

# Модуль - JavaScript

---

8/15 занятие

# Практика: Доработка плагина карусели

---

Задача сделать компонент как функцию

# Практика: Создание поповера

---

<https://www.figma.com/file/BRY70kY4IKf2kTUEhQ0ogH/F2023?type=design&node-id=881-1627&mode=design&t=QMUQaoUhyKJbfRP5-4>

Задание - создать ровер, который показывает дополнительный текст + название тега и наведении на лейбл

# Глобальная область видимости

---

Переменные объявленные вне функций и объектов попадают в глобальный объект `window` (`var` записываются как свойства, `let` только в лексическое окружение)

Из глобального пространства переменные и функции доступны в других объектах и функциях

# Замыкание

---

Это выделение области памяти(переменной) внутри функции

Такие области не удаляются сборщиком мусора(garbage collector) и существуют до перезагрузки страницы

Доступ к ним возможен только из внутренних функций - то есть в каком-то смысле это приватные переменные

```
let global = 1 // видна во всем скрипте
let fn = () => {
  let private = 0 // видна только в fn и дочерней ф-ции. не удаляется
  return () => {
    private++ // эта функция будет возвращаться со ссылкой на
    console.log(private) // на переменную private, то есть замкнется
  }
}
let newFn = fn() // при присвоении в newFn запишется только ф-ция
// из fn return
```

< JS >

# Модули в ES6

---

Реализуют принцип **Один скрипт – это один модуль**

Каждый модуль имеет свою собственную область видимости

Содержимое файла-модуля carousel.js:

```
export function createCarousel(elemId, data) {  
  ...  
}
```

< JS >

Подключение файла-модуля в main.js:

```
import {createCarousel} from './carousel.js'  
  
createCarousel('div.carousel', ['img1.png', 'img2.png'])
```

< JS >

**перерыв**

# Практика: Создание авторизации и валидации

---

<https://www.figma.com/file/BRY70kY4IKf2kTUEhQ0ogH/F2023?type=design&node-id=878-15855&mode=design&t=xKILpx58yDG0y71E-4>



**Конец занятия**