# Теоретическая часть

* Функции для работы с таймером – **setTimeout()** и **setInterval()**
* Управление таймерами – функции **clearTimeout()**, **clearInterval()**
* Объект **document.location** и его свойства
* Понятие о кросс-браузерном коде, полифилах
* Получение данных о геометрии окна браузера
* Получение данных о разрешении экрана устройства
* Открытие/закрытие дополнительных окон браузера
* Динамический HTML – манипулирование свойствами CSS элементов в коде JavaScript
* Задание классов CSS в коде JavaScript
* Получение стилей в коде JavaScript
* Примеры реализации простейшей анимации в коде JavaScript

# Практическая часть

Разработайте функции JavaScript для решения следующих. Скрипты должны размещаться на странице в стиле ненавязчивого JavaScript, выполнение обработки – по клику на кнопки (элементы ввода типа **button**, формы не использовать).

Выводите результаты работы скрипта в разметку, используйте стилизацию, flex, навигацию по страницам. Каждая задача должна размещаться на собственной странице, на главной странице разместите задание на разработку.

Вывод обрабатываемых данных реализуйте с выбором конкретного элемента разметки, без вызовов document.write().

**Задача 1**. Спроектировать класс в синтаксисе ES6+ для представления данных о погоде: температура, давление, влажность, скорость и направление ветра, графическое отображение атмосферных явлений (ясно, облачно, дождь, и т.д. – не более 5). Определите метод формирования строки для вывода данных в разметку. Переопределите метод **toString()** для простого вывода в консоль. Создайте массив данных о погоде за неделю, выведите его на страницу. По командам от кнопок выводите данные о погоде, упорядоченные (только при выводе, порядок элементов в исходном массиве не менять): по убыванию температуры, по возрастанию давления, по убыванию скорости ветра. По командам от кнопок выделяйте самые ветреные и самые тихие дни, дни с северным ветром. *Выделение должно сниматься через 10 секунд.* Предусмотрите кнопку для вывода исходного массива погодных явлений, без выделений элементов.

**Задача 2.**Спроектировать иерархию классов для предприятия ремонта бытовой техники. Создайте массивы объектов и выполните обработку по заданию. Базовый класс – персона (фамилия и инициалы, дата рождения, город проживания), производный от базового класс – клиент (дата обращения за услугой, название услуги, стоимость услуги), производный от базового класс – мастер (тарифная ставка, дата начала работы, признак занятости ремонтом). Переопределите методы **toString()**, **equals()** для клиента и мастера. Создайте массивы клиентов и мастеров (не менее 12 элементов), выводите эти массивы в разметку в порядке инициализации. По кликам по кнопкам упорядочивайте клиентов при выводе по алфавиту, мастеров – по тарифной ставке (порядок в исходном массиве не менять). Также предусмотрите кнопки для вывода массивов без упорядочивания*.* По кликам на кнопки выделяйте клиента/клиентов из города Иловайск, мастера/мастеров с максимальным тарифом. Через 10 с после выполнения команды выводите массивы в исходном порядке, без выделения.

**Задача 3.** Реализуйте «светофор» в выделенной части экрана: переключайте цвет области по циклу красный – желтый – зеленый, период переключения 3 с, предусмотрите кнопки разрешения и запрещения работы «светофора»

# Дополнительно

Запись занятия можно скачать по [**этой ссылке**](https://cloud.mail.ru/public/NT4U/xEtdrXjRF). Материалы занятия в этом же архиве.