# Различия между var и let в JavaScript

## 1. var — функциональная область видимости

Переменные, объявленные с помощью var, имеют функциональную область видимости. Это означает, что переменная доступна внутри всей функции, в которой она объявлена, независимо от того, где именно внутри функции произошло объявление.

Пример:

function example() {  
 if (true) {  
 var x = 10; // Объявляем x внутри блока if  
 }  
 console.log(x); // x доступен здесь  
}  
  
example(); // Выведет: 10

## 2. let — блочная область видимости

Переменные, объявленные с помощью let, имеют блочную область видимости. Это означает, что переменная доступна только внутри блока {}, где она была объявлена.

Пример:

function example() {  
 if (true) {  
 let x = 10; // Объявляем x внутри блока if  
 console.log(x); // x доступен здесь  
 }  
 console.log(x); // Ошибка: x is not defined  
}  
  
example();

## 3. Основные различия

|  |  |  |
| --- | --- | --- |
| Особенность | var | let |
| Область видимости | Функциональная | Блочная |
| Поддержка блоков {} | Игнорирует блоки | Учитывает блоки |
| Всплытие (hoisting) | Поднимается и инициализируется как undefined | Поднимается, но доступна только после объявления |

## 4. Всплытие (Hoisting)

Переменные, объявленные с помощью var, поднимаются (hoisting) в начало своей функции и инициализируются как undefined. Переменные с let также поднимаются, но остаются в временной мёртвой зоне до момента, когда выполнение доходит до их объявления.

Пример с var:

function example() {  
 console.log(x); // undefined  
 var x = 10; // Переменная поднялась, но инициализация произошла позже  
 console.log(x); // 10  
}  
  
example();

Пример с let:

function example() {  
 console.log(x); // Ошибка: Cannot access 'x' before initialization  
 let x = 10;  
 console.log(x);  
}  
  
example();

## 5. Практическое значение

Использование let предотвращает случайные ошибки, связанные с доступом к переменным из других блоков или до их объявления. Рекомендуется всегда использовать let (или const, если значение не будет изменяться) вместо var для написания более безопасного и предсказуемого кода.