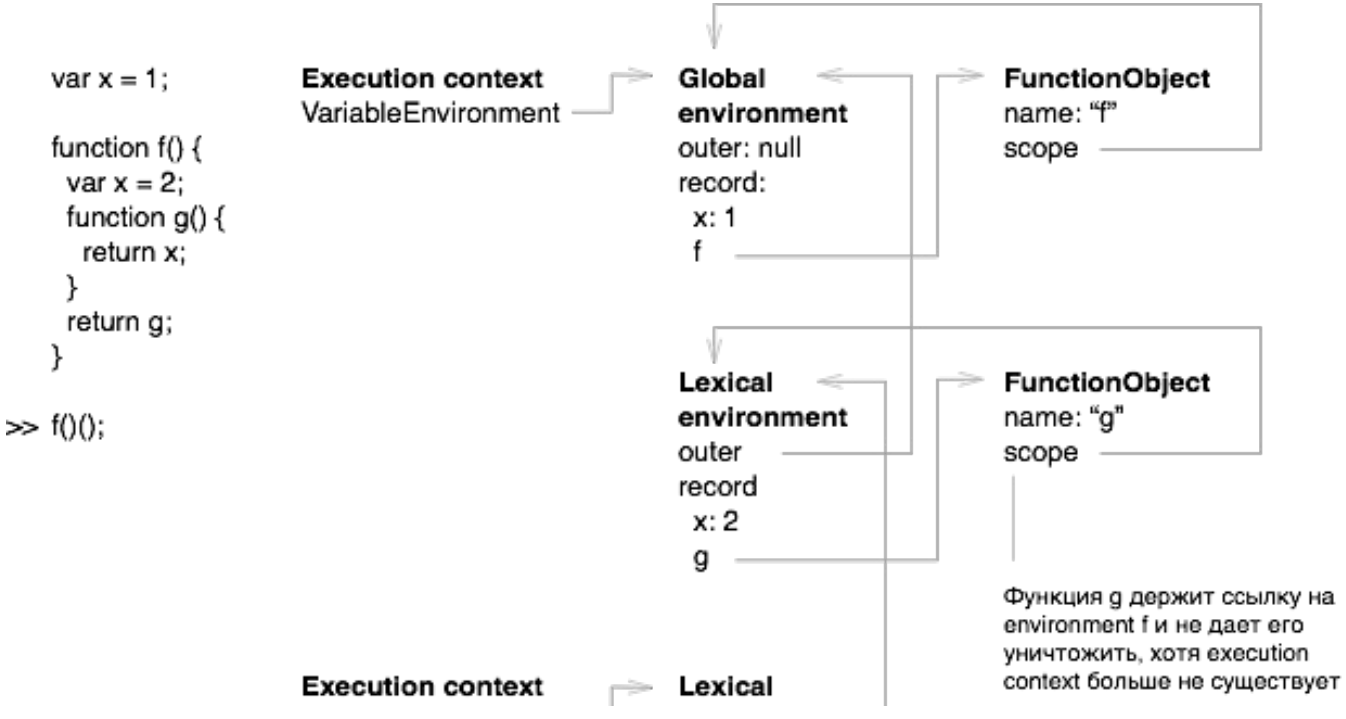
Теперь рассмотрим замыкание, когда одна функция возвращает другую.

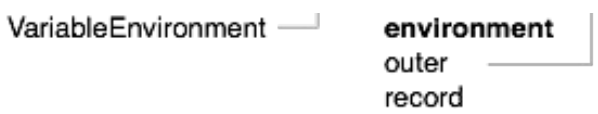
```
var x = 1;

function f() {
  var x = 2;
  function g() {
    return x;
  }
  return g;
}

f()();
```

При вызове функции *f* создается *context* и *environment* для *f*, а также *FunctionObject* для `function g`. В *scope* записывается ссылка на текущий *environment* из *VariableEnvironment*. При выходе из *f* контекст уничтожается, но *environment* останется, так как на него есть ссылка из возвращенного *FunctionObject*.



При вызове возвращенной из *f* функции *g* создается контекст и *environment* для *g*. В *outer* нового *environment* записывается *scope* из вызываемого *FunctionObject*. Поиск переменной *x* начинается с текущего *VariableEnvironment* и затем продолжается по *outer*. В результате будет возвращено значение `x = 2`.





В следующей части статьи разберем как с точки зрения ECMAScript работает *this*.

Теги: javascript, ecmascript, ecma-262

Хабы: JavaScript

7,0 Карма 0,0 Рейтинг

Краснояров Станислав @alvin777

Full-stack разработчик

ПОХОЖИЕ ПУБЛИКАЦИИ

25 марта 2016 в 10:11

Стандарт ECMA-262 (JavaScript) в картинках, часть 3

+6 12,8k 124 1

21 марта 2016 в 09:25

Стандарт ECMA-262 (JavaScript) в картинках, часть 2

+11 11,3k 81 1

31 декабря 2011 в 12:23

ISO 8601 и ECMAScript — головная боль от разночтения стандартов

+48 8,4k 27 37

КУРСЫ

Node.js: серверный JavaScript
17 августа 2020 • 6 недель • 27 000 ₽ • Loftschool

Комплексное обучение JavaScript
14 сентября 2020 • 7 недель • 27 000 ₽ • Loftschool

JavaScript/ DOM/ интерфейсы
4 августа 2020 • 2 месяца • 27 000 ₽ • Javascript.ru

JavaScript: новый уровень
15 сентября 2020 • 4 месяца • 786 $ • WAYUP

Профессиональный онлайн-курс JavaScript, уровень 1
15 сентября 2020 • 2 месяца • 21 900 ₽ • HTML Academy

Больше курсов на Хабр Карьере

Комментарии 1

vba 17 марта 2016 в 16:22 +2

Простите, где собственно картинки?

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

ЧТО ОБСУЖДАЮТ

Сейчас | Вчера | Неделя

Что случилось с транспортом и путешествиями за июль

1,8k 5

Советы руководителю от руководителя

38,9k 187

Какая асинхронность должна была бы быть в Python

3,1k 9

Twitter будет блокировать ссылки на экстремистский контент

1,6k 17

Опрос про big data в российских IT

Опрос

САМОЕ ЧИТАЕМОЕ

ПАРТНЕРСКИЕ МАТЕРИАЛЫ

Сутки | Неделя | Месяц

USB-флешки: заряжать нельзя игнорировать

+157 75,2k 215 141

Где прячется Российская электроника

+102 31,5k 61 145

Не надо портить инженерам десктопы своими мобильными решениями, одумайтесь

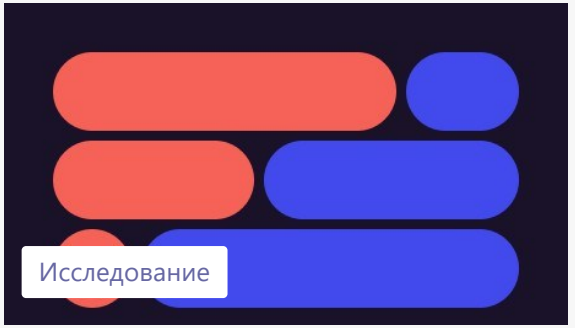
+206 45,3k 88 392

Коронавирус: первые итоги пандемии и карантина

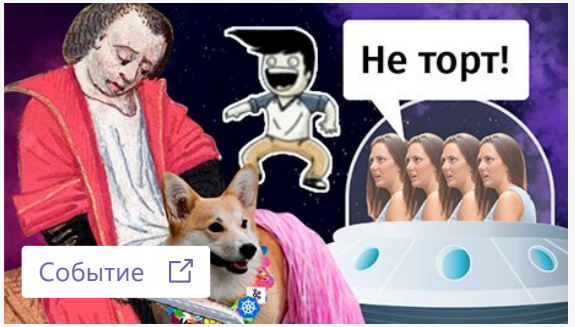
+34 14,8k 43 101

Вебкаст Habr Pro: IT vs HR — бой в 4 раунда

Интересно



Зарплаты айтишников в первой половине 2020



Улыбайтесь, господа, улыбайтесь: голосование за хабрамемы

Разместить