

Практическая работа №17

Разработка объектов на JavaScript

1 Цель работы

- 1.1 Научиться описывать прототипы объектов и создавать объекты на JavaScript;
- 1.2 Закрепить навык составления программ методами процедурного и объектно-ориентированного программирования.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18-19.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.15.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Разработать класс Cat (Кот) с параметрами (свойствами) name (имя), age (возраст), colour (окрас).
- 5.2 Создать объект класса Cat, задать имя, возраст и окрас.
- 5.3 Добавить метод meow для вывода «Мяу» на страницу. Добавить метод info для вывода информации о коте. Вызвать метод meow у созданного объекта.
- 5.4 Добавить возможность пользователю создать кота и задать его параметры. Вывести на страницу информацию о только что созданном объекте.
- 5.5 Создать класс tiger, наследник класса Cat. Переопределить функцию meow, создать объект класса и вызвать переопределенную функцию.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Как программно разместить элемент управления на странице?
- 8.2 Каков синтаксис функции создания объекта на JavaScript?
- 8.3 Как обратиться к свойствам (полям) объекта, используя JavaScript?

Практическая работа №18

Разработка веб-приложений с помощью библиотеки jQuery

1 Цель работы

1.1 Научиться разрабатывать веб-приложения и настраивать их внешний вид с помощью библиотеки jQuery.

1.2 Закрепить навык верстки веб-страниц и программирования на стороне клиента.

2 Литература

2.1 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.21.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Подключение и применение jQuery

Изучить, записать и проверить работу следующего примера:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h2>Заголовок страницы</h2>
  <p>Первый абзац</p>
  <p>Второй абзац</p>
  <button>Скрыть все абзацы</button>

  <script src="jquery-3.6.0.min.js "></script>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("p").hide();
      });
    });
  </script>
</body>
</html>
```

Вынести последний js-скрипт из html-страницы в отдельный файл.

5.2 Применение селекторов для скрытия элементов

Добавить на страницу из п.5.1 следующие кнопки:

- “Скрыть кнопку” (при нажатии должна скрываться кнопка, на которую нажали.

При реализации использовать селектор `this`);

- “Скрыть первый абзац” и “Скрыть второй абзац” (при нажатии должен скрываться соответствующий абзац). При реализации использовать селектор вида `#идентификаторЭлемента`);

- “Скрыть все элементы класса `test`” (при нажатии должны скрываться соответствующие объекты). При реализации указать у заголовка и второго абзаца класс `test`, использовать селектор вида `.классЭлемента`).

5.3 Применение селекторов для скрытия элементов

Добавить на страницу из п.5.1-5.2 следующие кнопки:

- “Показать абзацы” (при нажатии должны показываться оба абзаца. При реализации использовать метод `show()`).

- “Показать все элементы” (при нажатии должны показываться все элементы. При реализации использовать метод `show()` и селектор `*` для всех элементов тэга `body`).

5.4 Отображение и скрытие элементов на странице

Создать `html`-документ с абзацем и двумя кнопками “Скрыть быстро” и “Скрыть медленно”, подключить к нему `jQuery`. При реализации для отображения и скрытия абзаца вместо методов `hide` и `show` использовать метод `toggle()` в первой кнопке и `toggle(1000)` во второй кнопке.

6 Порядок выполнения работы

6.1 Используя `Notepad++` и браузер `Google Chrome`, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего предназначена библиотека `jQuery`?

8.2 Каков синтаксис обращения к различным типам элементов `html`-страницы с помощью `jQuery`?

8.3 Как подключить `jQuery`?

8.4 Как обработать действие пользователя с помощью `jQuery`?

9 Приложение

Базовая команда для библиотеки выглядит как:

`$(селектор).action(параметры, если требуются);` где

- `$` - предписание использовать jQuery;
- (селектор) - это запрос или элементы поиска в HTML элементах страницы;
- `action()` - это функция, которая вызывается при определенном действии или выполняет определенное действие над найденными элементами, которые удовлетворяют условиям селектора.

Селекторы:

- `window` - окно
- `this` - текущий элемент
- `тэг` - все элементы с указанным тэгом
- `.класс` - все элементы с указанным классом
- `#идентификатор` - элемент с указанным идентификатором

Селекторы могут быть скомбинированы (в этом случае они указываются через пробел)

В большинстве примеров jQuery-методы находятся внутри события документа `ready()`:

```
$(document).ready(function(){  
    // jQuery-методы размещаем здесь...  
});
```

Это необходимо для предотвращения любых срабатываний jQuery-кода до того, как документ полностью загрузится.

Это хорошая практика, чтобы дожидаться, пока документ будет полностью загружен и готов к работе с ним. Это также позволяет подключать свой JavaScript код в заголовочной части документа, а не в конце тела документа.

Синтаксис обработчика событий:

```
$(селектор).action(function(){  
    // описание действий в функции при наступлении указанного action  
});
```

Данные конкретного элемента можно получить следующими способами:

- атрибут элемента: `$("#идЭлемента").attr("название атрибута")` - значение указанного атрибута указанного элемента
- значение элемента: `$("#идЭлемента").val()`
- текст, находящийся между тэгами элемента: `$("#идЭлемента").text()`
- html-код, находящийся между тэгами элемента: `$("#идЭлемента").html()`
- значение css-свойства: `$("#идЭлемента").css("название свойства")`