**Конспект «События в JavaScript». Раздел 2**

Области видимости

У каждой функции есть область видимости — все значения, доступные для этой функции.

Область видимости ограничена функцией, поэтому снаружи нельзя получить локальные переменные и параметры функции.

**Локальные переменные** — переменные, у которых область видимости ограничена функцией, где они объявлены. Такая область видимости называется локальной.

**Глобальные переменные** — переменные, которые объявлены на уровне всей программы, их видно из любого блока кода. Область видимости, в которой они объявлены, называется глобальной.

Если внутри функции обратиться не к локальной переменной, JavaScript будет искать переменную снаружи, переходя наверх от уровня к уровню, пока не найдёт переменную. Если переменной не будет ни внутри функции ни снаружи, будет ошибка.

Так как функция может использовать переменные, объявленные снаружи, их можно переопределять.

var food = 'макароны';

var eatDinner = function () {

console.log('Поел ' + food);

};

eatDinner();

// Выведет: Поел макароны

// Переопределяем переменную food

food = 'сельдерей';

eatDinner();

// Выведет: Поел сельдерей

Переопределять снаружи переменные, которые использует функция — не лучшая практика. Это может приводить к неожиданным последствиям и ошибкам в коде. Использовать это нужно осторожно.

Области видимости создаются только функциями. Поэтому, если переменная была создана в другой конструкции, например, в цикле, она будет доступна для чтения из функции.

Замыкания

**Замыкание** — функция, которая помнит о своём окружении. Это функция + все значения вне локальной области видимости, которые она использует.

Благодаря замыканиям мы можем зафиксировать какое-то значение в функции, а использовать саму функцию позже.

var collectContainer = function (food) {

return function () {

console.log('Поел ' + food);

};

};

var schoolkid = collectContainer('макароны');

schoolkid();

// Выведет: Поел макароны

Замыкания и асинхронность

Некоторые функции выполняются асинхронно, поэтому в момент выполнения кода значение переменной может уже измениться. Чтобы избавиться от этой проблемы, нужно создать отдельную область видимости. Так все переменные будут под контролем и замыкания не позволят потерять необходимые значения.

var thumbnails = document.querySelectorAll('.gallery\_\_photo-preview');

var fullPhoto = document.querySelector('.full-photo');

var addThumbnailClickHandler = function (thumbnail, photo) {

thumbnail.addEventListener('click', function () {

fullPhoto.src = photo;

});

};

for (var i = 0; i < thumbnails.length; i++) {

addThumbnailClickHandler(thumbnails[i], photos[i]);

}