

**ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам по дисциплине
«Интернет-программирование»**

ОГЛАВЛЕНИЕ

Содержание отчетов по лабораторным работам.....	3
Лабораторная работа №1. Статический html-документ	4
Лабораторная работа №2. Каскадные таблицы стилей CSS.....	15

Содержание отчетов по лабораторным работам

Отчеты по лабораторным работам оформляются в электронном виде с именами **Иванов, Петров (лр1).doc** (или *.docx, *.rtf, *.odt, *.pdf).

Отчеты оформляются по лаб. работам: №№1 и 2 (один отчет по двум лабораторным работам).

Лабораторная работа №1. Статический html-документ

Цель работы:

Изучить основы языка разметки гипертекста HTML 5.

Задание:

Создать html-документ, разметить его в соответствии с вариантом.

Для **ВСЕХ ВАРИАНТОВ** в разметке html-документа использовать:

- тег <DOCTYPE>
- тег <meta> для определения кодировки текста;
- тег <!-- --> комментария;
- теги: <p>,
, <div>, , <hr>, один из <h1> ÷ <h6>, , <i>, <u>, <sub>, <sup>, <pre>, <mark>, <details>, <summary>;
- продемонстрировать отличие в тегах <div> и ;
- теги <figure>, , <figcaption> для вставки изображения;
- тег <a> для разметки гиперссылок, разметить ссылки: **на другой документ, в пределах размечаемого документа, на email**;
- теги для разметки списка, таблицы, формы в соответствии с вариантом.

Вариант 1:

HTML 5

← → ↻ 🏠 🔍 ⋮

5. HTML
6. CSS
7. JavaScript

☐ GET
☒ POST

Выберите браузер:

Chrome
Firefox
Opera

Отправить

Вариант 2:

HTML 5

← → ↻ 🏠 🔍 ⋮

m. HTML
n. CSS
o. JavaScript

☐ HTML
☒ CSS
☒ JavaScript

Выберите цвет:

Отправить Сбросить

Вариант 3:

HTML 5

← → ↻ 🏠 🔍 ⋮

AA. HTML
AB. CSS
AC. JavaScript

☐ GET

☒ POST

Выберите дату:

дд . 02 . 2018

Отправить

Февраль 2018

◀ ● ▶

Пн	Вт	Ср	Чт	Пт	Сб	Вс
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4

Вариант 4:

HTML 5

← → ↻ 🏠 🔍 ⋮

c. HTML
ci. CSS
cii. JavaScript

☐ HTML

☒ CSS

☒ JavaScript

E-mail:

somebody@host.ru

Отправить

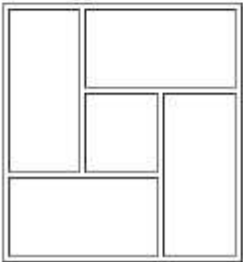
Сбросить

Вариант 5:

HTML 5

← → ↻ 🏠 🔍 ⋮

MM. HTML
MMI. CSS
MMII. JavaScript



☐ GET
☒ POST

Выберите четное число:

от 0 до 10

Отправить

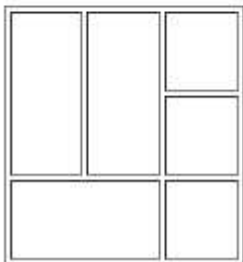
Сбросить

Вариант 6:

HTML 5

← → ↻ 🏠 🔍 ⋮

0. HTML
1. CSS
2. JavaScript



☒ HTML
☐ CSS
☒ JavaScript

Выберите число:

0 10

Отправить

Сбросить

Вариант 7:

с. HTML
ci. CSS
cii. JavaScript

☒ GET
☐ POST

Выберите время с 08:00 до 16:00:

12:00 x [dropdown]

Вариант 8:

XLIX. HTML
L. CSS
LI. JavaScript

☒ HTML
☒ CSS
☒ JavaScript

Укажите URL:

http://host.ru [dropdown]

Вариант 9:

HTML 5

← → ↻ 🏠 🔍 ⋮

Z. HTML
AA. CSS
AB. JavaScript

☒ GET
☐ POST

Выберите неделю:

Неделя --, ---- ▾

Отправить Сбросить

Вариант 10:

HTML 5

← → ↻ 🏠 🔍 ⚙ ⋮

aa. HTML
ab. CSS
ac. JavaScript

☐ HTML
☒ CSS
☐ JavaScript

Volvo ▾

Swedish Cars

Volvo

Saab

German Cars

Mercedes

Audi

Отправить Сбросить

Вариант 11:

HTML 5

c. HTML
ci. CSS
cii. JavaScript

☒ GET
☐ POST

Выберите число:

0 10

Вариант 12:

HTML 5

XLIX. HTML
L. CSS
LI. JavaScript

☒ GET
☐ POST

Volvo

- Swedish Cars
 - Volvo
 - Saab
- German Cars
 - Mercedes
 - Audi

Порядок выполнения лабораторной работы:

1. Запустить текстовый редактор.
2. Разметить html-документ в соответствии с заданным вариантом. Для справки о тегах и их атрибутах использовать справочники, расположенные по адресам <https://www.w3schools.com/html/default.asp> и <https://html5book.ru/html-html5>.
Стандарт HTML 5 расположен по адресу <https://www.w3.org/standards/techs/html>.
3. Проверить соответствие выполненной разметки стандарту HTML 5, используя валидатор WWW Консорциума, расположенный по адресу <http://validator.w3.org>.

Содержание отчета:

По двум лабораторным работам №№1 и 2 оформляется один отчет. См. раздел «Содержание отчета» для лабораторной работы №2.

Теоретические сведения

Язык разметки гипертекста HTML (Hypertext markup language) — язык разметки, используемый для создания гипертекстовых html-документов, отображаемых браузером.

Гипертекст — форматированный текст, содержащий ссылки на другие документы (гиперссылки).

Разметка — вставка в текст дополнительных служебных символов, каждый из которых является командой, указывающей браузеру, как следует отображать документ.

Язык разметки гипертекста HTML не является языком программирования.

Основным элементом языка разметки гипертекста HTML является тег (tag).

Теги содержат указания браузеру о способах отображения документа.

С помощью тегов в html-документ вставляются файлы, содержащие дополнительные данные (например, графику) и размечаются гиперссылки, посредством которых данный html-документ связывается с другими html-документами.

Как правило, теги состоят из начального и конечного элементов, между которыми размещаются текст и другие элементы html-документа. Имя конечного тега совпадает с именем начального, но перед именем конечного тега ставится косая черта.

Базовый синтаксис тега:

<name>

Содержимое тега

</name>

где <name> — это начальный элемент тега, содержащий имя тега, а </name> — конечный элемент тега.

В начальном элементе тега может располагаться перечень атрибутов тега. Атрибуты тега следуют за именем и отделяются друг от друга одним или несколькими пробелами. Порядок записи атрибутов в начальном элементе тега значения не имеет. Значение атрибута, если имеется, следует за знаком равенства, стоящим после имени атрибута. Если значение атрибута — одно слово или число, то его можно указать после знака равенства, не заключая в кавычки. Все остальные значения необходимо заключать в кавычки, особенно если они содержат пробелы. Если атрибут не указан, браузером используется его значение по умолчанию.

Регистр символов в именах тегов и атрибутов не учитывается.

<name attribute_1="value_1" attribute_2 ... attribute_n="value_n">

Содержимое тега

</name>

Конечные теги никогда не содержат атрибутов.

При использовании вложенных тегов их нужно закрывать, соблюдая правильную вложенность.

В некоторых случаях конечные теги можно опускать. Тем не менее, рекомендуется использовать конечные элементы тегов, чтобы избежать ошибок в отображении html-документа браузером.

Некоторые теги, не имеющие конечного элемента, называются автономными тегами.

<name>

<name attribute_1="value_1" attribute_2 ... attribute_n="value_n">

Html-документ состоит из заголовка документа и тела документа.

<html>

<head>

<title>Название html-документа</title>

Заголовок html-документа

</head>

<body>

Тело html-документа

```
</body>  
</html>
```

Весь html-документ заключается в тег `<html>`. Html-документ состоит из заголовка и тела, которые выделяются, соответственно, тегами `<head>` и `<body>`. В заголовке, с помощью тега `<title>`, указывается название html-документа, а также другие данные, которые браузер будет использовать при отображении документа.

Тело html-документа — та его часть, в которую помещается собственно содержимое html-документа. Тело включает предназначенный для отображения текст и управляющую разметку документа (теги), которые используются браузером.

Перечень тегов языка HTML и их атрибутов можно посмотреть в справочнике <http://htmlbook.ru>.

Лабораторная работа №2. Каскадные таблицы стилей CSS

Цель работы:

Изучить основы каскадных таблицы стилей CSS 3.

Задание:

Оформить html-документ, полученный в результате выполнения лабораторной работы №1 «Статический html-документ», добавив каскадные таблицы стилей. Изменить оформление:

- документа в целом (фон и т.п.);
- текста;
- гиперссылок;
- списка;
- таблицы.

Использовать определение стилей для тегов и классы стилей, псевдоклассы.

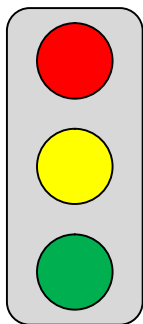
Использовать три способа определения каскадных таблиц стилей:

- с помощью тега <link>;
- с помощью тега <style>;
- с помощью параметра style тега.

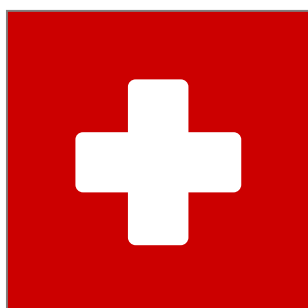
Продемонстрировать действие приоритетов при применении различных способов определения CSS;

Создать изображение в соответствии с вариантом, используя только свойства CSS.

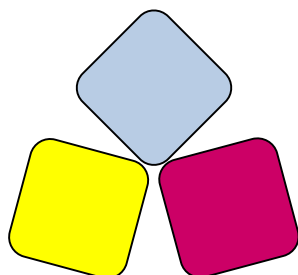
Вариант 1:



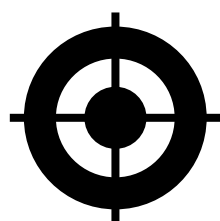
Вариант 2:



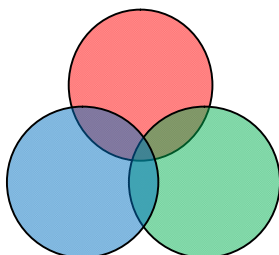
Вариант 3:



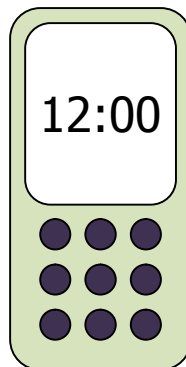
Вариант 4:



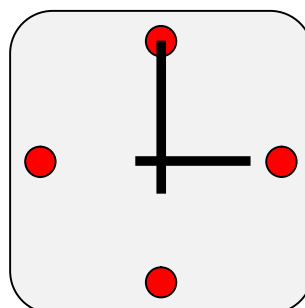
Вариант 5:



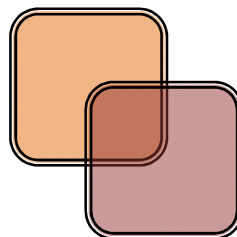
Вариант 6:



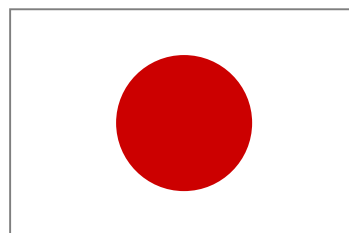
Вариант 7:



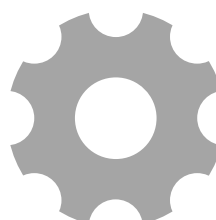
Вариант 8:



Вариант 9:



Вариант 10:



Порядок выполнения лабораторной работы:

1. Запустить текстовый редактор.
2. Оформить с помощью каскадных таблиц стилей html-документ, полученный в результате выполнения лабораторной работы №1, в соответствии с заданным вариантом. Для справки о свойствах CSS и их значениях использовать справочники, расположенные по адресам <https://www.w3schools.com/css/default.asp> и <http://htmlbook.ru/css>. Стандарты CSS 3 расположены по адресу <https://www.w3.org/Style/CSS>.
3. Протестировать оформленный документ в браузере.

Содержание отчета:

По двум лабораторным работам №1 и №2 оформляется один отчет.

- цель работы;
- задание;
- порядок выполнения лабораторной работы;
- html-разметка созданного документа, включая созданные стили;
- скриншот с результатами валидации html-документа;
- скриншот оформленного html-документа;
- выводы.

Теоретические сведения

Каскадные таблицы стилей CSS (Cascading style sheets) — формальный язык описания внешнего вида документа, созданного с использованием языка разметки гипертекста.

Каскадные таблицы стилей позволяют разделить описание логической структуры html-документа (выполненное с помощью языка разметки) и описание внешнего вида html-документа (выполненное с помощью CSS).

Существует три способа определения стилей: 1) в отдельном файле, подключаемом к html-документам, 2) с помощью тега `<style>` непосредственно в некотором html-документе и 3) с помощью атрибута `style` непосредственно в некотором теге.

Наиболее высокий приоритет имеет стиль, определенный в теге, затем следует определение стиля с помощью тега `style` и самым низким приоритетом обладают свойства, определенные в отдельном файле.

Каскад приоритетов особенно удобен при разработке больших проектов, например, сайтов, состоящих из большого числа html-документов. В этом случае общее оформление может быть вынесено в отдельный файл, в html-документе могут быть внесены изменения в стиль документа с помощью тега `<style>`, атрибут тега `style` позволяет изменить оформление одного тега.

Стили определяются парами свойств и значений, перечень пар заключается в фигурные скобки и пары разделяются точкой с запятой:

```
{property_1:value_1; property_2:value_2; ... ; property_n:value_n}
```

где `property` — это свойство, а `value` — значение свойства.

Стиль можно определить для конкретного тега, например, задать для тега `<body>` отображение белого текста на черном фоне:

```
body  
    {background-color:black; color:white}
```

Можно определить «чистый» стиль, не привязанный заранее к конкретному тегу, в этом случае речь идет об определении класса стиля:

```
.small_silver  
    {font-size:10px; color:silver}
```

или

```
#big_gold  
    {font-size:150px; color:#D7B56D}
```

Применение класса стиля:

```
<p class=small_silver>Текст светло-серого цвета размером 10 пиксел</p>  
или
```

```
<p id=big_gold>Текст светло-желтого цвета размером 150 пиксел</p>
```

Описание стилей для тегов или классов стилей выполняется одинаково как в отдельном файле, так и в теге <style>.

Файл со стилями должен иметь расширение *.css и быть подключен к html-документу с помощью тега <link>, расположенного в теге <head>.

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Тег <style> также должен быть расположен в теге <head>, после тега <link>.

Стили, определяемые непосредственно в теге с помощью атрибута style:

```
<p style="text-decoration-line:underline; color:rgb(255,0,0)">Подчеркнутый текст  
красного цвета</p>
```

Возможно задание различных стилей отображения одного и того же html-документа в различных средах представления, например, на экране или печати с помощью атрибута media тега <link>.

Файл screen.css

```
body  
    {color:silver; background:black}  
.forprint  
    {display:none}
```

Файл print.css

```
body
    {color:black; background:white}
.forscreen
    {display:none}
```

Файл media.html

```
<html>
<head>
<link href="screen.css" rel="stylesheet" type="text/css" media="screen">
<link href="print.css" rel="stylesheet" type="text/css" media="print">
</head>
<body>
<h1>Версия для <i class=forscreen>экрана</i><i class=forprint>печати</i></h1>
<div class=forscreen>Изображение будет видно на экране:
<p><img src=css3.jpg height=200px>
</div>
<div class=forprint><b>Внимание!</b> В версии для печати изображения
нет.</div>
</body>
</html>
```

