Именно фокусы, поскольку для меня это что-то новое.

Анонимные Самовызывающиеся функции

Как понятно из названия, такие функции не имеют названия и вызываются после описания.

Рассмотрим небольшой пример:

var greeting = (function(name) {

return 'Привет, ' + name;

})('Ivan');

alert(greeting);

Как вы можете видеть, мы обернули функцию в круглые скобки, после чего снова поставили круглые скобки в конце функции и передали туда нужный аргумент. Такая функция вызовется сразу же, как до нее дойдет интерпретатор, поэтому мы увидим "**Привет, Ivan**", хотя мы данную функцию не вызывали.

Это синтаксис JS. Кстати, в книге пока не нашёл. Чтобы вызвать функцию нужно имя, но поскольку функция без имени, то её же берём в круглые скобки и дальше передаём в неё аргументы.

(анонимная функция(аргументы функции){тело})(передаваемые агрументы)

Реальный пример:

(function($) {

…

Все наши скрипты jQuery

…

})(jQuery);

В анонимную функцию передаем объект **jQuery** и далее во всех наших функциях везде **$** - это **jQuery**.

Есть вики: <https://en.wikipedia.org/wiki/Immediately_invoked_function_expression>. Там описаны все синтакисы.

Последняя вервия JS это на данный момент ES8. Там всякие новшества и апгрейты, поэтому в книгах нет этого.

<https://tproger.ru/translations/es6-cheatsheet-1/> пояснения к новшествам.

Функции обратного вызова

Рассмотрим такой код

var func = function(callback) {

var name = 'Ivan';

callback(name);

}

Здесь мы передаем в функцию другую функцию, которую потом вызываем. Вот так теперь мы можем использовать наш код.

func(function(n) {

alert('Привет, ' + n);

});

Мы определили свою функцию, которая передается в параметре. Такой тип функций называется "функции обратного вызова" или callback по-английски.

В зависимости от того какую функцию передали внутрь, результат работы изменяется.