**2 СЛАЙД**

**Актуальность создания web-сайта:**

В наше время у каждого человека есть домашний компьютер или ноутбук. Люди пользуются ими не только для выполнения каких-либо задач, но и для получения информации. Каждый современный компьютер имеет возможность подключения к интернету.

Необходимость подключения к интернету обусловлена тем, что большинство полезной информации, полученной с разных источников, содержится именно на страницах Всемирной паутины. Раньше, когда не существовало интернета, основными источниками получения информации являлись телевидение, радио, книги, газеты и т.д. Но теперь они отошли на задний план, так как в интернете гораздо больше возможностей, таких как поисковые системы (Google, Yandex и т.д.), которые помогут очень быстро и эффективно найти то, что вас интересует, а так же колоссальное количество разнообразных источников, которые помогут собрать информацию по кусочкам в одно целое. Именно легкость в обращении с интернетом и удобство в получении нужной информации делает интернет очень популярным во всём мире. И, поскольку большинство людей планеты отдают своё предпочтение не книгам, а web-сайтам — создавать их становится очень актуально и прибыльно.

***Актуальность темы «Энергосберегающие средства»:***

Тема энергосбережения приобрела большую актуальность, а повышение энергетической эффективности определено Президентом Российской Федерации в качестве одного из приоритетов инновационного развития страны.

Актуальность энергосбережения на предприятии и повышение энергетической эффективности производства обусловлена следующими факторами:

1. Высокие затраты на энергетические ресурсы.
2. Постоянный рост тарифов на энергетические ресурсы, затрудненность подключения дополнительных мощностей.
3. Увеличение энергопотребления ограничивает рост производства.
4. Предприятиям необходима качественная, надежная и бесперебойная поставка энергетических ресурсов, что не может гарантировать изношенное энергооборудование.

**Цель работы:**

Создание Web-сайта «Энергосберегающие средства»

**Задачи:**

* Разработка структуры web-сайта и его дизайна
* Поиск необходимой информации по выбранной теме
* Закрепление знаний языков HTML, CSS, JavaScript
* Создание оригинальных изображений в Adobe Photoshop

# 3 СЛАЙД

Структура web-сайта выполнена с помощью одинаковой для всех страниц блочной вёрстки, включающей в себя один главный блок (*page*), состоящий из 4 второстепенных блоков (*header*, *menu*, *content*, *footer*). (рис. 1)

Логотип будет являться блочным элементом только на главной странице, а на всех других он будет выполнен только лишь как изображение-ссылка.

Первый блок представляет собой логотип web-сайта, а так же выполняет роль ссылки на всех страницах, кроме главной (на этих же страницах при наведении курсора на логотип появляется стильная всплывающая подсказка). (рис. 1.1)

Второй блок представляет собой меню, состоящее из 4 изображений-ссылок и 1 блока, соответствующего активной странице (на главной все 5 разделов — 5 изображений-ссылок). (рис. 1.2)

1. Аллокатор. На данной странице располагается вся необходимая информация о распределителе затрат (аллокаторе) и 1 картинка.
2. Биллинговый сервис. На данной странице описываются различные расчеты, связанные с тепловыми затратами и имеются 2 картинки.
3. Храним тепло. На этой странице рассматривается принцип действия жидкой изоляции и ее преимущества и имеются 4 картинки.
4. Труба из полибутена. На этой странице оценивается уникальная технология по энергосбережению — труба из полибутена. Здесь так же расположен блок со скриптом — выпадающий контент.
5. Галерея. На данной странице располагается галерея (оформлена средствами CSS3) с динамичными изображениями различных счётчиков.

В центральном блоке располагается основная информация активной страницы (текст, картинки, слайдер, выпадающий контент и галерея). (рис. 1.3)

В четвертом блоке находится контактная информация, а также год создания, символ с названием фирмы и знак ©. (рис. 1. 4)

В данные блоки встроены изображения, создание которых описывается в главе 3.

**4-5 СЛАЙД**

Для редактирования и форматирования этой информации были использованы ***стили CSS***. Таким образом, для тегов **a.button1** и **a.button1:hover** были использованы стили, которые в качестве фона блока, находящегося в пассивном состоянии используют первое изображение, созданное при помощи Adobe Photoshop, а при наведении на блок курсора мыши — фон кнопки заменяется другой картинкой.

Для логотипа всех страниц, кроме главной был реализован стиль, выводящий подсказку при наведении курсора на шапку.

Так же с помощью свойств **CSS3** была создана галерея. При щелчке на любую картинку или наведении на нее курсора — она изменяется в соответствии с параметрами стиля.

**6-8 СЛАЙД**

Для расширения возможностей web-сайта были применены различные скрипты.

1) Скрипт часов. Создаётся функция *showTime*. Затем объявляем переменную, которая будет хранить текущую дату и время, исходя из настроек ОС; далее объявляем переменные для часов и минут. Задаем им условия для формата отображения чисел; объявляем еще одну переменную, которая будет отвечать за сам формат отображения уже готовых часов; выводим полученные часы как html-строку, стилизованную с помощью CSS. при этом задаем интервал, через который будет обновляться картинка часов.

2) Скрипт кнопок прокрутки. Создается функция jQuery. Объявляется переменная *play*, которая будет отвечать за выполнение прокрутки (изначально 0 — прокрутка отключена). Затем создается функция, запускающаяся при прокрутке окна, которая останавливает прокрутку, если окно находится достигает самого верха/низа страницы. Далее, если окно опускается на 250 или больше пикселей вниз от самого верха страницы, то появляется кнопка прокрутки вверх. Затем выполняется новая функция, и если окно поднимается выше (отступ сверху становится равен или меньше 250 пикселей), то кнопка прокрутки наверх исчезает, иначе она появится.

Затем выполняется тот же ряд действий, только для кнопки прокрутки вниз. Условия меняются на величину отступа сверху больше/меньше или равную высоте документа минус 999 пикселей.

Далее создаются функции, которые при определенных условиях запускают прокрутку страницы в нужном направлении при нажатии на соответствующую кнопку, и останавливают прокрутку при нажатии на любую кнопку. Здесь же указывается скорость прокрутки.

3) Скрипт слайдера. Сначала в переменных сохраняются настройки: *SlideSpeed* - скорость перелистывания слайдов, *TimeOut* - время спустя которое происходит автоматическая смена слайда, *NeedLinks* - управляет ссылками "Следующий " и "Предыдущий" — значение этой переменной равно *true* — ссылки будут отображаться.

Далее мы устанавливаем необходимые CSS свойства для слайдов с помощью метода *css()*. Блоки со слайдами мы накладываем друг на друга используя абсолютное позиционирование, затем скрываем их все *hide()*, а затем показываем только первый. Переменная *slideNum* — это номер активного слайда, то есть счётчик.

Основная логика работы нашего jQuery слайдера — это функция *animSlide*. Она принимает один параметр. Если мы передадим в неё строки *next* или *prew*, то сработают условные операторы и будет отображён соответственно следующий или предыдущий слайд. Если же мы пошлём в качестве значения цифру, то это число станет активным слайдом и он будет показан.

Таким образом эта функция скрывает текущий *div*, высчитывает новый и показывает его.

Методу *fadeIn()*, который делает видимым активный слайд, задан второй аргумент. Это, так называемая функция обратного вызова. Она выполняется, когда слайд полностью появится. В данном случае это сделано для обеспечения автоматической прокрутки слайдов. Функция *rotator*, определённая ниже, вызывает снова функцию *animSlide* для перехода на следующий слайд через необходимый нам интервал времени: мы получим замыкание, обеспечивающее постоянную прокрутку.

Всё работает нормально, но нам нужно остановить выполнение автоматики, если пользователь подводит курсор к слайдеру, или нажимает кнопки. Для этого создана переменная *pause*. Если её значение положительно — *true*, значит автоматического переключения не будет. С помощью метода *hover()*, мы указываем: очистить таймер если он запущен - *clearTimeout(slideTime)*, и установить *pause = true*. А после отвода курсора, отключить паузу и запустить замыкание *rotator()*.

Дополнительно нам нужно создать новые элементы на странице и поместить их в нужное место. Используя цикл *each()* для каждого слайда (*div*-а с классом *.slide*), создадим элемент *span* внутри которого будет число — номер слайда. Это число используется в функции анимации, для перехода к слайду с этим номером.

Затем на ссылки "следующий—предыдущий" повесим обработчики. В качестве обработчика клика используем ту же самую функцию *animSlide*, с аргументами *next* и *prew*.

Почему я создал разметку внутри скрипта, а не в HTML коде? Дело в том, что например, если у пользователя отключены скрипты - он не увидит элементов, которые не будут работать, и это не введёт его в замешательство.

4) Выпадающий контент...

**9-10 СЛАЙД**

Для разработки изображений был использован графический редактор Adobe Photoshop. Все операции описаны в документе ПЭР-Савинов.doc.

В основном использовались такие инструменты, как: "Кисть", "Ластик", "Прямоугольник", "Прямоугольник со скругленными углами", "Эллипс", "Горизонтальный текст", "Прямоугольная область", "Линейка".

Также была использована функция — "Свободное трансформирование".

Проводились манипуляции с такими параметрами, как: "Размер", "Цвет заливки", "Непрозрачность", "Нажим".

**11-12 СЛАЙД**

Результатом проделанной работы является web-сайт «Энергосберегающие средства".

В ходе разработки структуры web-сайта были закреплены знания языков HTML и CSS. При реализации различных скриптов закреплялись знания языка программирования JavaScript, а так же его библиотеки jQuery. Так же были улучшены навыки работы с графическим редактором Adobe Photoshop при создании оригинальных изображений.