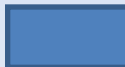


Знакомство с каскадными таблицами стилей.

Оглавление

- ✓ Занятие № 1. Основа ~~Занятие № 1. Основа~~ CSS ~~Занятие № 1. Основа~~
CSS – селекторы
- ✓ Занятие № 2. Оформление текста - работа со шрифтами
- ✓ Занятие № 3. Оформление текста - работа с параметрами абзаца
- ✓ Занятие № 4. Понятие блок, оформление блоков
- ✓ Занятие № 5. Создание каркаса сайта с помощью плавающих элементов
- ✓ Список использованных источников



Основа CSS - селекторы

HTML – основа WEB-страницы, ее каркас, т.е. **ЧТО** должно быть на странице.

Стиль – это указание браузеру как отображать тот или иной элемент, т.е. **КАК** страница должна выглядеть.

Всё, что находится внутри тега `<body>`, браузер будет отображать в соответствии с описанными **СТИЛЯМИ**.

✓ **Вложение**

прописывание в теге с помощью атрибута `style`

✓ **Встраивание**

прописывание внутри тега `Head` с помощью элемента `<style>`

✓ **Связывание**

размещение в отдельном файле `style.css` и прописывание внутри тега `Head` с помощью тега `<link>`



Основа CSS - селекторы

- **Основа – селектор и правило**

(записывается в {})

Правило состоит из свойств и их значений, разделенных «;», свойство и значение разделяются «:».

**Селектор { свойство1:значение;
 свойство2:значение;

 свойствоN:значение;
 }**



Основа CSS - селекторы

Селекторы тегов

```
name{ }
```

позволяет назначить свойства всем тегам одного плана.
name – имя тега

Селекторы идентификаторов

```
#name { }
```

id-селектор позволяет изменить свойства одного элемента.
Самому элементу в html-документе надо дать имя id.
name – имя идентификатора

Селекторы классов

```
.name { }
```

позволяет изменять свойства выбранной группы элементов.
Элементам надо в html-документе дать имя класса.
name – имя класса

- ✓ Групповые селекторы используются для задания одинаковых правил для разных элементов.
- ✓ Записываются селекторы через запятую, могут быть использованы все виды селекторов.



Основа CSS - селекторы

Правила

- Цвет элемента `color: #1250FF;`
- Рамки (с 4 сторон) `border-width: 2px;` - ширина рамки
`border-style: solid | dashed | dotted |` и др.;
- стиль рамки, `solid` – одинарная, `dashed` – пунктирная, `dotted` – в точку, `double` – двойная, `groove` – с тенью наружу, `ridge` – с тенью внутрь, `inset` – левая+верхняя, `outset` – правая+нижняя.
`Border-color: #235687;` - цвет рамки
- Сторона рамки `top | bottom | left | right` – приставки
(например: `border-top-width: 2px;`)
`top` – верхняя, `bottom` – нижняя, `left` – левая, `right` – правая
- Сокращенная запись – толщина стиль цвет. Пример:
`border: 2px solid #235687;`



Основа CSS - селекторы

Задание

- Дан некий файл index.html с текстом, имеющим заголовок и разбитым на абзацы.
- Написать файл style.css, в котором использовать
 - селекторы тегов (`name { }`) для всех абзацев и заголовка – установка цвета текста,
 - селекторы идентификаторов (`#name { }`) для нечетных абзацев – установка пунктирной рамки толщиной в 3 пикселя выбранного цвета,
 - селекторы классов (`.name { }`) для изменения свойств второго абзаца – установка рамки с сокращенной записью.
- Использовать правила для цвета текста и рамок.
- Связать файл index.html с файлом style.css.



Оформление текста - работа со шрифтами

Правила

- Гарнитура Шрифта `font-family:font1, font2, type_font;`

По умолчанию гарнитура шрифта – Times New Roman, `font1` – название основного шрифта, `font2` – название второго шрифта, `type_font` – тип шрифта, `serif` – с засечками, `sans-serif` – без засечек, `monospace` - МОНОШИРНЫЙ

Часто используемые шрифты	Arial, Arial, Helvetica, <i>sans-serif</i>
Arial Black, Arial Black, Gadget, <i>sans-serif serif</i>	Lucida Sans Unicode, Lucida Grande, <i>sans-serif</i>
Comic Sans MS, Comic Sans MS ⁵ , <i>cursive</i>	Palatino Linotype, Book Antiqua ³ , Palatino,
Courier New, Courier New, <i>monospace, serif</i>	Tahoma, Geneva, <i>sans-serif</i>
Georgia ¹ , Georgia, <i>serif</i>	Times New Roman, Times New Roman, Times
Impact, Impact ⁵ , Charcoal ⁶ , <i>sans-serif</i>	Trebuchet MS ¹ , Trebuchet MS, <i>sans-serif</i>
Lucida Console, Monaco ⁵ , <i>monospace</i>	Verdana, Verdana, Geneva, <i>sans-serif</i>



Оформление текста - работа со шрифтами

- **Размер шрифта** `font-size:14px;`
% - **проценты** (100%), em – единицы (1), **px** – **пиксели** (16), pt - пункты (1/72 дюйма) (12),
В скобках указаны базовые размеры
Абсолютный размер - xx-small, x-small, small, **medium**, large, x-large, xx-large,
относительный размер – larger, smaller,
- **Курсив** `font-style:italic;`
Перед отдельными фрагментами прописывается тег логического уровня , которому назначают класс
- **Насыщенность** `font-weight:normal | bold | 100-900`
Normal – нормальная (значение 400), или bold - полужирный шрифт, или числами от 100 до 900 через 100
- **Декор** `text-decoration:underline | overline | line-through | none`
Underline – подчеркнутый, overline – надчеркнутый, line-through – перечеркнутый, none – снятие подчеркивания
- **Интервал между словами** `word-spacing:12px;` (пиксели) (игнорируется, если установлено выравнивание по ширине)
- **Расстояние между буквами** `letter-spacing:2px;`
- **Изменение регистра** `text-transform:uppercase | uppercase | capitalize | capitel;`
uppercase – заглавные буквы (верхний регистр), uppercase - строчные буквы (нижний регистр),
capitalize - каждая первая буква заглавная, capitel – строчные буквы как малые заглавные.



Оформление текста - работа со шрифтами

Задание

- ✓ Дополнить файл `style.css` правилами для заголовка, устанавливающими гарнитуру шрифта из семейства `sans-serif`, размер шрифта 18 пикселей, полужирного начертания, с увеличенным расстоянием между буквами, каждая первая буква заглавная.
- ✓ Дополнить файл `style.css` правилами для всех абзацев, устанавливающими гарнитуру шрифта из семейства `serif`, размер шрифта 14 пикселей, курсивного начертания, с увеличенным расстоянием между словами.
- ✓ Дополнить файл `style.css` правилами для четных абзацев, устанавливающими декор – подчеркивание, изменение регистра на заглавные буквы



Оформление текста - работа с параметрами абзаца

- **Правила**
- **Выравнивание** `text-align:left | center | right | justify;`
- **Красная строка** `text-indent:20px;`
(можно в %, пикселях, единицах)
- **Межстрочный интервал** `line-height:12px | normal;`
Или в px, или множитель от текущего размера шрифта
- **Фоновые цвета и изображения**
Все блоки изначально прозрачные.
- **Фоновый цвет** `Background-color:#5621FF;`
Фоновый цвет распространяется на содержимое и отступы.
Рамка и внешние поля сами по себе, рамку раскрасить можно, поля – нет.
- **Контур для удобства верстки** `outline:1px solid #456879;`
- **Фон-изображение** `background-image:url();`
url - путь относительный (относительно файла style.css или папки css) записывается в скобках
- **Повторение фона** `background-repeat:no-repeat | repeat-x | repeat-y | repeat;`



Оформление текста - работа с параметрами абзаца

Задание

- ✓ Дополнить файл `style.css` правилами для заголовка, устанавливающими выравнивание по центру, установить фоновый цвет.
- ✓ Дополнить файл `style.css` правилами для всех абзацев, устанавливающими выравнивание по ширине, красную строку, межстрочный интервал с множителем 1.5.
- ✓ Фоновое изображение скопировать в папку `image`.
- ✓ Дополнить файл `style.css` правилом для тега `<body>`, устанавливающим фоновое



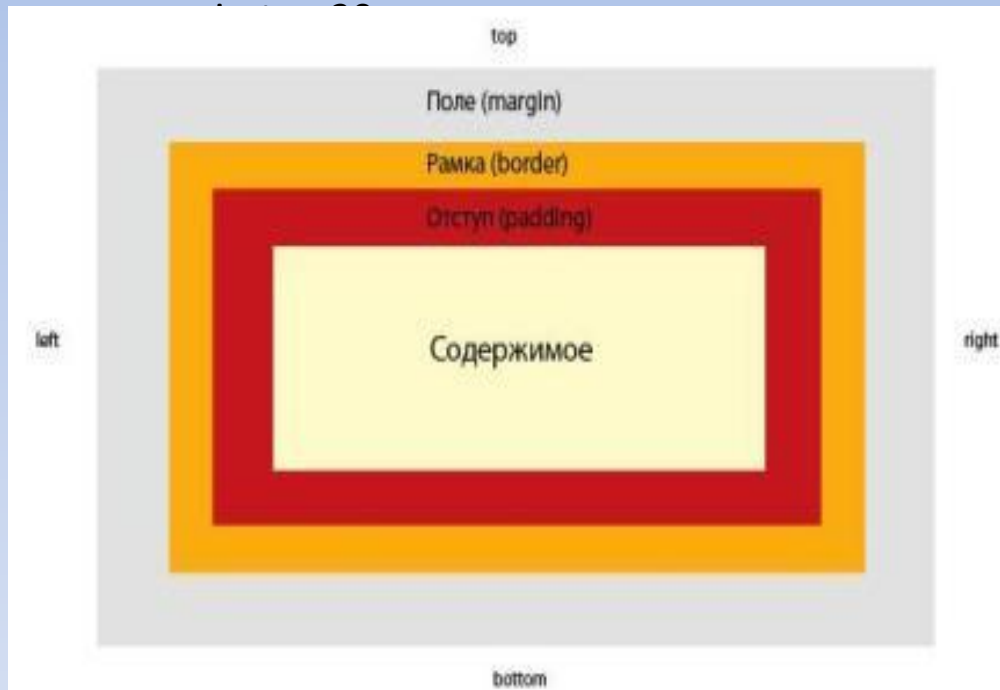
Понятие блок, оформление блоков

- Блочные модели**

Каждый тег для браузера (body – такой же блок, как и другие) – это блок.

Блочная схема: margin (поле), border (рамка), padding (отступ), content (содержимое).

Установка полей:



для остальных полей.

лке.

но.

eight:200px; – высота содержимого.
емых размерах, то:

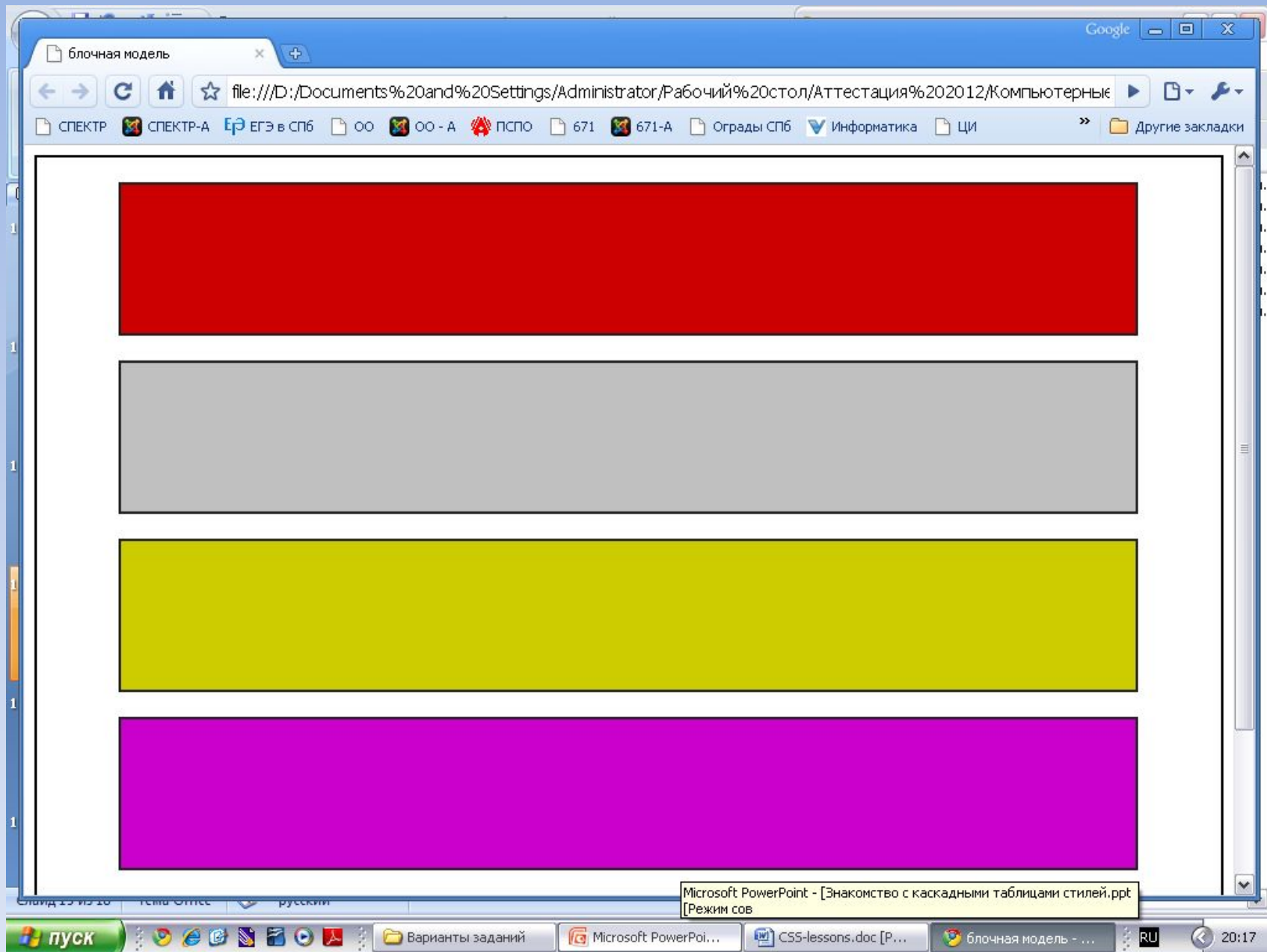
overflow:hidden (спрятать) | auto (полосы прокрутки, если не помещается) | scroll (полосы прокрутки в любом случае) | visible;

Понятие блок, оформление блоков

Задание

- Создать 5 блоков `<div>` `</div>` в html-файле:
 - размер каждого блока по высоте 100px (пока нет заполнения, затем это свойство снимается и блок имеет высоту, занимаемую заполняющей его информацией), по ширине они займут отведенное им место
 - отступы (padding) до рамки 10px,
 - поле (margin) снизу 20px,
 - рамка 4px одинарная черного цвета, цвет заливки каждого блока подобрать самостоятельно.
- Созданные блоки «обернуть» в блок `<div>` `</div>`, для которого задать ширину 800px, выравнивание по центру (auto).
- Все свойства задать в присоединенном css-файле.





Создание каркаса сайта с помощью плавающих элементов

Каркас с «шапкой», левом меню, содержимым и нижней частью страницы

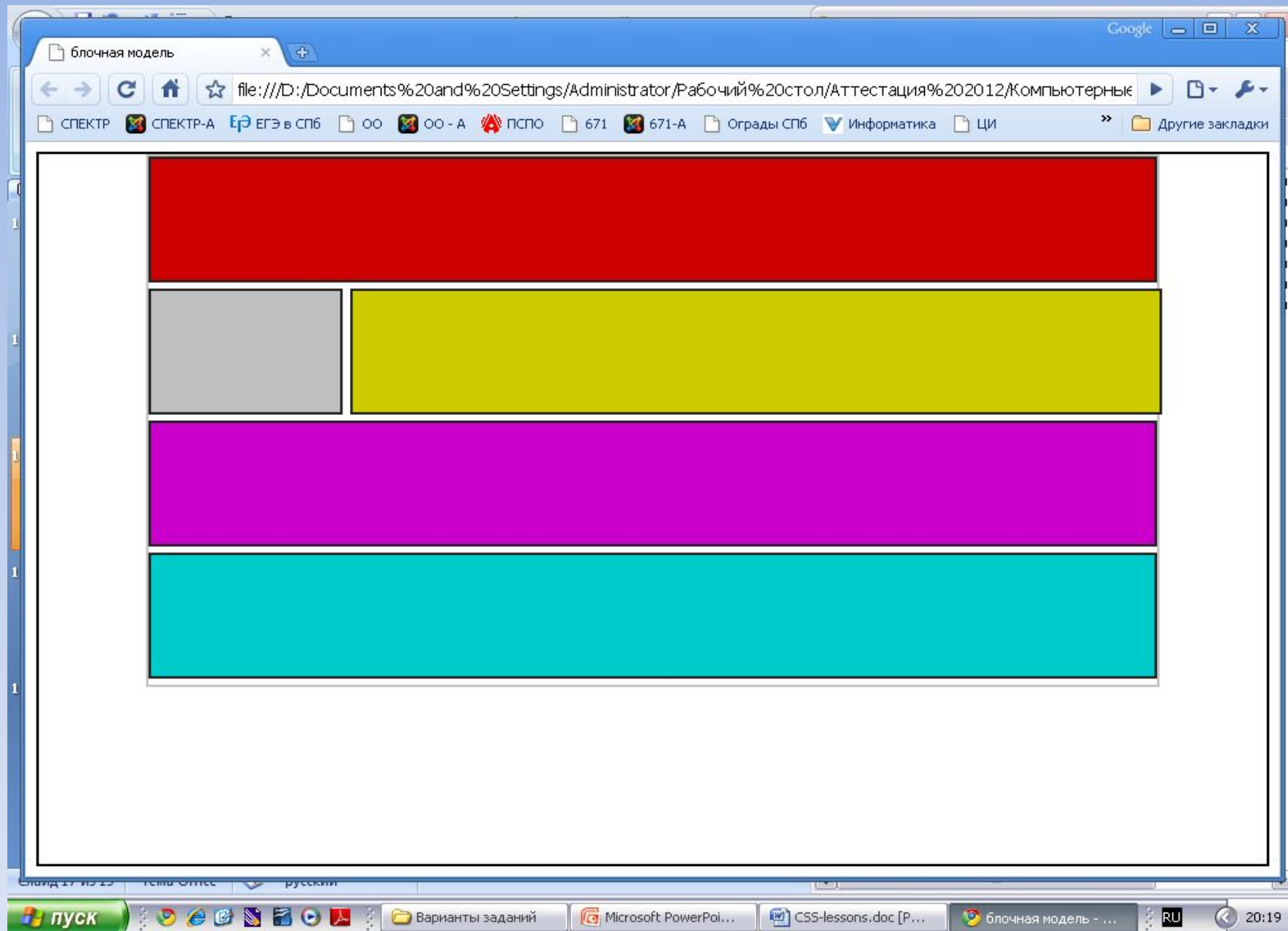
Свойство плавающего элемента –

- ✓ `float:left;` - выравнивание по левому краю
- ✓ `float:right;` - выравнивание по правому краю
- ✓ `clear:both;` - очистка обтекания по двум сторонам (слева и справа).

Для плавающего элемента обязательно надо указывать ширину (`width`) для корректного отображения разными браузерами

Для элемента (левое меню) указывается нужная ширина (например: 150px) и выравнивание по левому краю. Для элемента (содержимое) указывается ширина, равная разности общей ширины с шириной левого меню и небольшого отступа между ними (например: $800 - 150 - 10 = 640\text{px}$), а также левое поле, равное ширине левого меню + небольшой отступ (160px). Перед блоком нижней части страницы надо вставить дополнительный блок, которому задать класс (в html-файле). В css-файле этому классу надо задать свойство очистки обтекания по двум сторонам.





Создание каркаса сайта с помощью плавающих элементов

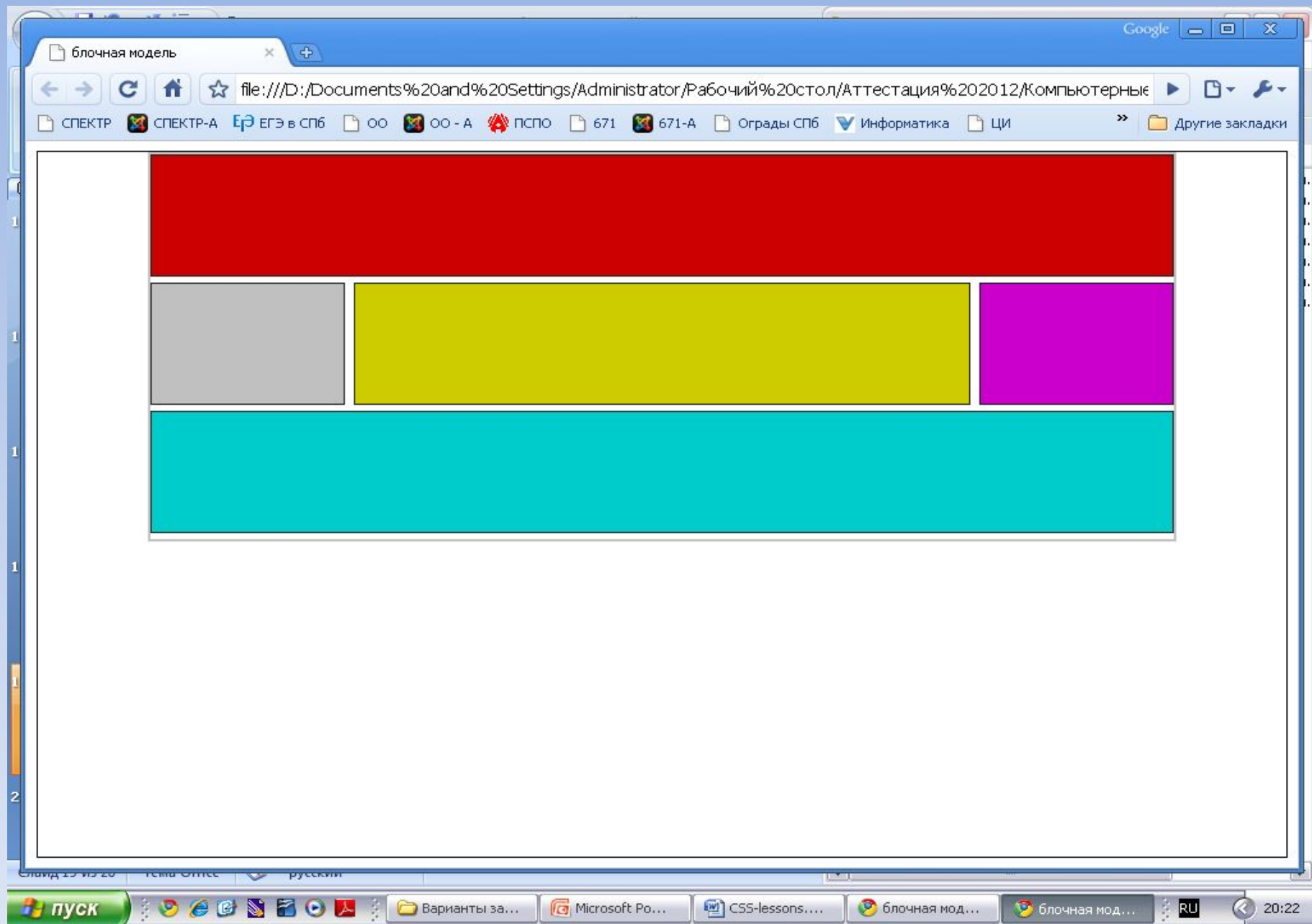
Каркас с «шапкой», левом меню, правом меню, содержимым и нижней частью страницы

- ✓ Для элемента (левое меню) указывается нужная ширина (например: 150px) и выравнивание по левому краю.
- ✓ Для элемента (правое меню) указывается нужная ширина (например: 150px) и выравнивание по правому краю.
- ✓ Для элемента (содержимое) указывается ширина, равная разности общей ширины с шириной левого и правого меню и небольшого отступа между ними (например: $800 - (150 + 150) - (10 + 10) = 480\text{px}$), а также выравнивание по центру.
- ✓ Перед блоком нижней части страницы надо вставить дополнительный блок, которому задать класс (в html-файле). В css-файле этому классу надо задать свойство очистки обтекания по двум сторонам.

Задание

Создать два каркаса для сайтов,
используя вышеописанные правила





Список использованных источников

- <http://ab-w.net/> - самоучитель по сайтостроению
- <http://evgeniyropov.com/> - видеоуроки по HTML и CSS

