

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Рукович Александр Владимирович

Должность: Директор

Дата подписания: 30.05.2025 14:39:52

Уникальный программный ключ:

f45eb7c44954саас05ea7d4f32eb8d7d0b5cb9dae6c9b4bda094a0da0b703f

Министерство науки и высшего образования Российской Федерации
Технический институт (филиал) федерального государственного автономного
образовательного учреждения высшего образования «Северо-Восточный федеральный
университет имени М.К. Аммосова» в г. Нерюнгри

КАФЕДРА МАТЕМАТИКИ И ИНФОРМАТИКИ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Б1.В.07 Интернет-программирование

для программы бакалавриата

по направлению подготовки 09.03.03 «Прикладная информатика»

Направленность программы: Прикладная информатика в менеджменте

Форма обучения: заочная

УТВЕРЖДЕНО
на заседании выпускающей кафедры

«22» апреля 2020 г., протокол № 10

Заведующий кафедрой МиИ  В.М. Самохина
«22» апреля 2020 г.

СОГЛАСОВАНО:

Эксперты¹:

Самохина В.М., зав. кафедрой МиИ, ТИ (ф) СВФУ 
Ф.И.О., должность, организация, подпись

Похорукова М.Ю., доцент кафедры МиИ, ТИ (ф) СВФУ 
Ф.И.О., должность, организация, подпись

СОСТАВИТЕЛЬ:

Юданова В.В., старший преподаватель кафедры МиИ, ТИ (ф) СВФУ 
Ф.И.О., должность, организация, подпись



¹ Эксперт первый: со стороны выпускающей кафедры (или работодатель). Эксперт второй: со стороны обеспечивающей кафедры.

Паспорт фонда оценочных средств
Б1.В.ДВ.06.02 Интернет-программирование

№	Контролируемые разделы (темы)	Код контролируемой компетенции (или ее части)	Требования к уровню усвоения компетенции	Наименование оценочного средства
1.	Введение в web-технологии. Язык HTML. Язык таблиц каскадных стилей CSS.	УК-2.1: Выявляет и описывает проблему УК-2.2: Определяет цель и круг задач УК-2.3: Предлагает и обосновывает способы решения поставленных задач УК-2.4: Устанавливает и обосновывает ожидаемые результаты УК-2.5: Разрабатывает план на основе имеющихся ресурсов в рамках действующих правовых норм УК-2.6: Выполняет задачи в зоне своей ответственности в соответствии с	знать: концептуальные основы интернет, базовые протоколы передачи данных и служебные сервисы веб-сети; технологию создания гипертекстовых документов при помощи языков HTML и CSS; основы современных клиентских и серверных языков программирования, применяемых в разработке веб-приложений; базовые приемы проектирования веб-приложения с использованием СУБД.	лабораторные занятия, самостоятельные работы, аттестационная работа, курсовой проект, экзамен
2.	Клиентский язык программирования JavaScript	запланированными результатами и точками контроля, при необходимости корректирует способы решения задач УК-2.7: Представляет результаты проекта, предлагает возможности их использования и/или совершенствования ПК-1.1: Способен осуществлять выбор программно-технической архитектуры, средств и методов разработки программных продуктов, технических средств ПК-1.3: Способен вырабатывать и согласовывать требования к программному обеспечению с заинтересованными сторонами, оценивать и согласовывать сроки выполнения поставленных задач	УК-2.6: Выполняет задачи в зоне своей ответственности в соответствии с запланированными результатами и точками контроля, при необходимости корректирует способы решения задач УК-2.7: Представляет результаты проекта, предлагает возможности их использования и/или совершенствования ПК-1.1: Способен осуществлять выбор программно-технической архитектуры, средств и методов разработки программных продуктов, технических средств ПК-1.3: Способен вырабатывать и согласовывать требования к программному обеспечению с заинтересованными сторонами, оценивать и согласовывать сроки выполнения поставленных задач	
3.	PHP и MySQL	ПК-2.1: Способен применять методы и средства проектирования структур данных, баз данных, программных интерфейсов, программного обеспечения ПК-2.2: Способен приобретать новые и использовать существующие профессиональные знания в области типовых решений, библиотек программных модулей, используемых при разработке программного обеспечения ПК-2.3: Способен использовать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения ПК-3.1: Способен выполнять формализацию и алгоритмизацию поставленных задач в соответствии с требованиями технического задания ПК-3.2: Способен написать программный код с	уметь: разрабатывать веб-сайты, как совокупность гипертекстовых документов; разрабатывать веб-приложения, используя современные технологии и методы; обеспечивать проектирования веб-приложений исходя из требований	

		использованием языков программирования, использовать выбранную среду программирования и средства системы управления базами данных, стандартные библиотеки языка программирования ПК-3.3: Способен применять методы и приемы отладки программного кода, интерпретировать сообщения об ошибках, применять современные компиляторы, отладчики программного код	предметной области. владеть: навыками верстки веб-страниц; интегрированными средами разработки веб-приложений; типовыми приемами проектирования и разработки веб-приложений.	
--	--	---	---	--

Лабораторные работы

В период освоения дисциплины студенты посещают лекционные занятия, самостоятельно изучают дополнительный теоретический материал к лабораторным занятиям. Критериями оценки работы на лабораторных занятиях является: полнота и правильность выполненного задания; степень осознанности, понимания изученного; оформление задания.

Содержание отчета.

1. Титульный лист: название дисциплины; номер и наименование работы; фамилия, имя, отчество студента; дата выполнения.

2. Задание.

3. Листинг программы с выполнением задания.

4. Результаты работы программы.

5. Вывод по работе, сформулированные из цели.

Темы лабораторных работ

Тема 1. Введение в web-технологии. Язык HTML. Язык таблиц каскадных стилей CSS.

Задание 1. Выровняйте рисунки и текст так, как показано ниже:



Этот рисунок находится слева и текст выравнивается по левой границе. Эффект достигается без таблицы



Этот рисунок находится слева и текст выравнивается по центру. Эффект достигается без таблицы

Этот рисунок находится справа и текст выравнивается по правой границе. Эффект достигается без таблицы



Задание 2: создайте ссылку-картинку и ссылку на электронную почту.

Задание 3: создайте маркированные и нумерованные списки на странице.

Задание 4: Воссоздайте код представленной на рисунке веб-страницы. Используйте внутреннюю таблицу стилей, селекторы элементов p и span, css-свойства color и width.

Ширину абзаца установите в 300px. Цвета используйте red и blue.

Текст абзаца должен быть красного цвета.

А вот и нет, т.к. содержимое элемента 'span' должно иметь синий цвет.

Задание 5: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Используйте селекторы идентификаторов, классов и атрибутов.

Первая красная строка абзаца.

Синий 'span'.

Зеленый 'span'.

Задание 6: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Разрешается использовать в css-правилах только контекстные, дочерние и соседние селекторы.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link                                rel="stylesheet"
href="http://localhost/test/css/styles_1.css">
</head>
<body>
  <em>Оранжевый 'em'</em>
  <p>
    Первая красная строка абзаца. <br>
    <span> Синий 'span'.</span><br>
  </p>
</body>
</html>
```

Оранжевый 'em'

Первая красная строка абзаца.

Синий 'span'.

Задание 7: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Используйте селекторы псевдоклассов и псевдоэлементов :hover, :visited, ::first-letter. Для абзаца используйте обычный селектор элементов.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link rel="stylesheet" href="http://localhost/test/css/styles_1.css">
</head>
<body>
  <em title="Используйте псевдокласс">При наведении на меня
курсора
  я становлюсь оранжевым.</em>
  <p>
    Первая красная строка абзаца. <br>
```

```

<a href="http://localhost/test/index.html">
  Меня уже посетили.</a><br>
</p>
</body>
</html>

```

При наведении на меня курсора я ста Используйте псевдокласс

Первая красная строчка абзаца.

Меня уже посетили.

Задание 8: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Ширину (width) абзацев установите 500px, фон (background-color), соответственно, установите желтый и фиолетовый (violet), расстояние между словами, соответственно, 3px и 12px, расстояние между отдельными буквами – 1px и 2px, высоту строк – 1.1em и 1.6em, отступ первой строки – 1% и 3%. Во втором абзаце текст нужно выровнять по правому краю. Не забудьте перечеркнуть (подчеркнуть) соответствующие строки. Сравните результат форматирования обоих абзацев.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link
                                rel="stylesheet"
href="http://localhost/test/css/styles_1.css">
</head>
<body>
  <p class="p_1">
    <span id="p_2">
      Первое предложение абзаца.</span>
      Второе более длинное предложение абзаца.
      Третье и последнее предложение абзаца.
    </p>

  <p class="p_3">
    <span id="p_4">
      Первое предложение абзаца.</span>
      Второе более длинное предложение абзаца.
      Третье и последнее предложение абзаца.
    </p>
</body>
</html>

```

~~Первое предложение абзаца. Второе более длинное предложение абзаца. Третье и последнее предложение абзаца.~~

Первое предложение абзаца. Второе более длинное предложение абзаца. Третье и последнее предложение абзаца.

Задание 9: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой

нужно восстановить. Ширину (width) абзацев установите 500px, фон (background-color), соответственно, установите желтый и фиолетовый (violet). Для первого абзаца используйте шрифт Arial, а для второго – Times New Roman. Размер шрифта в обоих абзацах установите в 1em. Используйте, где потребуется, свойства font-variant, font-weight, font-style. Сравните результат форматирования текста с результатом форматирования в задаче 1.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link                                rel="stylesheet"
href="http://localhost/test/css/styles_1.css">
</head>
<body>
  <p class="p_1">
    <span id="p_2">
      Первое предложение абзаца.</span>
      Второе более длинное предложение абзаца.
      Третье и последнее предложение абзаца.
    </p>
    <p class="p_3">
      <span id="p_4">
        Первое предложение абзаца.</span>
        Второе более длинное предложение абзаца.
        Третье и последнее предложение абзаца.
      </p>
</body>
</html>
```

Первое предложение абзаца. Второе более длинное предложение абзаца. Третье и последнее предложение абзаца.

Первое предложение абзаца. Второе более длинное предложение абзаца. Третье и последнее предложение абзаца.

Задание 10: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Ширину (width) списков установите 500px, фон (background-color), соответственно, установите желтый и фиолетовый (violet). Используйте, где потребуется, свойства list-style-type, list-style-position. Сравните результат форматирования обоих списков.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link                                rel="stylesheet"
href="http://localhost/test/css/styles_1.css">
</head>
<body>
  <ul class="list_1">
    <li>
      Маркеры размещаем вне списка.
    </li>
    <li>
```



```

        Используем вид маркера decimal.
    </li>
</ul>
<ol class="list_2">
    <li>
        Маркеры размещаем вне списка.
    </li>
    <li>
        Используем вид маркера square.
    </li>
</ol>
</body>
</html>

```

1. Маркеры размещаем вне списка.
2. Используем вид маркера decimal.

- Маркеры размещаем вне списка.
- Используем вид маркера square.

Задание 11: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Используйте свойства caption-side, border-collapse border-spacing, empty-cells. Сравните результат форматирования обеих таблиц.

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Задачи по CSS</title>
    <link                                rel="stylesheet"
href="http://localhost/test/css/styles_1.css">
</head>
<body>
    <table class="table_1" border="1">
        <caption>Таблица №1</caption>
        <tr>
            <td>Ячейка 1.1</td><td>Ячейка 1.2</td>
        </tr>
        <tr>
            <td>Ячейка 2.1</td><td>Ячейка 2.2</td>
        </tr>
    </table> <br>

    <table class="table_2" border="1">
        <caption>Таблица №2</caption>
        <tr>
            <td>Ячейка 1.1</td><td>Ячейка 1.2</td>
        </tr>
        <tr>
            <td></td><td>Ячейка 2.2</td>
        </tr>
    </table>
</body>
</html>

```

Таблица №1	
Ячейка 1.1	Ячейка 1.2
Ячейка 2.1	Ячейка 2.2

Ячейка 1.1	Ячейка 1.2
	Ячейка 2.2

Таблица №2

Задание 12: Ниже представлен код html-страницы, а также внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Ширину первого абзаца установите 400px, второго – 300px. Высоту первого абзаца установите 10vh, второго – 200px. Во втором абзаце задайте минимальный предел ширины 400px, а максимальный предел высоты 100px. Используйте свойства width, height, min-width, max-height. Обратите внимание на конечные результаты ширины и величины второго абзаца из-за применения ограничений.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Задачи по CSS</title>
  <link                                rel="stylesheet"
href="http://localhost/test/css/styles\_1.css">
</head>
<body>
  <p class="p_1">
    Ширина абзаца равна 400px, высота &ndash; 10%
    от высоты области просмотра окна браузера.
  </p>

  <p class="p_2">
    Ширина абзаца будет 400px, т.к. width меньше min-
width,
    а высота &ndash; 100px, т.к height превышает max-
height.
  </p>
</body>
</html>

```

Ширина абзаца равна 400px, высота – 10% от высоты области просмотра окна браузера.

Ширина абзаца будет 400px, т.к. width меньше min-width, а высота – 100px, т.к height превышает max-height.

Задание 13: В условии задачи 1 представлен код html-страницы, а ниже показан внешний вид страницы при отображении ее браузером. К странице подключена внешняя

таблица стилей, код которой нужно восстановить. Воспользуйтесь результатом задачи 1, добавив в таблицу стилей дополнительные свойства, устанавливающие соответствующие границы абзацев: `border`, `border-top`, `border-bottom`. Ширину всех границ установите 4px.

Ширина абзаца равна 400px, высота – 10% от высоты области просмотра окна браузера.

Ширина абзаца будет 400px, т.к. `width` меньше `min-width`, а высота – 100px, т.к. `height` превышает `max-height`.

Задание 14: В условии задачи 1 представлен код html-страницы, а ниже показан внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Воспользуйтесь результатом задач 1 и 2, добавив в таблицу стилей дополнительные свойства, устанавливающие скругление соответствующих углов первого абзаца и внешние границы второго абзаца: `border-radius`, `border-bottom-right-radius`, `border-bottom-left-radius`, `outline`, `outline-offset`. Скругление границ установите 1em, а для нижних углов переопределите на 2em. Расстояние между внутренней и внешней границами установите 2px. Ширину внешней границы установите 4px.

Ширина абзаца равна 400px, высота – 10% от высоты области просмотра окна браузера.

Ширина абзаца будет 400px, т.к. `width` меньше `min-width`, а высота – 100px, т.к. `height` превышает `max-height`.

Задание 15:. В условии задачи 1 представлен код html-страницы, а ниже показан внешний вид страницы при отображении ее браузером. К странице подключена внешняя таблица стилей, код которой нужно восстановить. Воспользуйтесь результатом задач 1, 2 и 3, добавив в таблицу стилей дополнительные свойства, устанавливающие внутренние и внешние отступы абзацев: `padding`, `padding-left`, `margin`, `margin-top`. Внутренние поля в обоих абзацах установите 0.5em, а затем их внутренние левые отступы переопределите на 0.8em. Внешние отступы установите 30px, но во втором абзаце внешний отступ сверху переопределите на 50px. Сравните результаты всех четырех задач.

Ширина абзаца равна 400px, высота – 10% от высоты области просмотра окна браузера.

Ширина абзаца будет 400px, т.к. width меньше min-width, а высота – 100px, т.к. height превышает max-height.

Задание 16: создать форму для обратной связи и придумать собственный красивый дизайн.

Задание 17: Используя атрибуты объединения ячеек, изменения цвета фона и выравнивания текста создайте таблицу в HTML с красивым дизайном.

Задание 18: Создать собственный сайт, состоящий минимум из трех страниц. На сайте должны быть следующие элементы: меню, списки, заголовки, текст, картинки, ссылки, форма обратной связи или регистрации, таблицы. Обязательно использование каскадных таблиц стилей.

Тема 2. PHP и MySQL. JavaScript и jQuery.

Задание 1. Вывести в окно браузера следующие данные: Ваше ФИО, возраст, хобби (каждое на новой строке)

Задание 2. Написать сценарий (javascript) для вывода двух строк текста, красной и синей.

Красная строка
Синяя строка

Задание 3. Найдите и исправьте ошибки во фрагментах кода:
`document.whrit("Проблемы?");`

`alert(Hello);`

Задание 4. Исправьте ошибки во фрагменте кода:

`alert("Hello World!");` / это однострочный комментарий

Задание 5. Объявите две переменные разными способами и присвойте им значения. Выведите на экран значения переменных при помощи метода `alert()`.

Задание 6. Исправьте ошибки в правильности объявления локальных переменных во фрагменте кода:

`String s = "String";`

`int a = 10;`

`long b = 25;`

Задание 7. Выполните задание по шагам:

• Создайте 3 переменные с использованием ключевого слова `var` с идентификаторами: a, b, c.

• Переменной a присвойте значение `false`.

• Переменной b присвойте значение `null`.

• Переменная c должна принимать значение `undefined`.

• Отобразите значение 3-х переменных последовательно в модальных окнах (то есть с помощью метода alert()).

Задание 8. Что должно быть в ответе на следующие присваивания?
"1"+2+3=
1+2+"3"= ?

Задание 9. Вывести в окно браузера при помощи метода alert() следующие данные: Ваше ФИО, возраст, хобби (каждое на новой строке: использовать специальные символы).

Задание 10. С помощью javascript метода document.write() вывести в окно браузера строку: *Кто ты такой?* (с пробелами между словами).

Последовательно выполните:

1. Создать 4 переменные с использованием ключевого слова var с именами str1, str2, str3, concatenation.

2. Переменной str1 присвоить фразу 'Кто ', str2 – 'ты ', str3 – 'такой?'

3. Локальной переменной concatenation присвоить результат конкатенации 3-х строк: str1, str2, str3.

4. Вывести в документ содержимое переменной concatenation.

+x // к числу

x+" // к строке

!!x // к булеву типу

Задание 11. Исправьте ошибки при объявлении локальных переменных во фрагментах кода:

```
String s = "String";  
int a = 10;  
long b = 25;
```

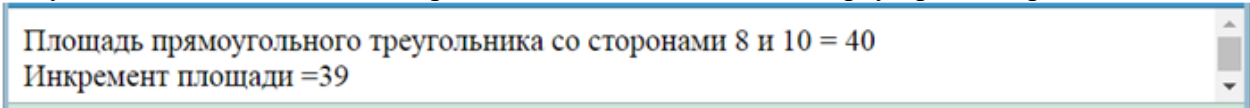
```
var name = "Меня зовут Вася ";  
var 2b = 10;  
Var _@c = 15;  
alert(Name);
```

Задание 12. Какие значения выведет в окно браузера следующий фрагмент кода?

```
var str = "20";  
var a = 5;  
document.write(str + a + "<br/>");  
document.write(str - a + "<br/>");  
document.write(str * "2" + "<br/>");  
document.write(str / 2 + "<br/>");
```

Имейте в виду, так как переменная str является строковым типом, то переменная a типа Number неявно преобразуется в строку и далее производится операция конкатенации. Но если операция сложения для строкового значения существует, то операции деления и вычитания для строк отсутствуют, соответственно, действия будут происходить с числами.

Задание 13. Необходимо написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам ($S = ab/2$). Сценарий разместить в разделе body документа. С помощью скрипта вывести в окно браузера инкремент площади.



Площадь прямоугольного треугольника со сторонами 8 и 10 = 40
Инкремент площади = 39

Алгоритм решения задачи на javascript:

- Инициализация двух переменных.
- Вычисление площади.
- Вывод инкремента с использованием метода write().

Задание 14: Даны переменные:

a="12";

b="7.15";

Найдите остаток от деления числовых значений переменной a на переменную b.

Используйте функции преобразования типов.
Результатом должно быть число 5.

Рекомендации:

Для округления числа следует использовать метод Math.round():

x = Math.round(x)

Задание 15. Допишите фрагмент кода для конвертации строкового значения в числовое. Для вывода результата используйте метод document.write:

```
var x = prompt("Введите значение x ",  
"0");  
var y = prompt("Введите значение y ",  
"0.0");
```

```
var integerX = ...
```

```
var floatY = ...
```

```
var result = integerX + floatY;
```

```
...;
```

Задание 16. Найдите ошибки в следующем фрагменте кода:

```
var firstName = prompt("Введите имя";  
"сюда");  
var lastName = promt("Введите  
фамилию");  
int age = prompt("Введите возраст");  
alert(Age);
```

Задание 17. Вычислить значение выражения по формуле (все переменные принимают вещественные значения):

$$\frac{x^2 - 7x + 10}{x^2 - 8x + 12}$$

Предлагаемый алгоритм:

- Запрос значения x
- Вычисление выражения
- Вывод результата при помощи метода alert

Задание 18. Найдите ошибки в следующем фрагменте кода:

```
var a = prompt("Введите число 11", "0");  
if (a = 11)  
    document.write("Введенное значение верно.");  
}
```

Задание 19. Запрашивать у пользователя возраст (метод prompt()). В зависимости от введенного возраста выдавать сообщение «Вы слишком молоды» (в случае введенного значения менее 20 лет) или «Вы нам подходите» (в случае введенного значения более или равного 20 годам). Итоговое сообщение выводить с помощью метода document.write().

Задание 20. Написать программу, которая по паролю определяет уровень доступа сотрудника к секретной информации в базе данных. Доступ к базе данных имеют сотрудники, разбитые на три группы по уровням доступа. Они имеют следующие пароли:

- 9583, 1747 – выдавать сообщение "доступны модули баз A, B и C";

- 3331, 7922 - выдавать сообщение "доступны модули баз В и С";
- 9455, 8997 – выдавать сообщение "доступен модуль базы С".

Выводить доступные модули на запрос.

Задание 21. Создать два массива: countries – с названием стран, и population – с населением этих стран. Вывести название страны и ее население (использовать метод document.write).

Комментарии к выполнению:

- Вывод элементов массива оформить в виде функции.
- Сначала выполнить задание с помощью цикла for с счетчиком, затем — с помощью цикла for in.

Удаление элементов массива

В javascript удалить элемент массива можно при помощи оператора delete:

```
var myColors = new Array("red", "green", "blue");
delete myColors[1];
alert(myColors); // red,,blue
```

Задание 22. Создать массив из трех элементов, значения элементов запрашивать у пользователя. Удалить второй по счету элемент, после чего вывести элементы массива на экран, каждый элемент с новой строки.

Задание 23. Дополните код согласно заданию: В переменную a присвоить возвращаемое значение метода concat, передав в этот метод два значения 4 и 5. Что будет выведено на экран?:

```
var a = [1, 2, 3];
a = ...
document.write(a);
```

Задание 24. Дан массив из чисел: 1, 2, 3, 4, 5. Распечатать массив в обратном порядке (5, 4, 3, 2, 1), используя функцию javascript reverse()

Задание 25. Дан массив из элементов: 'c', 5, 2, 'b', 3, 1, 4, 'a'. Распечатать массив, предварительно отсортировав его по возрастанию (1, 2, 3, 4, 5, a, b, c)

Задание 26. Дан массив из элементов: 1, 2, 3, 4, 5. Распечатать исходный массив и массив, получившийся из элементов исходного массива, с разделителем «+» (метод javascript join – 1+2+3+4+5)

Задание 27. Дана матрица 5 x 5. Вывести элементы матрицы, принадлежащие интервалу[-5; 7].

При исходной матрице:

```
[3, 15, 25, 15, -2],
[4, 11, 5, 11, 12],
[12, 22, 12, -3, -10]
[22, 22, 32, -33, -12]
[2, 22, 12, -12, -10]
```

Результат должен быть:

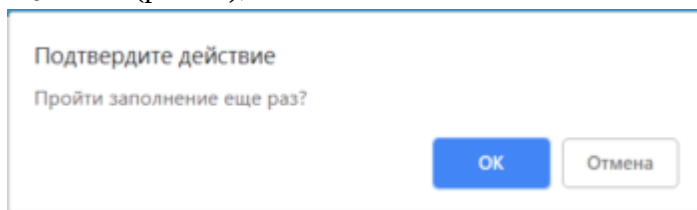
```
3 -2 4 5 -3 2
```

Задание 28. Дополните код согласно заданию: В переменную povtor присвоить результат выполнения функции [confirm](#), которая выводит фразу «Пройти заполнение еще раз?» и принимает ответ пользователя (ок или отмена).

Фрагмент кода:

```
function say() {
    document.write("Вы здесь?" +
"<br/>");
}
do {
```

```
say();
povtor = ...
} while (povtor);
```



Задание 29. Создать функцию без параметров, которая вычисляет разность двух вводимых пользователем в диалоговые окна числа. Функция должна выводить сообщение с результатом.

Задание 30. Необходимо запросить у пользователя ввести возраст (метод `prompt()`). Вызывать функцию, которая выводит диалоговое окно с сообщением *Привет, малыши!*, если возраст менее 18, и *Здравствуй, юноша!*, если возраст более или равен 18.

Задание 31. Создать объект Сотрудник, который содержит сведения о сотрудниках некоторой фирмы, такие как Имя, Отдел, Телефон, Зарплата (использовать функцию-конструктор и ключевое слово `this`). Создать экземпляр объекта.

Задание 32. Используя методы объекта `date`, вывести на экран время в формате: час:минута:секунда (пример: 14:23:11)

При этом, если секунды и минуты попадают в интервал от 0 до 10, они должны выводиться с нулем впереди. т.е. вместо 18:7:3 у Вас должно выводиться 18:07:03

Задание 33. В html-документе создайте элемент текстовое поле с установленным по умолчанию текстом «Привет!»:

Посмотрите на результат в браузере.

- ☐ Выведите в модальное окно значение атрибута `value`.
- ☐ Измените значение атрибута `type` на значение `<button>`. Посмотрите на результат работы скрипта.

Задание 34. Выдать сообщение о размерах окна браузера: например, «размеры окна браузера 600 x 400»

Задание 35. Дана страница с элементами для заполнения:

Дополните пустые места фрагмента кода, расположенного ниже, проверяющего правильность заполнения двух текстовых полей: *имя* (`id="name"`) и *минут* (`id="minutes"`). Используйте проверку на оставление поля пустым ("") и правильного формата заполнения числового поля (`isNaN()`).

Задание 36. В HTML-документе **s-1.html** создайте форму с текстовыми полями, в

которые пользователь вводит два числа, и кнопкой типа Submit. После нажатия кнопки вызывается PHP-скрипт, который выводит только большее из двух чисел или сообщение, что два числа равны.

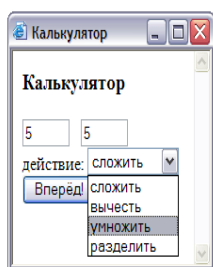
Задание 37. Используя PHP-скрипт и форму в одном документе, создать PHP-файл **s-2.php**, который позволяет пользователю передать два числа и указать операцию, выполняемую над ними (см. рис.). Предусмотреть проверку вводимых данных.

В предыдущем примере файл `top.php` будет включён в код страницы только один раз, несмотря на то, что в коде есть две команды.

Вот и всё...

Ну вот, вроде бы и всё, что я хотел рассказать Вам на сегодня. Будут какие-нибудь проблемы с PHP — [пишите мне](#), я помогу!

В качестве домашнего задания: сделать **простейший калькулятор**. В качестве подсказки, вот Вам рисунок:



И вот ещё что: стоит ли вообще задавать подобные «домашние задания»?

Успехов Вам!

Комментариев к статье нет

Добавить комментарий:

Задание 38. Создать HTML-документ **s-3.html**, реализующий следующую задачу (*Вариант определяется номером студента в списке в таблице рейтинга*):

Нечетный вариант. Список содержит следующие операции: четные, нечетные, простые, составные. Вывести все числа из диапазона от 1 до N, согласно выбранному из списка действию (N вводится пользователем в текстовое поле).

Четный вариант. Список содержит следующие операции: четные делители, нечетные делители, простые делители, составные делители, все делители. Вывести делители числа N, согласно выбранному из списка действию (N вводится пользователем в текстовое поле).

Задание 39. В HTML-документе **s-4.html** создайте форму с текстовым полем, в которое пользователь вводит свой логин, и кнопкой типа Submit. Далее после нажатия кнопки вызывается PHP-скрипт который проверяет, зарегистрирован ли этого пользователь. При этом таких пользователей (разных логинов) должно быть 4. Если введен один из существующих логинов, должно выводиться приветствие для этого человека. Например, если введен логин `Ivan_php`, должно выводиться приветствие «Здравствуйте, Иванов Иван Иванович». Если введен неизвестный логин, должно выводиться сообщение «Вы не зарегистрированный пользователь!».

Задание 40. Реализовать обработку формы на PHP для регистрации/авторизации пользователя на сайте (добавление нового пользователя в базу данных, проверка пароля для существующего пользователя).

Тема 3. Регистрация домена и выбор хостинга

Задание 1. Создайте резервную копию сайта и базы данных на локальном сервере.

Задание 2. Выберите хостинг для переноса сайта.

Задание 3. Зарегистрируйте доменное имя для сайта.

Задание 4. Создайте на хостинге базу данных, выполните импорт.

Задание 5. Перенесите данные с локального сервера на хостинг.

Задание 6. Настройте сайт на хостинге.

Задание 7. Выполните тестирование сайта.

Критерии оценки:

0 баллов - ставится, если студент не выполнил лабораторную работу.

1 балл - ставится, если студент обнаруживает знание и понимание основных положений лабораторной работы, но при выполнении заданий допущены ошибки или задание выполнено на 50%; оформление работы выполнено недостаточно последовательно (отсутствуют цель/листинг/результаты/выводы).

2 балла - ставится, если студентом при выполнении заданий допущены неточности или задание выполнено на 70%; оформление работы выполнено с ошибками (отсутствуют цель/выводы).

3 балла - ставится, если студент полностью выполнил задание, правильно ответил на теоретические вопросы преподавателя, оформление работы выполнено последовательно и полно (присутствуют цели работы, задания, листинг программ, результаты и выводы).

Самостоятельная работа студента

Включает проработку конспектов лекций, обязательной и дополнительной учебной литературы в соответствии с планом занятия; выполнение заданий. Основной формой проверки СРС является устный фронтальный опрос на занятии или письменные ответы на вопросы для проверки знаний по теме.

СРС 1. Основы JavaScript

Задача 1.

Создать скрипт для отрисовки таблицы умножения в виде таблицы:

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Задача 2.

Написать функцию, которая выводит на экран дату, которая наступит через n дней от сегодняшней.

- n — количество дней, которое нужно запросить у пользователя при загрузке страницы (от 1 до 1000), это значение передается в функцию в качестве параметра.
- при этом итоговую дату надо вывести в нормальном формате на русском языке.

Например так (пример для ввода числа 79):

Через 79 дней будет вот такая дата:

День недели: Воскресенье

Число: 10

Месяц: Октябрь

Год: 2010

Задача 3.

Создайте веб-страницу и расположите в ней тег `img` с изображением грустного смайлика. Щелчок на изображении (`onclick`) вызывает метод (пользовательская функция). В скрипте опишите метод, который «спрашивает», как зовут пользователя, а затем приветствует его по имени, и меняет изображение на улыбающийся смайлик:

Как вас зовут?

Рад вас видеть, Иван!

☐ Не давать этой странице создавать дополнительные диалоговые окна

СРС 2. Основы PHP

Задача 1.

В HTML-документе `s-1.html` создайте форму с текстовыми полями, в которые пользователь вводит два числа, и кнопкой типа `Submit`. После нажатия кнопки вызывается PHP-скрипт, который выводит только большее из двух чисел или сообщение, что два числа равны.

Задача 2.

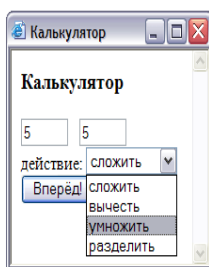
Используя PHP-скрипт и форму в одном документе, создать PHP-файл **s-2.php**, который позволяет пользователю передать два числа и указать операцию, выполняемую над ними (см. рис.). Предусмотреть проверку вводимых данных.

В предыдущем примере файл `top.php` будет включён в код страницы только один раз, несмотря на то, что в коде есть две команды.

Вот и всё...

Ну вот, вроде бы и всё, что я хотел рассказать Вам на сегодня. Будут какие-нибудь проблемы с PHP — [пишите мне](#), я помогу!

В качестве домашнего задания: сделать **простейший калькулятор**. В качестве подсказки, вот Вам рисунок:



И вот ещё что: стоит ли вообще задавать подобные «домашние задания»?

Успехов Вам!

Комментариев к статье нет

Добавить комментарий:

Задача 3.

Создать HTML-документ **s-3.html**, реализующий следующую задачу:

Список содержит следующие операции: четные, нечетные, простые, составные. Вывести все числа из диапазона от 1 до N, согласно выбранному из списка действию (N вводится пользователем в текстовое поле).

Критерии оценки:

0 баллов – самостоятельная работа не выполнена.

1 балл – демонстрирует, лишь поверхностный уровень выполнения работы, в содержании выполнения задания допущены принципиальные ошибки.

2 балла – ставится тогда, когда студент выполнил самостоятельную работу, но дает не точные ответы на заданные вопросы.

3 балла – ставится тогда, когда студент выполнил самостоятельную работу, показан высокий уровень освоения студентом учебного материала, содержание выполнения задания не содержит ошибок.

Аттестационная работа

Аттестационная работа проверяет знание студентов по изученному разделу. Проводится в форме тестирования.

Тестовые задания:

1. Текст заключенный в теги отобразится
 - a. Подчеркнутым текстом
 - b. Курсивом
 - c. Без изменений
 - d. Жирным шрифтом
2. Какое расширение нужно использовать для сохранения интернет-страницы, набранной в текстовом редакторе?
 - a. .doc
 - b. .html
 - c. .php
 - d. .js
4. Укажите правильный формат задания атрибутов
 - a. имя_тега имя_атрибута="значение"
 - b. имя_атрибута имя_тега="значение"
 - c. имя_тега значение="имя_атрибута"
 - d. значение имя_атрибута="имя_тега"
5. Программа для просмотра гипертекстовых страниц называется
 - a. Браузер
 - b. Протокол
 - c. Сервер
 - d. HTML
6. С помощью какого атрибута можно изменить цвет ссылки?
 - a. Color
 - b. Link
 - c. Bgcolor
 - d. Text
7. Какую функцию выполняет атрибут border в теге <TABLE>
 - a. Задаёт толщину рамки в пикселах
 - b. Выравнивает текст в ячейке
 - c. Определяет задний фон
 - d. Задаёт расстояние между содержимым ячейки и её рамкой
8. Какой атрибут является обязательным для тега
 - a. align
 - b. src
 - c. border
 - d. alt
9. Выберите правильные значения свойства position
 - a. absolute
 - b. left
 - c. top
 - d. relative
 - e. bottom
10. Сколько ошибок в следующем коде:

```
#wrapper{  
    position:bottom;  
    color:black,  
    z-index:100  
}
```

- a. 3
 - b. 2
 - c. нет ошибок
 - d. 1
11. Для чего предназначено свойство padding в CSS
- a. Внешний отступ
 - b. Фоновый цвет
 - c. Внутренний отступ
 - d. Прозрачность
12. Для чего предназначено свойство z-index:
- a. Обычное позиционирование влево-вправо, вверх-вниз
 - b. Располагать элементы в определенном порядке, имитируя третье измерение, которое перпендикулярно экрану
 - c. Задаёт размер элемента по диагонали
 - d. Ни на что оно не влияет
13. Какие из вариантов задания цвета являются неверными
- a. background-color:&hE99295;
 - b. background-color: red;
 - c. background-color:#ff00ff;
 - d. background-color: RGB(255, 0, 255);
 - e. background-color:\$00E99295;
14. За что отвечает свойство text-transform?
- a. Преобразованием текста в заглавные и прописные символы
 - b. Вращение текста по часовой стрелке
 - c. Изменением цвета текста
 - d. Динамическим изменением шрифта
15. Чтобы задать прозрачность элемента на веб-странице необходимо использовать свойство...
- a. opacity
 - b. display
 - c. background-color
 - d. position
16. Предназначение свойства clear в CSS
- a. Очищает всю страницу
 - b. Очищает всю страницу, кроме блока к которому применяется данное свойство
 - c. Устанавливает, с какой стороны элемента запрещено его обтекание другими элементами
 - d. Очищает блок, к которому применяется данное свойство
17. Как правильно вставить внешний скрипт?
- a. <script src="sample1.js">
 - b. <script href="sample1.js">
 - c. <#include "sample1.js">
 - d. <script name="sample1.js">
18. Как можно вывести сообщение "Hello World!" с помощью JavaScript?
- a. msg("Hello World");
 - b. alertBox("Hello World");
 - c. alert("Hello World");
 - d. msgBox("Hello World");
19. Как объявить функцию в JavaScript?
- a. function MyFunction()
 - b. function = MyFunction()
 - c. function = New MyFunction()

d. function:MyFunction()

20. Как создать новый массив в JavaScript?

a. var colors = (1:"yellow", 2:"purple", 3:"blue")

b. var colors = "yellow", "purple", "blue"

c. var colors = ["yellow", "purple", "blue"]

d. var colors = 1 = ("yellow"), 2 = ("purple"), 3 = ("blue")

Критерии оценки:

Процент выполненных тестовых заданий	Количество набранных баллов
91% - 100%	10
81% - 90%	9
71% - 80%	8
61% - 70%	7
51% - 60%	6
<50%	0

Контрольная работа

Контрольная работа предполагает выполнение письменной работы с обязательными практическими примерами по одной из тем.

Тематика контрольных работ

- Тема 1.** Создание интернет-сайта для музыкальной группы.
Тема 2. Создание интернет-сайта для студенческой группы.
Тема 3. Создание интернет-магазина ювелирного салона.
Тема 4. Создание интернет-магазина бытовой техники.
Тема 5. Создание интернет-сайта для салона автомобилей.
Тема 6. Создание интернет-сайта мебельного салона.
Тема 7. Создание интернет-магазина музыкальных инструментов.
Тема 8. Создание интернет-магазина промышленных товаров.
Тема 9. Создание интернет-магазина стройматериалов.
Тема 10. Создание интернет-сайта для галереи фотографий.
Тема 11. Создание интернет-магазина детских товаров.
Тема 12. Создание интернет-магазина одежды.
Тема 13. Создание интернет-магазина косметики.
Тема 14. Создание интернет-сайта библиотеки.
Тема 15. Создание интернет-сайта мобильного оператора.
Тема 16. Создание интернет-магазина аудио-техники.
Тема 17. Создание интернет-магазина обуви.

Критерии оценки:

№	Критерий	
1.	Соответствие содержания заявленной теме	16
2.	Логичность и последовательность в изложении материала	16
3.	Способность к работе с литературными источниками, Интернет-ресурсами, справочной и энциклопедической литературой	16
4.	Способность к выполнению практических заданий по заданной тематике	16
5.	Использование компьютерных программ при выполнении задания	16
6.	Анализ полученных результатов, обоснованность выводов	16
7.	Правильность оформления (наличие всех структурных частей, структурная упорядоченность, ссылки на литературу, цитаты, таблицы, рисунки и т.д.);	16
8.	Соответствие оформления правилам компьютерного набора текста (соблюдение объема, шрифтов, интервалов, выравнивания текста на страницах, нумерация страниц и т.д.);	16
9.	Наличие презентационного материала	16
10.	Правильность ответов на заданные вопросы по заявленной теме	16
	Итого	10