

# Обработка ошибок

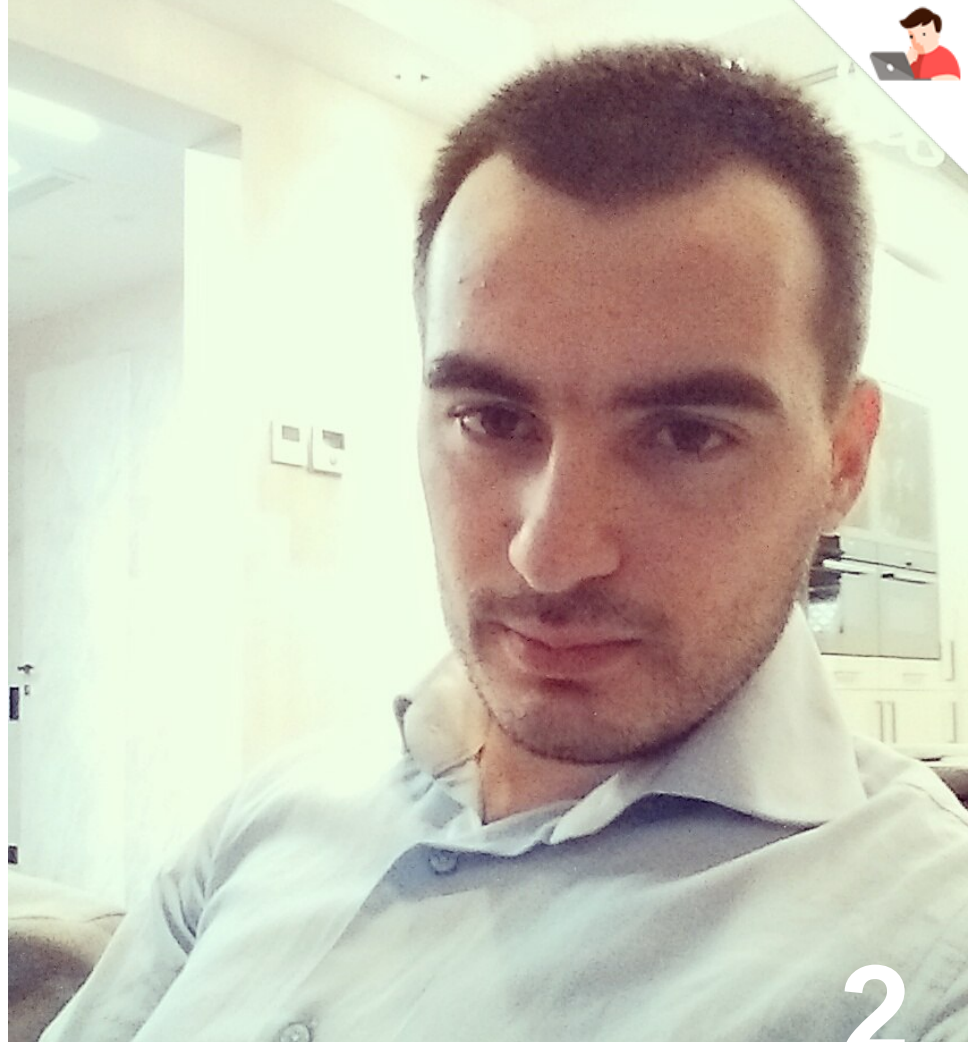
# Ведущий вебинара

---

**Сергей Мелюков**

**Круг интересов:** Backend, Frontend, GameDev, MobileDev

**Место работы:** Frontend разработчик профессиональных инструментов Avito



# Содержание

---



1. Инструменты разработчика (dev tools)
2. Работа с консолью
3. Обработка исключений
4. Точки останова (breakpoints)
5. Отладка (debug)



# Инструменты разработчика



# Инструменты разработчика

---

Типичные инструменты разработчика (dev tools):

- JS консоль
- сниффер http-запросов
- сканер ресурсов страницы (cookie, storage, изображения)
- сканер исходных текстов
- профайлер производительности

The background features a stylized illustration of mountains and a cloud. The mountains are composed of several overlapping triangles in shades of light gray and white, creating a layered effect. A single white cloud is positioned on the left side, partially overlapping the mountain peaks. The entire scene is set against a light gray background. In the top right corner, there is a yellow L-shaped line. In the bottom left corner, there is another yellow L-shaped line. At the bottom of the slide, there is a solid yellow horizontal bar.

# **Работа с консолью**

# Консоль

---



<b>log</b> ( <i>данные</i> )	сообщение в консоль
<b>info</b> ( <i>данные</i> )	информационное сообщение в консоль
<b>warn</b> ( <i>данные</i> )	предупреждение о чем-либо
<b>error</b> ( <i>данные</i> )	ошибка
<b>trace</b> ()	вывести стек вызовов, приведших текущей строке
<b>timer</b> ( <i>имя таймера</i> )	запустить таймер
<b>group</b> ( <i>имя группы</i> )	создать группу сообщение
<b>profile</b> ( <i>имя профайла</i> ) функции	запустить сбор статистике о вызовах

# Обработка исключений





# Обработка исключений

---

**Исключения** - механизм, позволяющий сообщать об ошибках во время выполнения программы, а так же обрабатывать такие ошибки.

**throw new Конструктор(описание)** - выбросить исключение типа Конструктор

В качестве аргумента, можно передать описание исключительной ситуации.

Исключения, после выброса, поднимаются вверх по текущему стеку вызовов.

На любом слое стека можно поймать исключение и обработать его при помощи оператора **try..catch..finally**

**Stack trace** - это путь, который прошел код, чтобы попасть

The background features a stylized illustration of mountains in shades of gray and white, with a single white cloud on the left side. There are also yellow L-shaped corner brackets in the top right and bottom left corners.

# Точки останова (breakpoints)



# Точка останова

---

**Точка останова (breakpoint)** позволяет моментально остановить выполнение кода и проанализировать состояние всех переменных, доступных на данный момент.

Может быть установлена при помощи вкладки Sources или при помощи служебного оператора языка JS - **debugger**.

Так же, при помощи вкладки Sources, можно по шагам пройти по всему исходному коду и отследить любые изменения переменных и вызовы всех функций. Такой процесс называется - отладка.

**Время ваших вопросов**