Препод: Крыж

**Тут какая-то тема, но я тупой так что да**

Гипертекст – текст, содержащий гиперссылки.

Гиперссылка – активная ссылка на другой объект.

Веб-страница – гипертекстовый документ в интернете. HTML + сопутствующие документы и возможно скрипты.

Веб-сайт – группа веб-страниц, которые объединены общей темой и оформлением, связаны гиперссылками.

Веб-сервер – программа, которая принимает запросы по протоколу HTTP и отвечает на них – возвращает веб-страницы.

Браузер­ – программа для просмотра веб-страницы на экране монитора.

**Какие бывают веб-страницы:**

1. Статические(готовые файлы. Быстро грузятся, нет нагрузки на сервер, но, невозможно работа с БД);
2. Динамические (полностью или частично создаются на сервере в момент запроса. Работают БД, но, загружаются медленнее и нагружают сервер(зависит от величины запроса)).

CMS – система управления содержимым сайта.

**Функции CMS:**

1. Создание разделов сайта;
2. Создание страницы;
3. БД пользователей;
4. Управление доступом;
5. Обеспечение навигации и поиска.

DHTML – интерактивная страница.

**Функции DHTML:**

1. Замена текста, оформления, рисунков;
2. Многоуровневые выпадающие меню;
3. Скрытие и показ частей страницы;
4. Проверка данных, введённые пользователем;
5. Выполнение вычислений;
6. Игры.

Эммит – набор сокращения для быстрейшего написания кода.

**\*Тут короче рассказывали про теги, но мне было скучно, и я смотрел на дополнения в маркетплейсе вижлы. Так что их не будет**

Неразрывный пробел – пробел, который не позволяет переносить текст. Является специальным символом.

Специальные символы – я не могу адекватно описать, так что проще: символы, которые пишутся в текст через амперсанд

**Список:**

Используется <ul> </ul> - в ненумерованном

<ol> </ol> - в нумерованном

Пункты: <li> </li>

Многоуровневые списки – списки просто пихаются в li списка

**Гиперссылка:**

<a href = “dd.html”> txt<\a> - гиперссылка на dd.html, текст, который отображается – txt. НЕ УКАЗЫВАТЬ АБСОЛЮТНЫЙ ПУТЬ

<a href = “mailto:dda@mail.com”> txt<\a> - Открыть мейл-клиент

Относительный путь на папку выше - ../

**Метки:**

<a href = “#s”> f</a> - по клику на гиперссылку, он переходит на метку

<a name = “s”> - метка, на которую она переходит

<a href = “nextpage.html#s”> f</a> - то же самое, только в файле nextpage.html

**\***

**Разметка:**

1. Логическая – .html
   1. Выделение – <em>
   2. Сильное выделение – <strong>
   3. Код – <code>
   4. Определение – <dfn>
   5. Цитата – <cite>
   6. Сокращение - <abbr>
   7. Отформатированный - <pre> - короче, сохраняет переносы
2. Графическая – .css

CSS подключается через

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

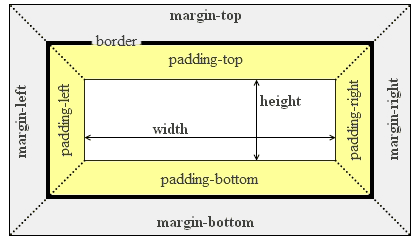
В эммите: link:css

**CSS:**

**Шрифты:** font-family; font-size; font-style; font-weight (100-900).

**Размеры и выравнивание:** background; width; height; text-align; text-indent – красная строка.

**Рамка, поля, отступы**: background; border; margin – верх, право, низ, лево; padding – отступы.



**Стили гиперссылок: color, text-decoration: none – убрать подчёркивание;**

**Так же, есть псевдоклас: hovered и visited;**

**Фоновый рисунок: «background:»я**

**Классы в css имеют синтаксис**: .name {}

**Так же:**

1. эту всю шнягу можно через запятую чтобы было несколько (стиль применялся к нескольким)
2. Если без запятой, то будет вложенность

**КАРТИНКА – СТРОЧНЫЙ ЭЛЕМЕНТ!**

**Отступы в пикче:** vspase, hspase.

**Альтернативный текст:** alt (тег обязательный).

**Всплывающая подсказка:** title.

**Размеры пикчи:** width, height (или как-то так я не шарю. Если картинка не подгрузится – будет пустой блок, что круче, ибо альтернатива – поломка разметки страницы).

**Вставка векторных рисунков:** <object data="vector.svg" type="image/svg+xml"></object> **IMAGE/SVG+XML – не путь, а «картинка ИЛИ svg+xml».**

**Вставка мультимедиа в HTML4:** <embed src=”” autostart=””></embed>.

**Вставка аудио в HTML5:** <audio src=""></audio>.

**Вставка видео в HTML5:** <video src="">Текст</video> **(Текст будет если видео не смогло).**

**Таблицы:** <table> <tr> <td>asdd</td><td>asdd</td> </tr> </table> - таблица на 2 ел-та

Если нужен заголовок - <th>

Если нужно объеденить колонки – colspan = “num”, где Num – сколько объеденить

Если нужно объеденить строки – rowspan = “num”, где Num – сколько объеденить

Выравнивание – align.

Выравнивание контента внутри ячейки- alingn, valign.

В CSS – text-align, vertical-align.

Интервал между ячейками – cellspasing.

Отступ внутри ячеек – cellpadding.

Отступ в CSS между ячейками – border-spacing

Интервал в CSS внутри ячеек - border-collapse

**БЛОКИ:**

<div> **-** самый простой. Контейнер.

Обтекание блока – float. Указывается сторона с которой обтекает

**DHTML**

DHTML- технология создания интерактивных сайтов, использующая HTML, CSS, ЯП, DOM (объектную модель документа)  
/\*child в DOM дереве – вложенный а не дочерний\*/

**Подключение JS-файла:** <script type=”text/javascript” src=”test.js”></script>. Type вроде не обязателен

**Вызов функции:** <a href=”#” onClick=”show(‘details’); return false;”>еуче</a>

/\*href=”#” – ведёт в никуда. Заглушка. Так можно.\*/

**Основные свойства CSS: width, height, background, background: url(‘image.jlf’), text-align, float, margin, a:hover, a:visited.**

**Новые свойства CSS:**

**Div.resize {width: 100px; height: 100px; resize: both; overflow: auto:}**

**Border-top-right-radius: 25pt 25pt; -** и прочая такая хрень

**Box-shadow, text-shadow**

**Rotate**

**Этапы создания сайта**

**Этапы:**

1. Подготовительный – создание любого сайта начинается с создания его структуры (карты), тематики, цели и определения аудитории;
2. Разработка макета – рисуется макет (примерно как всё расположено) и разрабатывается дизайн (как всё должно быть на сайте);
3. Вёрстка;
4. Программирование;
5. Наполнение контентом;
6. Раскрутка;
7. Администрирование.

**Формы**

**Создание формы:** <form method = {get (default), post – определяет метод отправки данных пользователя на сервер, action = адрес обработчика формы> txt and another shit and crap</form>

**Создание элементов управления:** <input type=тип\_создаваемого\_элемента name = имя элемента управления>

тип\_создаваемого\_элемента = text (поле ввода текста)

password (equal to text)

radio (radio button). Checked – показывает что активна

checkbox

submit – «подтверидить». Отправка инфы

reset – сброс элементов

textarea – многострочный текст

select > option – выпадающий текст,

button

**Фреймы**

Фреймы- /\*я не записал, так что своими\*/ это html док-т в html док-те

**Для работы, нужно вместо body написать frameset**

**Iframe –** То же что и фрейм, но нельзя изменять размеры динамически

**Форматирование CSS:**

**Красная строка –** text-indent: 1em/cm/in/mm/pt/pc/%

**Поля** – margin

**Отступы** – padding

**Рамка** – border (width – thin, medium, thick; style – none, solid, inset, outset, outset, groove, inset, ridge)

**Шрифт** – font-family

**Стиль –** font-style

**Капитель** – font-variant

**Жирность** – font-weight

**Межстрочный интервал** – line-height

**Оформление текста** – text-decoration

**Трансформация** – text-transform

**Выравнивание** – text-align

**Отступ первой строки** – text-indent

**Размер кегель** – font-size

**Цвет** – color

**XML**

XML – расширяемый язык разметки

**Краткое введение в Javscript**

**Javascript это:**

1. Интерпретируемый язык. Его интерпретатор обычно встроен в браузер.

2. Основное назначение – определять «динамическое» поведение страниц при загрузке (формирование страницы перед ее открытием) и при работе пользователя со страницей.

3. Текст на js может быть вложен в html-странице непосредственно или находиться в отдельном файле (как css).

4. Похож на языки джавы и шарпов синтаксически, но сильно отличается от них по внутреннему содержанию.

**Характеристика js:**

1. Язык ООП. Объекты в языке имеют «*тип, атрибут, метод*».
2. Переменные не имеют заранее заданного типа, то есть в разные моменты времени могут содержать значения разных типов.
3. Типы объектов могут быть: **number, string, function, object, undefined**.

Оператор ***typeof*** позволяет вычислить тип объекта.

**Сведения о синтаксе:**

В es-4:

var count = 25;

msg = ‘message about error’;

**Операторы как в шарпах.**

*For (j in obj) {dosmth;} – похожа на foreach*

***Включение js в html-страницу.***

Фрагменыт кода можно влючать в заголовок или тело html-дока. Кроме того, можно разместить код в отдельном файле, а в html-странице разместить ссылку на этот файл:

<html>

<head>

<script type=”text/javascript”>… </script>

<script type=”text/javascript” src = “scripts/myscript1.js”/ >

<head>

</html>

Код, ссылки на который размещены в заголовке, просто подсоединяется к странице и может быть использован, например, для определения реакций на пользовательские события.

Код, ссылки на который размещены в теле, исполняется при загрузке страницы и может непосредственно использоваться для формирования содержания страницы во время загрузки.

**Основные встроенные типы:**

1. Number – 64-разрядные числа с плавающей точкой.
2. String – строки в формате юникода
3. Array – массивы с переменными границами
4. Function: Функции. Каждая функция, кроме того, может служить конструктором объекта.
5. Boolean, Date, Math: логические значения, даты…

Тип String:

1. Строки могут содержать любое число символов или не содержать символов.
2. Строки объявляются путем указания составляющих их символов, заключенных в кавычки.
3. ДЖС интерпретирует одинарные кавычки как часть строки, если они находятся между двойными кавычками.

**jQuery**

jQery – библиотека JS, фокусирующаяся на взаимодействии JS и HTML.

Библиотека помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими.

Так же, библиотека предоставляет удобный API по работе с Ajax.

jQuery – создан в 2004 году Джоном Реззином. Библиотека была выпущена им на конференции «BarCamp» в 2006 году.

Возможности:

1. Кросс-браузерный выбор DOM объектов;
2. Перемещение и модификация DOM;
3. События;
4. Работа с CSS;
5. Визуальные эффекты и анимация;
6. AJAX-дополнения;
7. Расширяемость плагинами включая jQuery UI

jQuery Включается в станицу как один внешний JS-файл.

Вся работа с jQuery ведётся с помощью функции “$” или её псевдонима «jQuery».

Работу можно разделить на типа:

1. Получение jQuery-объекта с помощью функции $ и далее работать с ними с помощью различных методов jQuery-объекта.
2. Вызов глобальных методов у объекта $.

Плюсы от использования:

1. Не надо писать тривиальный код;
2. Простота использования и изучения;
3. Кросс-браузерность;
4. Отказ от длинных конструкций;
5. Библиотека Ajax
6. Огромное кол-во плагинов;
7. Повсеместное использование а значит кеширование браузерами и ускорение загрузки.

**jQuery UI**

Взаимодействия:

1. Перетаскивание;
2. Вложение;
3. Изменение размера;
4. Выделение;
5. Сортировка.

Виджеты – Аккордеон, Автодополнение, Кнопки, Выбор даты, Диалоговые окна, Progressbar, Sliders, Вкладки.

Чаще всего юзается по регуляркам в колонке script:

1. A[href^=asdasdasd] – всё что начинается с «asdasdasd»
2. a[href$=asd] – всё что заканчивается на «asd»
3. a[href\*=asd] – всё что содержит «asd» в рандомном месте
4. li:has(a) – li которые содержат a
5. :first – первое совпадение на странице
6. :last – последнее совпадение на странице
7. :first-child – первый дочерний элемент
8. :last-child(n) – первый дочерний элемент
9. :nth-child(even|odd) – чётный или нечётный дочерний элемент
10. :even – чётный
11. :odd - нечётный

**ОКР**

XML – расширяемый язык разметки

DTD – document type definition (задаёт возможную структуру для некоторого типа XML-документов: как могут называться эл-ты, как они могут вкладываться, какие у каждого эл-та могут быть атрибуты)

XSD – XML Schema Definition

**DTD не создаёт структуру DOM дерева в отличии от XSD**

Недостатки DTD:

1. Определения не являются расширяемыми;
2. С документом можно связать только одно определение DTD;
3. Не поддерживаются пространства имён;
4. Слабая типизация данных;
5. Синтаксис DTD не является синтаксисом XML;
6. Отсутствует поддержка DOM.

**CSS –** Каскадная таблица стилей. Описывает способы представления элемента веб страницы.

Способы добавления стилей в документ:

1. Встроеные – те, которые в браузере;
2. Вложенные – то, которые в самом html документе;
3. Внедрённные – в отдельном файле;
4. Импорт – через импорт.

Типы селекторов:

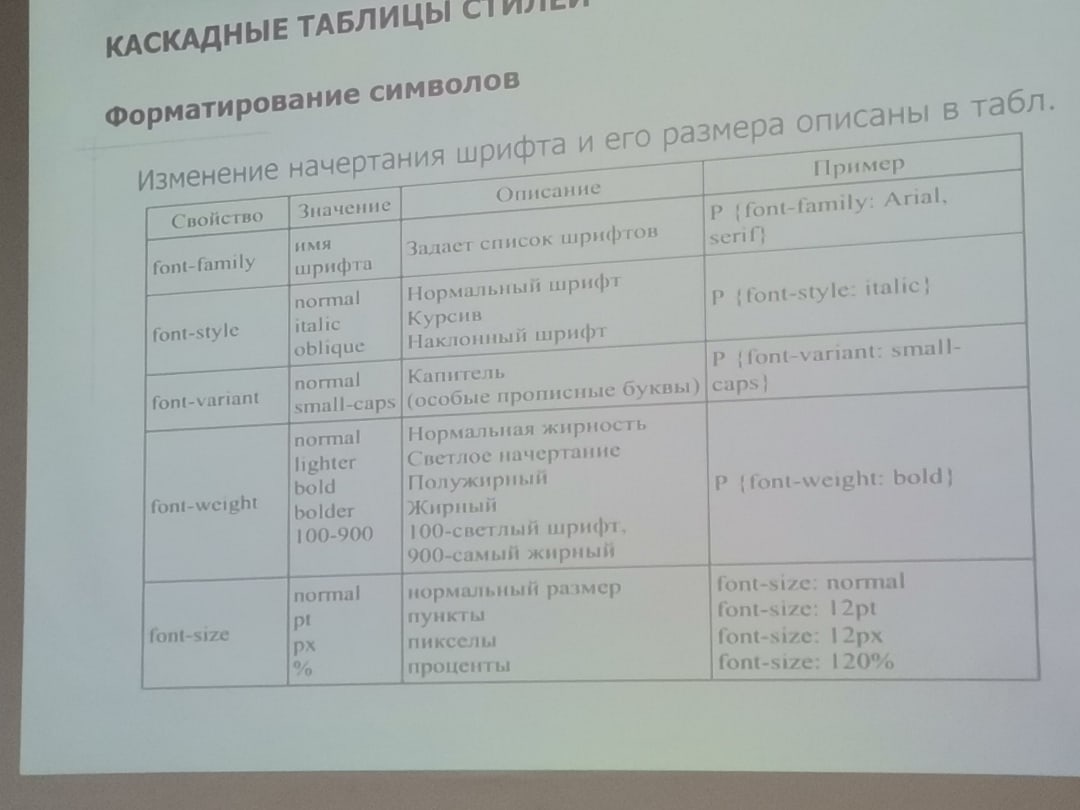
1. Теги
2. Класс
3. Id

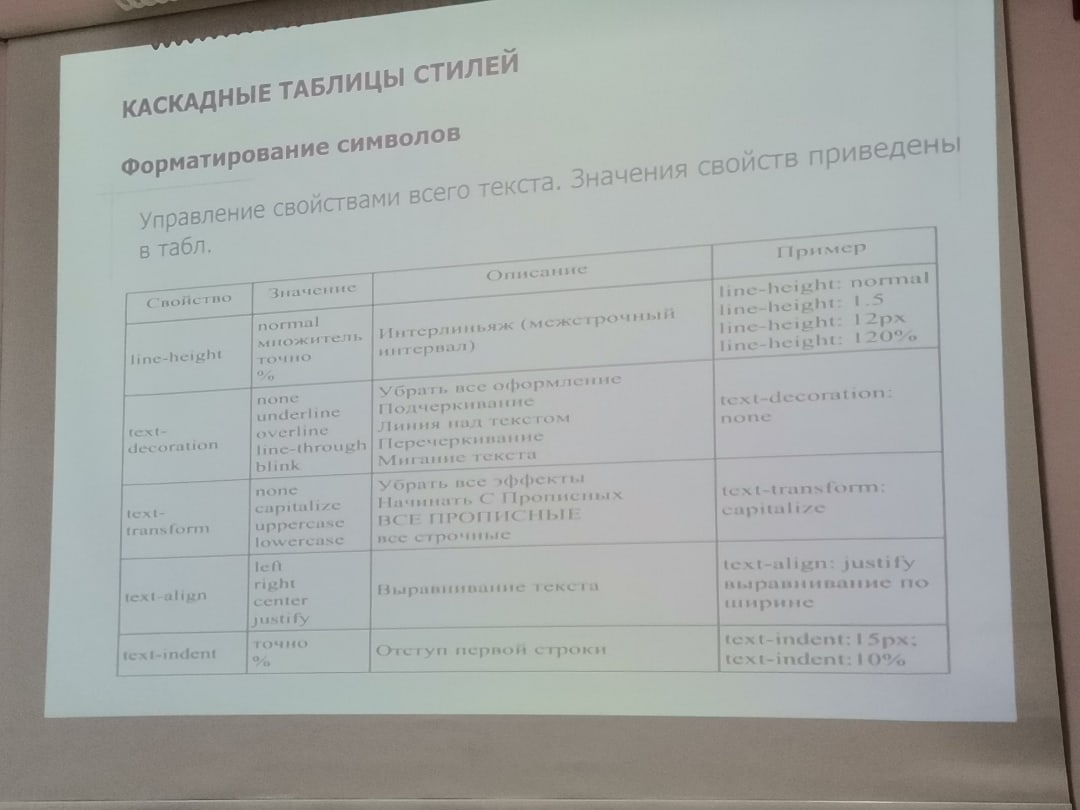
Наследование – перенос правил форматирования для элементов, находящихся внутри других.

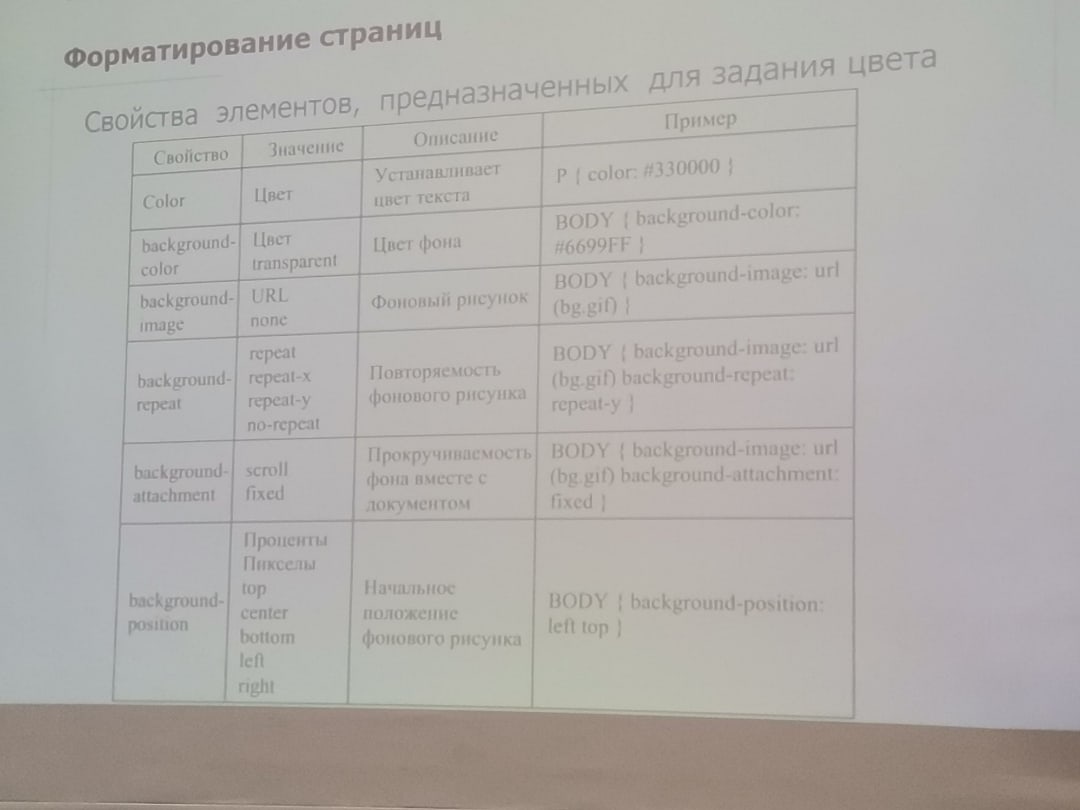
Px – относительные еденицы измерения

Padding – внутри

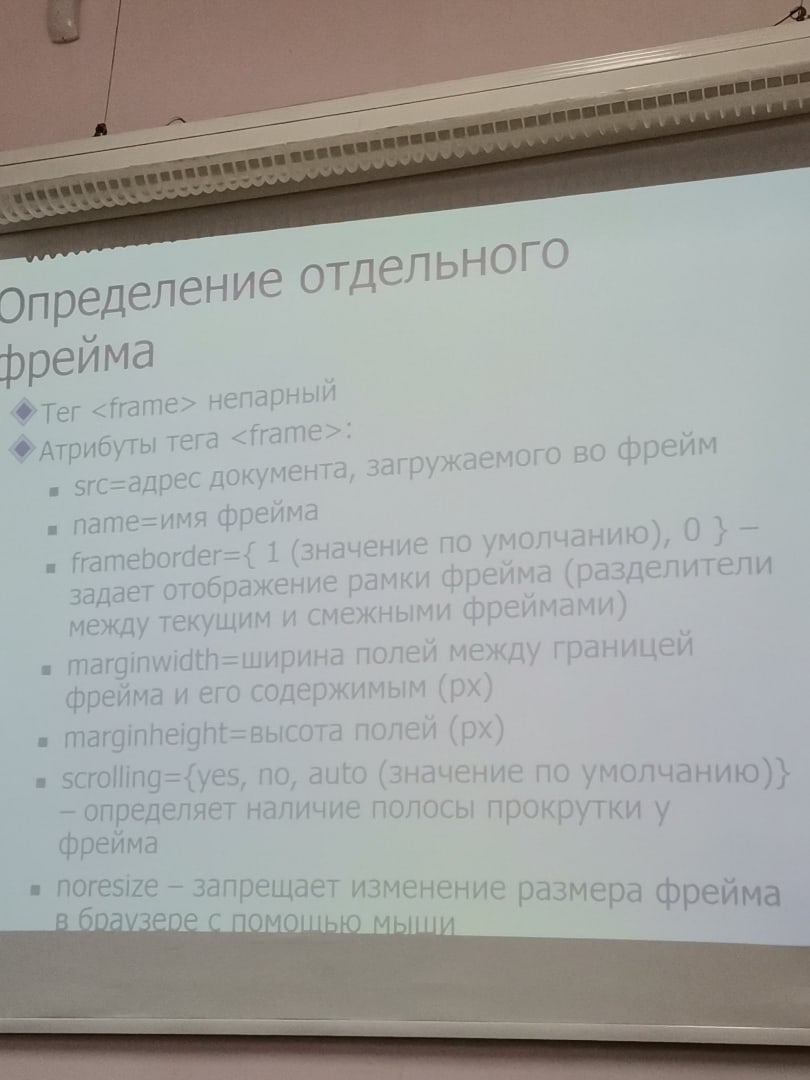
Margin – снаружи







BackgroundRepeat=”noRepeat” – или repeatX, repeat



Target – определяет в каком окне или фрейме загрузить гиперсылку.

**Введение в PHP**

PHP – это серверный язык создания сценариев.

Php – это аббревиатура от слов «personal home page».

Пхп программы могут выполняться 2 способами:

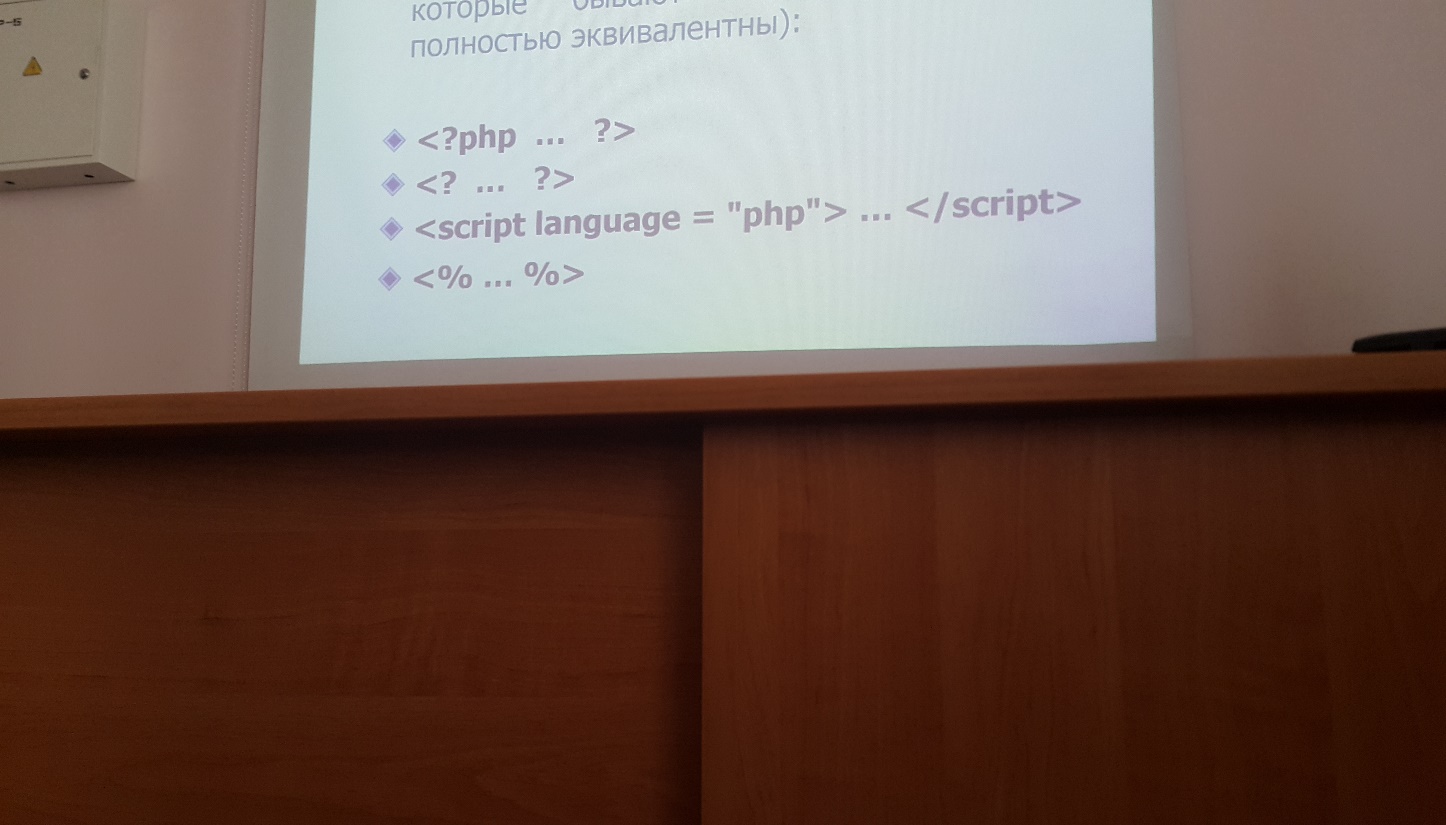
1. Как сценарное приложение веб-сервером
2. Как консольная программа

Процесс выполнения пхп-сценария при обращении браузера к серверу:

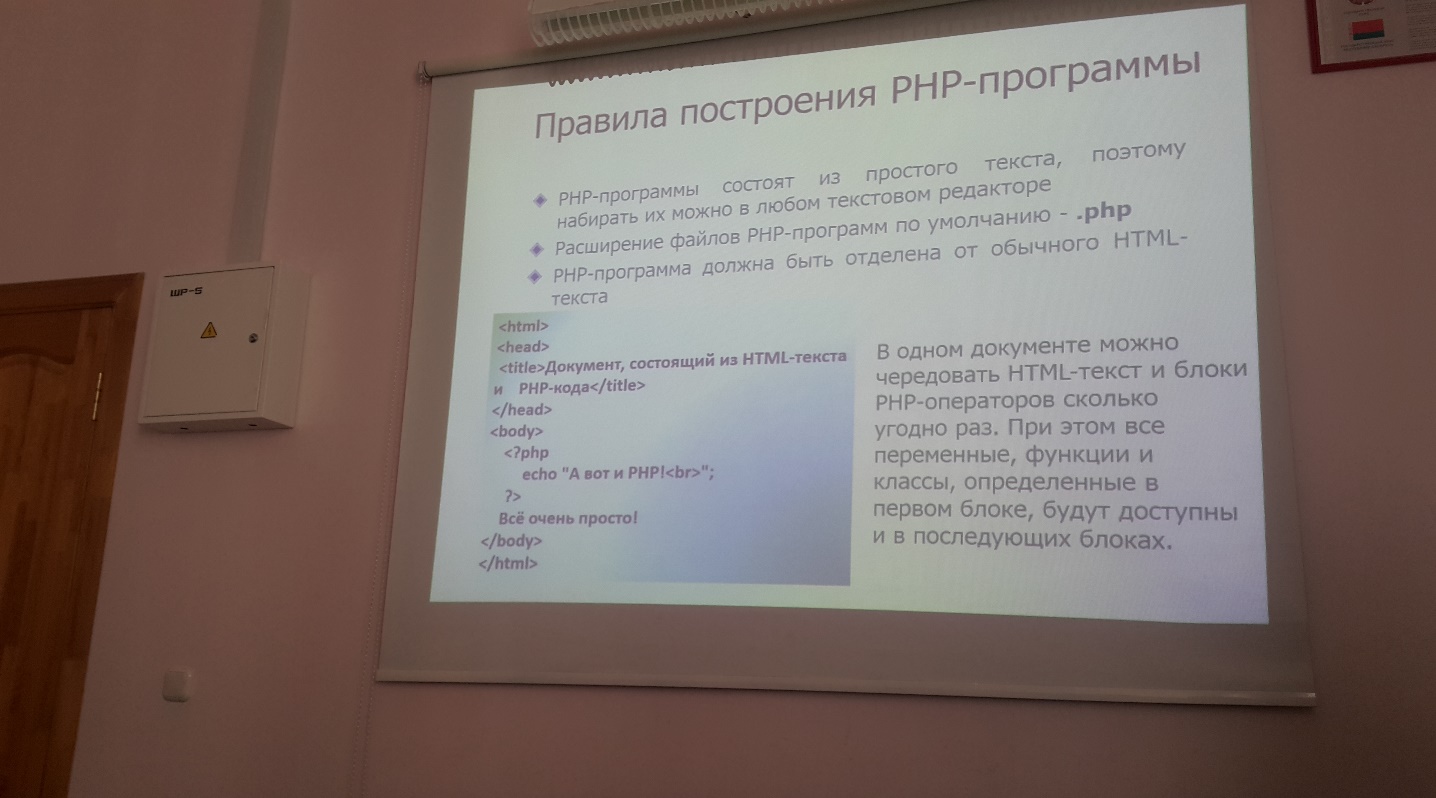
1. Вначале браузер запрашивает страницу с расширением *.php*
2. После чего веб-сервер пропускает программу через движок пхп.
3. Выдает результат в виде штмл-кода.

Синтаксис пхп подобен синтаксису языка си.

Команды заключаются в специальные теги, которые бывают четырёх видов:



Программа на пхп – это набор команд. Обработчику программы (парсеру) необходимо как-то отличать одну команду от другой. Для этого используются специальные символы-разделители.



Команда echo – вывод фактически всего, что встречается на веб-страницах (текст, разметка штмл, числа)

<?

Echo(“<p> Hello, world! <p>”);

Echo($var1);

Echo ‘<br>’;

?>

Константы пхп.

Объявляются при помощи функции – имя константы, второй – её значение.

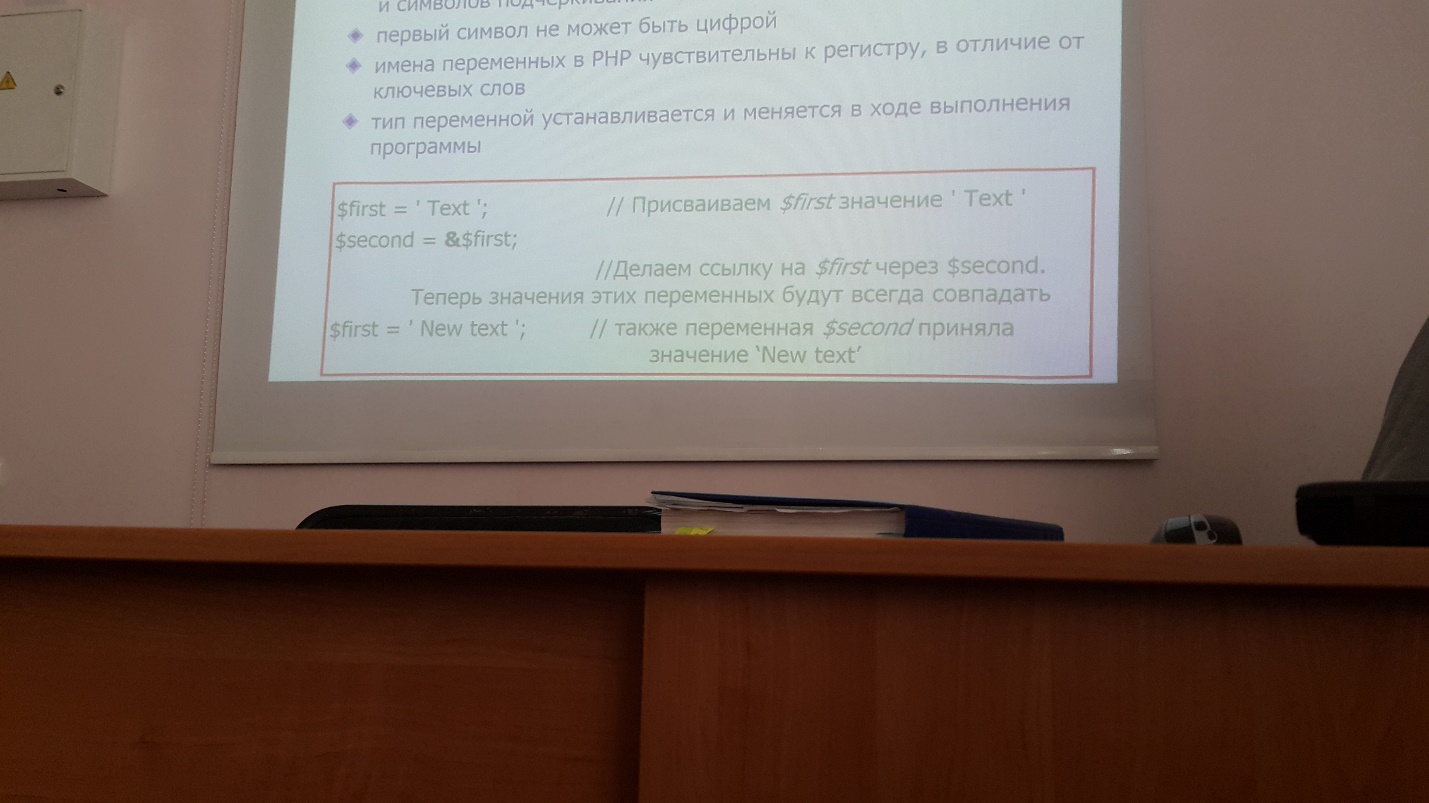
1. После определения константа не может быть изменена.
2. В имени константы обычно используются только заглавные буквы.
3. Пхп имеет ряд предопределенных констант.

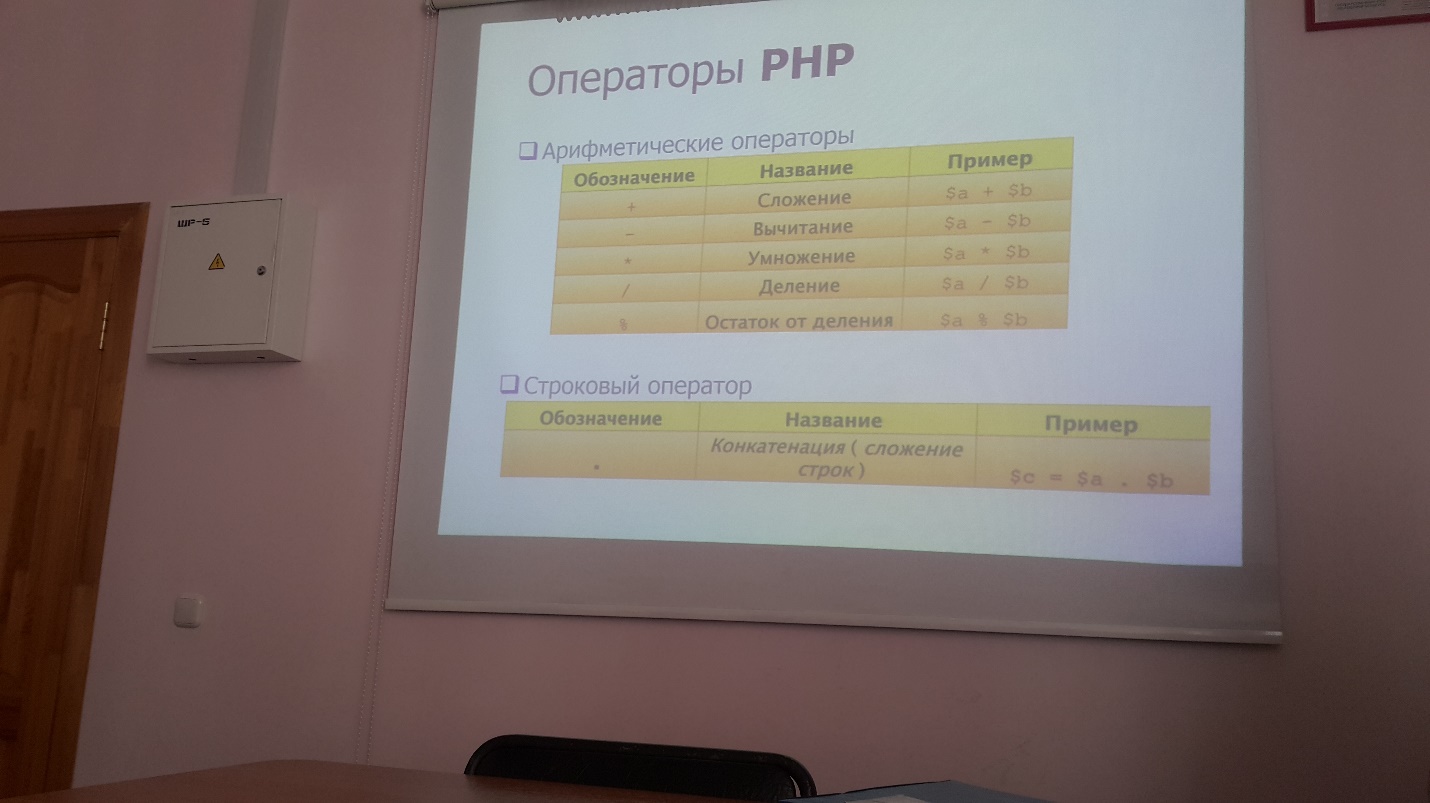
Например:

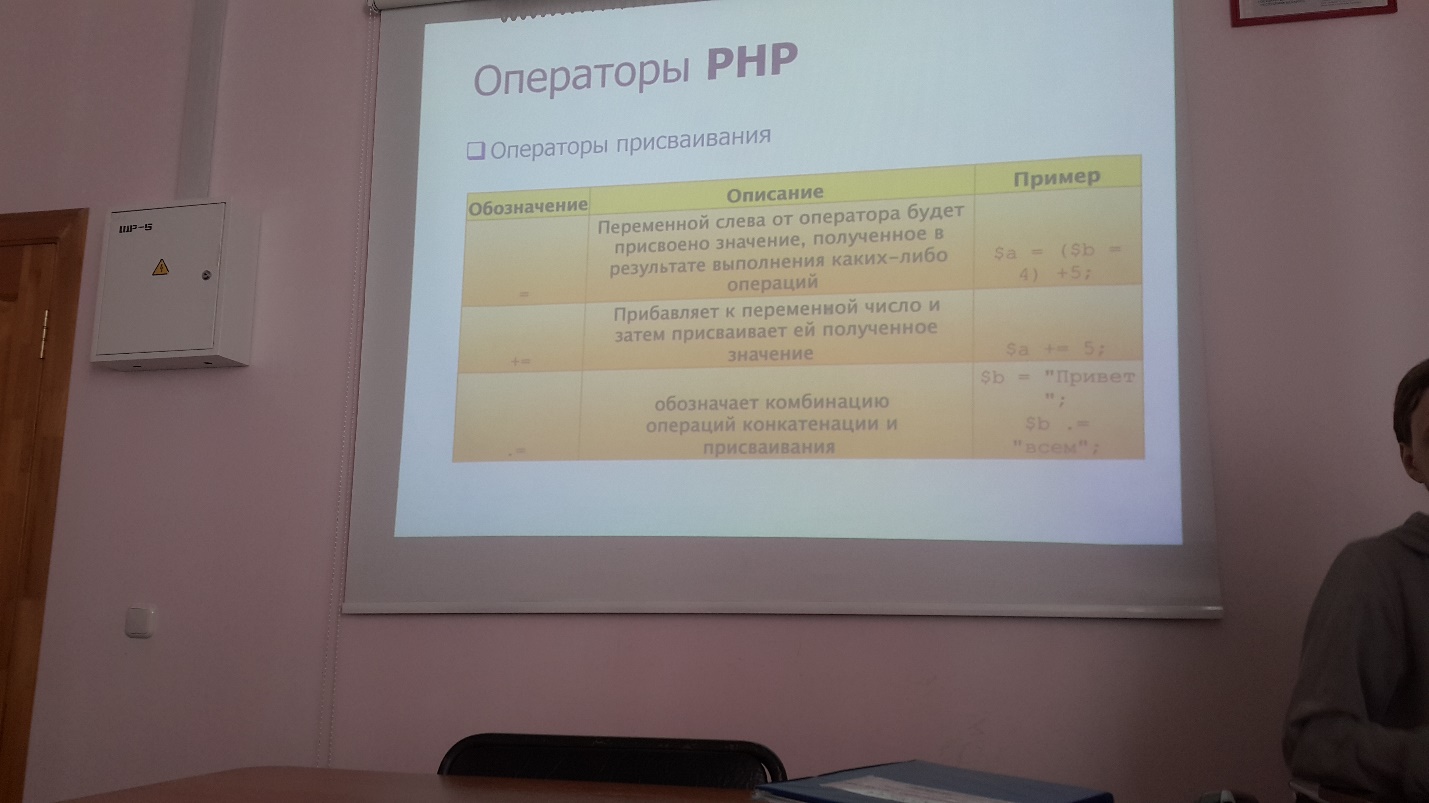
1. \_FILE\_ содержит имя файла, который в данный момент читает пхп
2. \_LINE\_ содержит номер строки этого файла

Переменные пхп

1. Начинаются со знака $
2. Может следовать любое количество буквенно-цифровых символов и символов подчеркивания
3. Первой символ не может быть цифрой
4. Имена переменных в пхп чувствительны к регистру, в отличие от ключевых слов
5. Тип переменной устанавливается и меняется в ходе выполнения программы







Пхп поддерживает восемь типов данных:

