JavaScript

Функции

Об авторе: Сергей Мелюков
Круг интересов: Backend, Frontend, GameDev, MobileDev
Место работы: отдел внутренних разработок одной из крупнейших международных GameDev студий

LoftSchool
от мыслителя к создателю

План вебинара

- введение в функции
- типы объявления функций
- привязка аргументов
- замыкания и каррирование
- анонимные функции
- всплытие переменных и функций

Функции

Функции - это кусочки JS-кода, которые можно вызывать повторно.

У функции может быть имя, параметры и аргументы.

Имя функции - имя, по которому мы будем обращаться к этой функции в дальнейшем.

Параметры - именованные данные, которые могут быть переданы в функцию при ее вызове и использоваться внутри функции.

Аргументы - непосредственно данные, которые передаются в функцию при ее вызове.

Параметры функции

К значениям параметров(аргументам) можно обращаться как по имени, так и через специальный объект arguments.

arguments - служебная переменная, которая доступна внутри любой функции.

Содержит в себе значения переданных в функцию аргументов и пытается вести себя как массив.

Содержит свойство length, в котором содержится количество переданных в функцию аргументов.

К аргументу можно обратиться, указав его номер в квадратных скобках.

Типы объявления функций

Declaration - объявление функции в основном потоке кода.

При таком объявлении, интерпретатор сам создает переменную и помещает в нее код функции.

Expression - объявление функции внутри выражения.

В таком случае, необходимо самостоятельно создать переменную и записать в нее код функции.

Возврат значений

Любая функция может возвращать результат своей работы при помощи ключевого слова return

Результатом может быть всё, что угодно (это решает разработчик).

Используя возврат значений из функции, мы можем более гибко управлять результатом работы функци.

Привязка аргументов до вызова функции

В JS есть возможность привязать к функции любое количество аргументов до ее непосредственного вызова.

Это значит, что в момент вызова такой функции, привязанные к ней аргументы будут подставлены автоматически.

Для этого, у каждой функции в JS есть метод bind, который возвращает копию нашей функции, но с указанными привязанными аргументами.

Первым параметром необходимо передать контекст (более подробно здесь)

Далее, через запятую, перечисляются аргументы, которые мы хотим привязать к функции.

Замыкания

Замыкания - способность функции запоминать переменные, которые были доступны в той области видимости, в которой эта функция была объявлена.

При обращении к какой-либо переменной, сначала переменная ищется внутри функции, затем в области видимости, где функция была объявлена.

Частый пример использования замыканий - каррирование:

Если из функции А вернуть функцию Б, то функция Б запомнит аргументы переданные в функцию А и сможет их использовать.

Анонимные функции

Анонимные функции - функции без имени.

Используются обычно в тех случаях, когда имя функции не важно:

- возврат функции из функции
- передача небольшой функции в качестве параметра

Очень часто, анонимные функции вызывают сразу же после их создания:
`(function(a, b){return a+b;})(1, 2);`

Такой способ вызова функций называется: **Immediately Invoked Function Expression (IIFE)**

Всплытие

Всплытие (hoisting) - побочный эффект замыканий.

Мы можем использовать переменные из родительской области видимости вне зависимости от того - выше или ниже объявления функции эти переменные были объявлены.

Функции объявленные при помощи Declaration могут быть вызваны до того, как будут объявлены (в пределах одной области видимости).

Функции объявленные при помощи Expression ведут себя как переменные и не могут быть вызваны до своего объявления