**Дополнительный материал**

**Динамическое управление веб-страницей на клиентской стороне с использованием языка JavaScript:**

1. Язык JavaScript позволяет реагировать на действия, выполняемые пользователем на веб-странице, и на некоторые другие события. JavaScript имеет возможность изменить содержание, атрибуты и стилевое оформление любых элементов веб-страницы. Эти возможности используются для динамического изменения веб-страниц на клиентской стороне, в браузере, без обращения к серверу.
2. Наиболее удобным способом доступа к элементам страницы через скрипты является использование атрибута id элемента. Поэтому управляемым через JavaScript элементам обычно задается атрибут id.
3. Библиотека jQuery предоставляет упрощенный способ управления элементами, используя их id. Обращение к элементу в jQuery выглядит как *$('#id\_элемента')*.
4. Получить и изменить атрибут элемента можно, используя функцию *prop*. При вызове с одним аргументом, который передается как текстовая строка в кавычках, эта функция возвращает строку, содержащую значение указанного атрибута элемента. При вызове с двумя аргументами атрибуту, заданному первым аргументом, присваивается значение, заданное вторым аргументом. Например, следующим образом элементу изображения, атрибут которого **id="im1"**, можно заменить файл – источник картинки:

https://courses.openedu.ru/assets/courseware/v1/4b103f756b5d5aa96ed9c05428254de3/asset-v1:spbstu+WEBPYT+spring_2021+type@asset+block/Sam.05.01.gif

1. Получить или изменить стилевое свойство элемента можно, используя функцию *css*. При вызове с одним аргументом, который передается как текстовая строка в кавычках, эта функция возвращает строку, содержащую значение указанного стилевого свойства элемента. При вызове с двумя аргументами стилевому свойству, заданному первым аргументом, присваивается значение, заданное вторым аргументом. Например, следующим образом абзацу, имеющему id p1, можно поменять цвет фона на желтый:  
   $('#p1').css('background-color', 'yellow')
2. Получить или изменить HTML содержание контейнерного элемента можно, используя функцию *html*. При вызове без аргументов эта функция возвращает текст, содержащий разметку внутри элемента. При вызове с одним аргументом содержание контейнерного тега заменяется на переданный через аргумент текст. Например, следующим образом можно задать абзацу, имеющему id p1, новый текст:  
   $('#p1').html('<em>Задание выполнено!</em>')
3. Существует ряд стандартных событий, происходящих с веб-документом и обрабатываемых в JavaScript. Это клики по элементам, перемещение мыши, выбор элемента формы, изменение его текста и т.д.
4. Обработчики событий обычно записываются как функции JavaScript, определение которых начинается со слова function, за которым следует имя функции и передаваемые ей аргументы, задаваемые именами переменных в круглых скобках. Далее следует тело функции в блоке из фигурных скобок. Для назначения элементу обработчика события через jQuery следует вызвать для элемента функцию с именем события, передав ей в качестве аргумента имя функции-обработчика события. В следующем примере при клике по кнопке b1 (событие click) в абзац p1 выводится сообщение, что кнопка была нажата.

<p id="p1">

</p>

<button id="b1">

Нажми меня!

</button>

<script>

function b1\_click()

{

$('#p1').html('Кнопка b1 нажата!');

}

$('#b1').click(b1\_click);

</script>

9. Событие входа курсора мыши в область элемента – **mouseover**.

Событие выхода курсора мыши из области элемента – **mouseout**.

Событие выбора элемента для ввода – **focus**.

Событие снятия выбора с элемента – **blur**.

10. Для того, чтобы функции библиотеки jQuery были доступны в шаблонах страниц Django, следует перенести строку

<script src="{% static 'app/scripts/jquery-1.10.2.js' %}"></script>

из секции **body** файла **templates\app\layout.html** в конец секции **head** этого файла.

11. Для включения кода на JavaScript в состав шаблонов страниц следует добавить в шаблон страницы контейнерный тег **script** и разместить в этом контейнере функции-обработчики событий и код, задающий обработчики событий конкретным элементам страницы по их id.

12. Если элемент формы создаётся на основе класса формы, определенного в файле forms.py проекта Django, то ему в разметке страницы автоматически присваивается атрибут id со значением id\_имя, где "имя" - имя атрибута класса формы, задающего этот элемент.

**Задание для самостоятельной работы**

**Цель самостоятельного задания**: Получение начального навыка по использованию языка JavaScript.

1. Для формы, созданной при выполнении задания практического занятия 5, используя код JavaScript, добавленный в шаблон страницы с формой, реализуйте динамические эффекты изменения цвета надписи на кнопке отправки формы на сервер при наведении на нее курсора мыши и изменения цвета фона полей ввода при их выборе для ввода. Для изменения цвета текста и цвета фона задавайте элементам стилевые свойства **color** и **background-color**. При выходе курсора мыши из области элемента и выходе из поля ввода исходное состояние должно восстанавливаться.
2. Убедитесь, что визуальные эффекты, которые требовалось реализовать, наблюдаются при выполнении операций с формой на соответствующей странице сайта.
3. При возникновении ошибок, исправьте ошибки и повторите проверку наличия визуальных эффектов при работе с формой в соответствии с заданием для самостоятельной работы.