Лабораторная работа № 2

Тема: Контекст выполнения и ключевое слово this.

Цель: Закрепить знания о работе ключевого слова this в разных контекстах и научиться управлять им с помощью call, apply, bind.

🔷 Вариант 1

1. Глобальный контекст.

Написать скрипт, который выведет значение this в глобальной области (в браузере). Объяснить, почему результат отличается в обычном и строгом режиме ("use strict").

2. Метод объекта.

Создать объект user с полем name и методом sayName, который выводит имя. Проверить работу this.

3. Потеря контекста.

Сохранить метод объекта в отдельную переменную и вызвать его. Объяснить, почему теряется this.

4. Стрелочная функция.

Создать объект с методом, внутри которого есть стрелочная функция. Проверить, какое значение примет this.

5. Использование bind.

Исправить задание 3 с помощью bind, закрепив контекст объекта.

🔷 Вариант 2

1. Функция и strict mode.

Написать обычную функцию, которая выводит this. Запустить с "use strict" и без него. Сравнить результаты.

2. Метод объекта + вложенная функция.

В методе объекта вызвать обычную вложенную функцию. Что будет с this? Исправить поведение с помощью стрелочной функции.

3. Использование call.

Создать функцию greet, которая выводит приветствие с именем (this.name). Вызвать её с помощью call, передав объект с именем.

4. Использование apply.

То же самое, что и в задании 3, но аргументы передать массивом через apply.

5. Использование bind для обработчика.

Создать объект-счётчик с методом inc. Запустить его через setInterval так, чтобы счётчик правильно увеличивался (сначала не работает, потом починить через bind).

⚡ **Отчет должен содержать (см. образец):**
- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна VSC с исходным кодом программ;
- скриншоты с результатами выполнения программ;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email:
colledge20education23@gmail.com

Шпаргалка: this и управление контекстом в JS

🔷 Как работает this

Где вызвана функция	Чему равно this
Глобально (браузер, без strict)	window
Глобально (строгий режим)	undefined
Метод объекта	Сам объект
Обычная функция	window (без strict) / undefined (strict)
Стрелочная функция	Берёт this из внешнего контекста
Конструктор (new)	Новый объект

🔷 Примеры

Глобальный контекст

```
console.log(this); // window (в браузере)
"use strict";
console.log(this); // undefined
```

Метод объекта

```
const user = {
  name: "Анна",
  sayName() {
    console.log(this.name);
  }
};

user.sayName(); // "Анна"
```

Потеря контекста

```
const fn = user.sayName;
fn(); // undefined (или window.name)
```

Стрелочная функция

```
Стрелочная функция
const user = {
  name: "Анна",
  show() {
    const arrow = () => console.log(this.name);
```

```
    arrow();
  }
};

user.show(); // "Анна"
```

🔷 Явное управление контекстом

call — вызвать функцию сразу с this и аргументами

```
function greet(greeting) {
  console.log(greeting + ", " + this.name);
}

const user = { name: "Иван" };
greet.call(user, "Привет"); // Привет, Иван
```

apply — то же, но аргументы массивом

```
greet.apply(user, ["Здравствуйте"]);
// Здравствуйте, Иван
```

bind — закрепить this, вернуть новую функцию

```
const boundGreet = greet.bind(user);
boundGreet("Привет"); // Привет, Иван
```

Применение bind в колбэках

```
const counter = {
  count: 0,
  inc() {
    this.count++;
    console.log(this.count);
  }
};

setInterval(counter.inc, 1000);
// Ошибка: this потерян

setInterval(counter.inc.bind(counter), 1000);
// 1, 2, 3...
```