11. Диаграммы

Практические задания.

Цель: научиться программно формировать данные для построения диаграмм на HTML и JavaScript.

№ 11.0: Подготовка.

1. Запустите терминальное окно. Обновите учебные материалы, введя команды:
   ```
   cd ~/CodeClub-IoT
   git pull
   ```
2. Для формирования тестовых данных запустите в терминальном окне сервер, имитирующий передачу значений датчиков температуры воздуха, относительной влажности и атмосферного давления модулем ESP8266, для чего выполните в терминальном окне команду:
   `~/CodeClub-IoT/samples/http_fake_esp.rb`
3. Проверьте работоспособность сервера ESPeasy, для чего запустите браузер и перейдите по адресу: `http://localhost:8000`

№ 11.1: Диаграммы в HTML-документе с применением JavaScript-библиотеки Chart.js

1. Откройте в браузере страницу с температурной диаграммой
   `~/CodeClub-IoT/samples/meteo_monitoring.html`
2. Откройте её исходный текст и изучите программу на JavaScript, которая применяется для формирования 3- хграфиков изменения температур.
3. Обратите внимание, что описание массива с данными о температурах `temp_sets` вынесено из программы в отдельные файлы, которые подключаются в тегах:
   ```
   <script src='chart_data1.js'></script>
   ```
4. `<script src='chart_data2.js'></script>`
5. Откройте эти файлы и посмотрите их содержимое.

№ 11.2: Программное формирование данных для диаграммы.

Нужно написать программу, которая будет запрашивать по HTTP данные о температуре, влажности и давлении от сервера, запущенного на выполнение в п. 11.0.2. Затем она будет накапливать данные в массиве и периодически формировать файл на JavaScript с данными для диаграммы изменения температуры.

1. Скопируйте из каталога `~/CodeClub-IoT/samples/` в свой каталог `~/projects/` файлы `meteo_monitoring.html`, `Chart.bundle.min.js`, `chart_data1.js`, `c art_data2.js`, которые нужно будет изменить.
2. Переименуйте документ `meteo_monitoring.html` в `meteo_chart.html` и измените его так, чтобы показывался только 1 диаграмма, а не 3:
3.
4. Откройте в браузере и проверьте, что всё продолжает работать.
5. Откройте программу `~/CodeClub-IoT/samples/http_json.rb` и сохраните её под именем `~/projects/meteo_chart.rb`.
6. Измените её так, чтобы она в бесконечном цикле 1 раз в 10 секунд считывала показания температуры и время сняти показаний и записывала их в массивы:
   ```
   # до цикла
   temperatures = []
   times  =[]
   # в цикле
   ```

```
temperatures << hash["Sensors"][0]["Temperature"]
times << Time.now.strftime("%H:%M:%S")
```

7. Заполненные массивы нужно записать в файл в формате JavaScript:
```
File.open("chart_data.js", "w") do |f|
    f.printf "temp_set=%s;\n", temperatures.to_s
    f.printf "label_set=%s;\n", times.to_s
  end
```
8. Измените документ **meteo_chart.html** так, чтобы данные для диаграммы брались из нужного файла, в котором формируются данные
```
<script src='chart_data.js'></script>
```
9. Измените имена в скрипте, чтобы данные брались из нужных массивов **temp_set** и **label_set**:
```
<script>
  // <!--
  temp_names = ["Температура воздуха, °C"];
  new Chart(document.getElementById("line-chart"), {
    type: 'line',
    data: {
      labels: label_set,
      datasets: [{.
          data: temp_set,
          label: temp_names[0],
          borderColor: "#FF0000",
          fill: false
        }
      ]
    },
    options: {
      title: {
        display: true,
        text: 'Температура'
      }
    }
  });
  // -->
  </script>
```
10. Проверьте, как всё работает, запустив программу формирования данных **meteo_chart.rb**.