Задание подготовлено в рамках проекта АНО «Лаборатория модернизации образовательных ресурсов» «Кадровый и учебно-методический ресурс формирования общих компетенций обучающихся по программам СПО», который реализуется с использованием гранта Президента Российской Федерации на развитие гражданского общества, предоставленного Фондом президентских грантов.

**Разработчики**

Лысенко Ирина Владимировна, ГАПОУ «Тольяттинский индустриально-педагогический колледж»

Ковбасенко Антон Сергеевич, ГАПОУ «Тольяттинский индустриально-педагогический колледж»

**Назначение задания**

Оценка результата (продукта) деятельности. Уровень I

МДК.08.01 Проектирование и разработка интерфейсов пользователя

Тема: HTML. Оформление кода

**Комментарии**

Задание предлагается обучающемуся 2 курса, который изучил ОП Основы алгоритмизации и программирования и теоретический материал по основным командам HTML в МДК 08.01.

Ваша организация занимается разработкой сайтов. Дизайнеры сайтов написали три программы (HTML кода) для дальнейшего использования при создании сайтов. Вам поручили провести приемку.

Изучите программы на HTML (источник 1). Ознакомьтесь с правилами оформления HTML кода (источники 2и 3).

**Оцените соответствие программ требованиям.**

Заполните бланк.

*Бланк*

| № | Критерий оценки | Оценка, +/- |
| --- | --- | --- |
| Код программы 1 | Код программы 2 | Код программы 3 |
|  | Структура программы соответствует требованиям к структуре HTML |  |  |  |
|  | Код структурирован, форматирован |  |  |  |
|  | Подключены необходимые программные модули HTML |  |  |  |
|  | Присутствует атрибут языка |  |  |  |
|  | Стилевой файл подключён правильно |  |  |  |
|  | Логические блоки выделены правильно |  |  |  |
|  | Тэги оформлены верно |  |  |  |
|  | Тэги поставлены в нужных местах |  |  |  |
|  | Aтрибуты оформлены верно |  |  |  |
|  | Присутствуют комментарии |  |  |  |
|  | Комментарии информативны |  |  |  |
|  | Кодировка символов в html-документе указана верно |  |  |  |
|  | Web страница после запуска отобразится верно |  |  |  |
|  | Вёрстка идентично отобразится в разных браузерах  |  |  |  |
|  | Скрипты подключены правильно  |  |  |  |

***Источник 1***

**Программа 1. Вывести на Web-страницу таблицу.**

<!DOCTYPE html>

<html>

 <head>

 <meta charset="utf-8">

 <title>Моя первая таблица</title>

 </head>

<body>

 <table border=2px>

 <caption align=top>Заголовок таблицы</caption>

<thead>

 <tr>

 <th> Заголовок столбца 1</th>

 <th>Заголовок столбца 2</th>

 </tr>

</thead>

<tfoot>

<tr>

 <td>подвал столбца 1</td>

 <td>подвал столбца 2</td>

 </tr>

</tfoot>

<tbody>

 <tr>

 <td>Ячейка 1</td>

 <td>ячейка 2</td>

 </tr>

</tbody>

 </table

<script src="app.js"></script>

 </body>

</html>

**Программа 2. Вывести на Web-страницу текст с таблицей.**

<html>

<meta charset="utf-8">

<!DOCTYPE html>

 <html lang="ru">

<head> <title>Сайт</title></head><body><p>Привет школьники!<p><p>

 <script src="app.js"></script>

Программирование - большой мир, в который страшно войти со стороны.

В частности из-за этого программирование, как и любая профессия, давно обросло мифами и легендами.Рассказываем, чего точно не стоит бояться, и помогаем развеять популярные мифы.

</p><table> <tr><td>№ п/п</td><td>Наименование товара</td><td>Ед.

изм.</td><td>Количество</td><td>Цена за ед. изм., руб.</td><td>Стоимость, руб.</td></tr>

 <tr><td>1.</td><td>Огурцы свежие

</td>

<td>кг

г</td><td>15,20</td><td>69,00</td><td>1048,80</td></tr>

<tr>

<td>2.</td>

<td>Томаты свежие</td>

<td>кг</td>

<td>2,

50</td>

<td>4,00</td>

<td>120,00</td> </tr><tr><td align="right">ИТОГО:

</td>

<td>1168,80 </td>

</tr>

</table><p>Программисты проводят много времени за написанием и отладкой кода. Иногда в процессе разработки допускаются ошибки, которые впоследствии тяжело найти. Исправить эти ошибки помогают современные редакторы.</p>

</body>

</html>

**Программа 3. Вывести на Web-страницу текст с таблицей.**

<!--

<h213tmml>

 <meta charset=’666’>

<!-- Этот раздел предназначен для заголовка страницы и технической информации.

<Head>

<title>

</title>

</heafad>

<BODY>

<tablesas>

<captionwq>

Объединение ячеек по 2-ум направлениям

 </capt

 ion>

 <td>

<td>

Столбец 1

</th>

<td>

Столбец 2

</th>

<th>

Столбец 3

<th>

<t

h>

Столбец 4

</td>

 </td>

<tr

<th colspan=’3’>

Строка2 Ячейка1 (соединена с ячейками 2 и 3)

</td>

<td rowspan=’3’>

Строка2 Ячейка4 (соединена с ячейкой 4 в строках 3 и 4)

</th>

</tr>

<tf>

<qd colspan="2" rowspan="2"

Строка3 Ячейка1 (соединена с ячейкой 2 строки 3 и ячейками 1 и 2

строки 4)

</td>

<th>

Строка3 Ячейка3

</td>

 </tr>

<tr>

<td>

Строка4 Ячейка3

</tr>

</td>

</tableasd>

<qwe>

Программисты проводят много времени за написанием и отладкой кода.

Иногда в процессе разработки допускаются ошибки, которые впоследствии

тяжело найти. Исправить эти ошибки помогают современные редакторы.

</p>

 <Script src=’app.js’></Script>

</BODY>

</htmlspasse> }

***Источник 2***

**Правила оформления кода программы**

Обычно, когда мы работаем над конкретным программным продуктом, эстетические качества кода заботят нас далеко не в первую очередь.

Нам гораздо важнее наша производительность, качество реализации функционала, стабильность его работы, возможность модификации и расширения и т.д.

Но являются ли эстетические качества кода фактором, положительно влияющим на вышеперечисленные показатели? Ответ: да, и при этом, одним из самых важных!

Это так, потому что красивый код, вне зависимости от субъективной трактовки понятия о красоте, обладает следующими (в той или иной степени сводимыми друг к другу) важнейшими качествами:

1. Читаемость - т.е. возможность, глядя на код, быстро понять реализованный алгоритм, и оценить, как будет вести себя программа в том или ином частном случае.
2. Управляемость - т.е. возможность в минимальные сроки внести в код требуемые поправки, избежав при этом различных неприятных предсказуемых и непредсказуемых последствий.
3. Минимизация кода
4. Единообразие имен переменных, констант, подпрограмм, объектов и т.д., смысловая нагрузка имен (Mashina, Green\_Car) или наличие комментариев для переменных.

Программа, созданная на любом языке программирования, должна соответствовать требованиям не только к работоспособности и функционированию, но и к правильному форматированию (оформлению) кода программы.

Правильно оформленный код является хорошо читаемым, что немаловажно, когда над кодом работает несколько специалистов - программисты, тестировщики, аналитики и программисты во время апгрейдов программ и т.д.

Любую программу намного удобнее читать и понимать, когда в ней:

1. Присутствуют комментарии - каждый новый функциональный блок программы, важные программные конструкции, цели создания переменных должны сопровождаться комментариями. Комментарии не являются запускаемыми элементами программы и нужны для описания на обычном языке (русском, английском). Ответственное комментирование намного облегчает понимание кода и повышает возможность его использования.
2. Код структурирован для блоков программы различного уровня вложенности желательно делать сдвиги относительно левого поля. Это улучшает визуальное понимание основных и зависимых конструкций и облегчает чтение кода и логическое понимание выполняемых операций. Также для облегчения чтения программы важно разделять функциональные блоки кода пустыми строками.

Еще одним признаком хорошо структурированного кода является размещение всех операторов блока на одном уровне, в том числе операторных скобок

Также необходимо соблюсти удобство пользования программой. Для этого необходимо во время работы программы осуществлять удобный диалог с пользователем, выводить данные в хорошо отформатированном виде. Например, если вы вводите массив данных, выведите комментарий к пользователю.

***Источник 3***

# Требования к оформлению HTML-кода

## 1. Синтаксис

### 1.1 Два пробела в качестве отступа - структура кода

Отступы позволяют визуально оценить структуру документа и быстро переключаться между его фрагментами. Размер отступа настраивается в редакторе. Также во многих редакторах можно включить отображение пробельных символов и преобразовать отступы.

*Хорошо:*

<!DOCTYPE html>

<html lang="ru">

 <head>

 <meta charset="UTF-8">

 <title>Ваша страница</title>

 </head>

 <body>

 <h1>Страница</h1>

 <p>Текст</p>

 </body>

</html>

*Плохо*

<!DOCTYPE html>

<html lang='ru'>

<head>

<meta charset='UTF-8'>

 <title>Ваша страница</title>

</head>

<body>

<h1>Страница</h1>

<p>Текст</p>

</body>

</html>

### 1.2 Теги и атрибуты записываются в нижнем регистре

Символы нижнего регистра не привлекают к себе большого внимания, и вам легче будет найти нужный.

*Хорошо:*

<div class="user">

 <img src="./i/userpic.png" title="Иванов Иван">

 <a href="/profile/">Перейти в профиль</a>

</div>

*Плохо*

<DIV Class="user">

 <Img Src="./i/userpic.png" titLE="Иванов Иван">

 <A href="/profile/">Перейти в профиль</a>

</DIV>

### 1.3 Отдельные логические блоки отбиваются пустой строкой

Это облегчает работу с кодом и визуально создает структуру документа.

*Хорошо:*

<h1>Заголовок первого уровня</h1>

<p>Первый параграф</p>

<h2>Заголовок второго уровня</h2>

<ul>

 <li>Первый пункт</li>

 <li>Второй пункт</li>

 <li>Третий пункт</li>

</ul>

<h3>Заголовок третьего уровня</h3>

<figure>

 <figcaption>Блок с картинкой</figcaption>

 <img src="../img/image.jpg">

</figure>

*Плохо:*

<h1>Заголовок первого уровня</h1>

<p>Первый параграф</p>

<h2>Заголовок второго уровня</h2>

<ul>

 <li>Первый пункт</li>

 <li>Второй пункт</li>

 <li>Третий пункт</li>

</ul>

<h3>Заголовок третьего уровня</h3>

<figure>

 <figcaption>Блок с картинкой</figcaption>

 <img src="../img/image.jpg">

</figure>

### 1.4 Используйте только двойные кавычки

При написании значений атрибутов используйте только двойные кавычки.

*Хорошо:*

<div class="first">

 <img src="first-img.jpg" alt="Первая картинка">

</div>

*Плохо:*

<div class='first'>

 <img src='first-img.jpg' alt='Первая картинка'>

</div>

### 1.5 Не ставим пробелов перед и после символа =

Поскольку правая часть непосредственно относится к левой, то атрибут и его значение должны быть написаны без пробелов.

*Хорошо:*

<p id="first" class="post">Это хороший пример</p>

*Плохо:*

<p id ="first" class= "post">Это плохой пример</p>

### 1.6 Между атрибутами один пробел

Не используем переносов строк между атрибутами, только один пробел. Перенос строк принят в css-документах, но не в html-разметке. Пишите все атрибуты элемента в строку.

*Хорошо:*

<figure class="postItem">

 <img src="logo.jpg" alt="Логотип">

 <figcaption>Логотип компании</figcaption>

</figure>

*Плохо:*

<figure class="postItem">

 <img

 src="logo.jpg"

 alt="Логотип">

 <figcaption>Логотип компании</figcaption>

</figure>

### 1.7 Одиночные теги без закрывающего слэша

Не используйте закрывающий слэш у одиночных тегов (<img>, <br> и другие). Этот элемент является пережитком прошлого, когда HTML был более строгим, а браузеры не умели распознавать и исправлять ошибки.

*Хорошо:*

<img src="logo.jpg" alt="Логотип">

<br>

*Плохо:*

<img src="logo.jpg" alt="Логотип" />

<br />

## 2DOCTYPE

Первой строчкой в HTML-документе должен идти актуальный doctype. Это необходимо чтобы браузер верно отображал страницу. Это обеспечит единообразное отображение во всех современных браузерах.

<!DOCTYPE html>

<html lang="ru">

 <head>…</head>

 <body>…</body>

</html>

## 3. Кодировка

Кодировка символов в html-документе всегда должна быть указана явно. Это обеспечит верное отображение текста. Кодировка utf-8 подходит всегда, за редким исключением.

<head>

 <meta charset="utf-8">

 <title>Заголовок</title>

</head>

## 4. Подключение стилей

Файлы со стилями подключаются внутри тега <head> при помощи <link>. Атрибут type для тега <link> указывать не нужно, так как значение text/css устанавливается по умолчанию.

*Хорошо: стилевой файл подключён в секции head*

<!DOCTYPE html>

<html lang="ru">

 <head>

 <link rel="stylesheet" href="style.css">

 </head>

 <body>…</body>

</html>

*Плохо: стилевой файл подключён в секции body*

<!DOCTYPE html>

<html lang="ru">

 <head>…</head>

 <body>

 <link rel="stylesheet" href="style.css">

 </body>

</html>

## 5. Подключение скриптов

Скрипты при загрузке блокируют отображение содержимого страницы. По этой причине следует подключать их в самом конце html-документа.

При подключении скриптов в теге <script> атрибут type указывать не нужно, так как значение text/javascript устанавливается по умолчанию.

*Хорошо: скрипт подключается перед </body>*

<!DOCTYPE html>

<html lang="ru">

 <head>…</head>

 <body>

 <!-- Содержимое страницы -->

 <script src="app.js"></script>

 </body>

</html>

*Плохо: скрипт подключается в секции <head>*

<!DOCTYPE html>

<html lang="ru">

 <head>

 <script src="app.js"></script>

 </head>

 <body>…</body>

</html>

## 6. Атрибуты и их порядок

У HTML-элементов атрибут class пишется первым. Единообразное написание помогает легче считывать код и по классам быстрее разбираться в назначении блоков.

Остальные атрибуты могут быть расставлены в любом порядке, но тоже единообразно для схожих элементов.

<a class="element element-big" id="element" href="/" data-name="element">Ссылка</a>

<input class="form-control" type="text" name="test">

<img class="picture" src="img.jpg" alt="Картинка">

## 8. Подписи input

Для улучшения опыта пользователя, при нажатии на подпись поля, само поле должно активироваться. Для этого поле формы связывается с описанием при помощи идентификатора и атрибута for тега <label>.

*Хорошо: элемент формы radio связан с подписью через id и for*

<input type="radio" id="choose">

<label for="choose">Радио кнопка</label>

*Хорошо: элемент формы radio и подпись обёрнуты в label*

<label>

 <input type="radio"> Радио кнопка

</label>

*Плохо: подпись не связана с элементом формы*

<input type="radio" id="choose"> Радио кнопка

## 9. Атрибут языка

Для элемента <html> в атрибуте lang должен указываться верный язык документа. Это помогает системам перевода определить, какие использовать языковые правила.

<!DOCTYPE html>

<html lang="ru">

 <head>…</head>

 <body>…</body>

</html>

Инструмент проверки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Критерий оценки | Код программы 1 | Код программы 2 | Код программы 3 |
|  | Структура программы соответствует требованиям к структуре HTML | + | **+** | + |
|  | Код структурирован, форматирован | + | **-** | - |
|  | Подключены необходимые программные модули HTML | + | **+** | - |
|  | Присутствует атрибут языка | - | **+** | - |
|  | Стилевой файл подключён правильно | - | **+** | - |
|  | Логические блоки выделены правильно | + | **-** | - |
|  | Тэги оформлены верно | + | **+** | - |
|  | Тэги поставлены в нужных местах | + | **+** | - |
|  | Aтрибуты оформлены верно | + | **+** | - |
|  | Присутствуют комментарии | + | **-** | + |
|  | Комментарии информативны | - | **-** | + |
|  | Кодировка символов в html-документе указана верно | + | + | - |
|  | Web страница после запуска отобразится верно | - | **+** | + |
|  | Вёрстка идентично отобразится в разных браузерах  | - | **+** | + |
|  | Скрипты подключены правильно  | + | - | + |

*Подсчет баллов*

|  |  |
| --- | --- |
| За каждую полностью верно оцененный код | 3 балла |
| *За каждый код, оцененный с одной ошибкой или пропуском* | *1 балл* |
| ***Максимальный балл*** | ***9 баллов*** |