



Практическая работа №15

Поля документов. Бегущая строка

Если необходимо убрать поля документа:

```
<Body text="#000000" bgcolor="ffffff" topmargin="0" leftmargin="0">
```

Topmargin - определяет ширину верхнего и нижнего полей документа, **leftmargin** - определяет ширину левого и правого полей документа. Чтобы убрать поля мы приравняли эти параметры к нулю.

Продолжим разговор о специфических тегах, и на этот раз речь пойдет о теге **Marquee** - бегущей строке (текста).

Тег **Marquee** поддерживается только Internet Explorer, но многих людей, начинающих осваивать сайтостроение, это не останавливает, и они с удовольствием его употребляют.

```
<marquee height="10" width="270" bgcolor="#99CCFF">
```

```
Бегущая строка, высота - 10, ширина - 270 пикселей, фон - 99CCFF
</marquee>
```

Как видите, все просто. Все атрибуты вам уже знакомы: **bgcolor** - цвет фона бегущей строки, **height** - высота строки, **width** - ширина строки.

Также для бегущей строки можно задать и другие знакомые вам атрибуты - **hspace**, **vspace** и **align**. Например:

```
<marquee height="10" width="270" loop="2">Бегущая строка</marquee>
```

Параметр **loop** задает сколько раз прокрутится строка, в нашем примере указано **loop="2"** - 2 раза.

Далее будут лишь описания параметров без примеров, подставляйте их сами в пример, и смотрите, что будет получаться.

Direction - задает направление движения бегущей строки - **direction="left"** (right, up, down) - движение влево (вправо, вверх, вниз).

Behavior - поведение строки - **behavior="scroll"** (slide, alternate).

Scroll - обычная прокрутка (можно не прописывать, оно так и есть по умолчанию). **Slide** - прокрутка с остановкой, строка пробегает до края и останавливается. Если одновременно с **behavior="slide"** употребить параметр **loop**, то строка прокрутится установленное количество раз и остановится у края.

Alternate - строка будет двигаться от края к краю.

Scrollamount - скорость движения строки, **scrollamount="1"**. Может принимать значения от 1 до 10.

1 - самое медленное движение, **10** - самое быстрое. Уф! Все :)

Задание :

Создайте 4 варианта бегущей строки, используя параметры приведенные в примере; Сохраните работу под старыми именами в свою папку.



HTML-первые шаги

Практическая работа №1

Стандартный шаблон HTML документа

В этом выпуске:

1. Открыть блокнот, набрать текст

```
<HTML> <html>
<head> <head>
</head> <title>Мой первый шаг
<body> </title>
</head>
<body>
Здравствуйте, это моя
первая страница.
<br>
Добро пожаловать! :)
</body>
</HTML> </body>
```

</html>

2. Сохраним этот документ, присвоив ему имя **index.html**

3. Откроем, допустим, Internet Explorer (не закрывайте блокнот, он нам еще пригодится).

Файл - Открыть - кнопка Обзор - Наш документ (index.html).

Если мы чего-то изменили в нашем *.html документе (в блокноте), то, чтобы посмотреть как это выглядит в Internet Explorer, надо не забывать нажимать в IE (Internet Explorer) кнопку ОБНОВИТЬ (тоже самое касается других браузеров:) Если изменений не видно, то это значит, что вы где-то что-то неправильно написали, или забыли сохранить документ.)

Посмотрим теперь, что у нас получилось, и разберемся как оно так получилось.)

Шаблон	1
Цвет текста и фона	3
Параграфы, линии	4
Заголовки	5
Форматирование текста	6
Картинки на странич-	7
Параметры картинок	8
Гиперссылки	9
Картинка - гиперссыл-	10
Списки	11
Таблица	12
Объединение ячеек	14
Навигационные карты	16
Якоря	19
Бегущая строка	20



Практическая работа №14 Якоря

Первое, что нам нужно усвоить, что html (html) это то, что мы сейчас изучаем. Второе, что нам нужно усвоить: Html не является языком программирования, он предназначен для разметки текстовых документов (т.е. для форматирования текста). То, как будет выглядеть ваш текст, определяют метки (tags или тэг).

Пример тэга: **
** (перенос текста на другую строку, что-то вроде Enter:). Попробуйте между тэгами Body его понатыкать - посмотрите, что получится:)

Т.е. все, что находится между **< и >** - это тэг. Текст, не находящийся между такими скобками **< >** - весь виден при просмотре в браузере. Как мы видим на нашем примере тэгов много и они разные:). Обратите внимание на картинку, она тут не просто так, она показывает нам **обязательные тэги**.

```
<html>
```

Этот тэг должен открывать документ. Если есть открывающий тэг, то должен быть и закрывающий:

```
</html>
```

И так всегда по умолчанию. Некоторые тэги, вроде **
**, не требуют закрывающего тэга. Но вернемся к нашему документу и рисунку.

```
<head> </head> - голова документа  
<body> </body> - тело документа
```

Все тэги, расположенные между **<head> </head>**, это что-то вроде служебной информации (наша служба и опасна и трудна, и на первый взгляд как будто не видна:). Например **<title>** - заголовок. Зачем он? Откройте IE (Internet Explore) с нашим документом и устремите свой взгляд выше всех командных строк (на заголовок окна)... Увидели?:)

Все тэги, расположенные между **<body> </body>** - непосредственное содержание документа. Следующие уроки будут посвящены именно этим тэгам: мы узнаем как менять цвет текста, фона, как делать текст крупнее-мельче, поговорим о картинках, таблицах и многом другом:)

Задание:

Набрать песню из 2-х куплетов и припева

Иногда возникает такая ситуация: нам нужно сделать ссылку не на другой документ, а внутри того же документа - **закладку**, в народе называемую **якорем**. Такая навигация внутри одного и того же документа весьма удобна. Создаваться она может двумя способами.

Способ первый - с помощью атрибута **name** (имя закладки) тега **A**:

```
<H2>Григорий Остер, "Вредные советы.<BR>  
Книга для непослушных детей и их родителей".</H2>  
<A href="#stih1">Ссылка на стих первый</A><BR>  
<A href="#stih2">Ссылка на стих второй</A><BR>  
<A href="#stih3">Ссылка на стих третий</A><BR><BR>  
<PRE>
```

Недавно ученые открыли, что на свете бывают непослушные дети, которые все делают наоборот. Им дают полезный совет: "Умывайтесь по утрам" - они берут и не умываются

```
</PRE>  
<H3><A name="stih1">Стих первый</A></H3>  
<PRE>  
... тра-ля-ля 1...
```

```
</PRE>  
<H3><A name="stih2">Стих второй</A></H3>  
<PRE>  
... тра-ля-ля 2...
```

```
</PRE>  
<H3><A name="stih3">Стих третий</A></H3>  
<PRE>  
... тра-ля-ля 3...
```

Заголовки (стих первый, стих второй, стих третий) в нашем примере мы сделали закладками, используя параметр тега **A** - **name**: **Стих третий**

Затем, без стеснения, мы сделали на них ссылки:

```
<A href="#stih3">Ссылка на стих третий</A>
```

Заметьте - **** - символ решетки (#) перед именем закладки, на которую мы ссылаемся, обязателен. Вы можете сослаться на закладку и из других документов, тогда ссылка будет выглядеть следующим образом:

```
<A href="ancorpri.html#stih3"> Ссылка на стих третий из какого-то другого документа </A> ИЛИ <A href="http://www.mysite.ru/ancorpri.html#stih1"> Ссылка на стих третий из какого-то другого документа </A>
```

Способ второй - есть такой атрибут (параметр) - **id**, который назначает тегу (элементу) уникальное имя в пределах одного документа

```
<H3 id="stih1">Стих первый</H3>
```

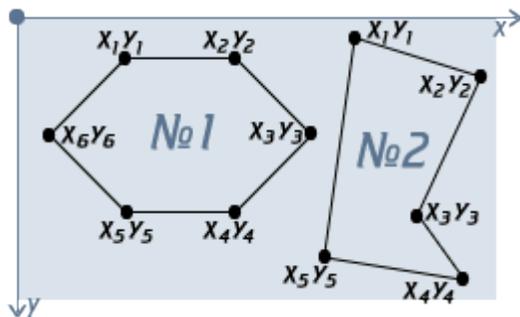
Посмотрите, внешне ничего не изменилось и все действует так же, как и в первом примере, поэтому вы можете использовать спокойно любой из этих двух способов (какой больше нравится).

Также хочу вас предостеречь, никогда не делайте так:

```
<A href="#stih">Ссылка на ...</A><BR>  
...  
<A name="СТИИ">Закладка</A>
```

Если вы, например, напишите имя закладки большими буквами, а ссылаясь укажете это имя маленькими, то браузер может посчитать это за два разных имени (а может и за одно), поэтому во избежание ошибок учитывайте это, и пишите имена в одном регистре.

Многоугольники



Вы указываете точки (координаты углов), они как бы соединяются, и мы можем получить очень разнообразные фигуры (посмотрите на фигуру 2, правда здорово?). Используя `poly`, мы можем делать самые разнообразные области, от скромного треугольника до широчайшей звезды.

Сейчас мы будем работать с фигурой №2

Для начала зададим **тип области**:

```
<map>
<area shape="poly">
</map>
```

Координаты пишутся по следующему принципу:

```
<area shape="poly" coords="x1,y1,x2,y2,...,xN,yN">
```

Расшифровывается это так: "координаты первого угла (x1,y1), координаты второго угла (x2,y2), еще много углов и их координат (...), координаты последнего угла (xN,yN)". Т.е. для нашего пятиугольника запись полностью будет выглядеть так:

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,x4,y4,x5,y5">
```

Теперь подставим реальные значения координат в наш код:

```
<map>
<area shape="poly" coords="168,9,232,29,200,97,223,129,153,119">
</map>
```

Дальше уже прописываем ссылку, имя карты, и привязываем карту к рисунку (это везде по одному и тому же принципу):

```

... Куча текста и всякого содержания, или ничего...
<map name="karta3">
<area href="drugoy_document_3.html" shape="poly" coords="168,9,232,29,200,97,223,129,153,119">
</map>
```

Не забудьте прописать самостоятельно первую область (шестиугольник) - x1=54, y1=20, x2=109, y2=20, x3=147, y3=58, x4=109, y4=96, x5=54, y5=96, x6=16, y6=58.

Несколько нюансов :

1 - Мы можем одновременно использовать разные области, например круг и многоугольник:

```

... Куча текста и всякого содержания, или ничего...
<map name="karta3">
<area href="drugoy_document_3.html" shape="circle" coords="46,48,35,">
<area href="drugoy_document_3.html" shape="poly" coords="168,9,232,29,200,97,223,129,153,119">
</map>
```

2 - Наши области могут пересекаться. Это не смертельно, главное помнить, что в этом случае при нажатии на область пересечения приоритет имеет область, которая указана первой (т.е. пользователь пойдет на страницу, куда она ссылается).

3 - Вы можете не мучаться с картами, а найти на просторах интернета специальную программу, с помощью которой вы без труда сможете карту расчертить, и не прописывать все вручную - это может сэкономить ваше время, силы, нервы.



Практическая работа №2

Цвет текста и фона

Все цвета прописываются таким способом: 16—ричная система счисления (0 1 2 3 4 5 6 7 8 9 A B C D E F) на каждой из 6 позиций любой символ

#RRGGBB (красный-зеленый-синий)

Например :

Черный цвет— 000000 (нет никакого цвета)

белый цвет - fffffff (присутствие всех цветов)

Но вернемся к нашей страничке. Давайте окрасим слова Добро Пожаловать в красный.

```
<font color="#CC0000">Добро пожаловать! :)</font>
```

Теперь попробуйте вместо CC0000 поставить другие значения цветов.

Тэг `` - многофункционален. Им может задаваться не только цвет текста в конкретной части документа, но и размер шрифта, и вид шрифта (Arial), но об этом чуть позже.

Как же еще задавать цвета в документе? - вспомним об открывающем тэге `<body>`:

```
<body>:
<body text="#336699">
```

Это значит, что весь текст страницы будет **синим**, кроме текста, для которого мы специально прописали `` (если цвет в `<body>` не задавать, то по умолчанию он будет черным).

Цвет фона страницы устанавливается в уже нам знакомом тэге `<body>`:

```
<body bgcolor="#000000">
```

Для наглядности я прописала черный цвет, вы же свой документ можете раскрасить любым другим (если цвет в `<body>` не указывать, то по умолчанию он будет белым, хотя иногда цветом по умолчанию может являться любой другой, кроме белого, поэтому лучше уж прописывайте цвет фона в `body`).

Обратите внимание: мы одновременно можем прописать в тэге `<body>` и цвет текста в документе, и цвет фона (это на всякий случай, если вы еще не поняли:).

```
<body text="#336699 bgcolor="#000000">
```

Задание:

Подобрать цвет фона станицы, установить цвет текста станицы, выделить куплеты и припевы разными цветами



Практическая работа №3 Параграфы, линии

Параграфы вводятся тэгом:

```
<p></p>
```

С помощью параграфов мы можем **центрировать** текст:

```
<p align="center">текст</p>
```

С помощью параграфов мы можем выровнять текст **по левому краю**:

```
<p align="left">текст</p>
```

Или **по правому краю** документа:

```
<p align="right">текст</p>
```

Выравнивание текста **по обоим краям** документа

```
<p align="justify">текст</p>
```

Линия задается тэгом <Hr> и не требует закрывающего тэга:

У линии есть много разных параметров:

- (1) <Hr align="right"> (center или left)
- (2) <Hr width="30%"> (ширина линии в процентах/пикселях)
- (3) <Hr size="6"> (толщина линии)
- (4) <Hr NoShade> (отмена объемности)
- (5) <Hr color="cc0000"> (цвет линии, только в IE)

Естественно, эти параметры могут употребляться одновременно.

В нашем примере у прямоугольника №1 координаты такие:

```
x1=25, y1=36, x2=114, y2=98
```

Значит код будет выглядеть следующим образом:

```
<map>
<area shape="rect" coords="25,36,114,98">
</map>
```

Теперь пропишем, куда будет ссылаться наша область, для этого нам понадобится уже знакомый нам параметр href:

```
<map>
<area href="document.html" shape="rect" coords="25,36,114,98">
</map>
```

Все бы хорошо, но этого все еще не достаточно, чтобы картинка стала ссылкой, нам нужно еще указать имя карты и связать ее с картинкой.

У тэга **Map** есть параметр **name - имя карты**, давайте назовем карту - karta1:

```
<map name="karta1">
<area href="document.html" shape="rect" coords="25,36,114,98">
</map>
```

Для того, чтобы связать карту с картинкой, надо использовать атрибут **usemap="#имя_карты"** для картинки:

```

```

... Куча текста и всякого содержания, или ничего...

```
<map name="karta1">
<area href="document.html" shape="rect" coords="25,36,114,98">
</map>
```

Теперь сами пропишите Area для прямоугольника №2 для тренировки (x1=153, y1=11, x2=219, y2=127)

Для создания **круглой области** нам нужны будут координаты ее центра (x и y) и длина радиуса (R) в пикселях. Т.е. порядок записи следующий:

```
<area shape="circle" coords="x,y,R">
```

Давайте будет опять работать с геометрической областью №1. В нашем случае координаты для маленького круга будут такие: x=46, y=48; а длина радиуса - R=35. Запишем:

```
<map>
<area shape="circle" coords="46,48,35">
</map>
```

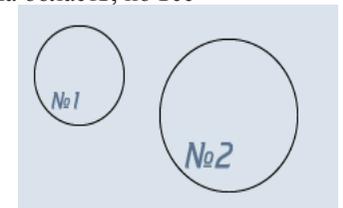
Теперь, когда самое главное записано, пропишем имя карты, куда она ссылается и привяжем карту к рисунку. Кстати, для карты мы можем прописать ALT для каждой области, сейчас у нас только одна область, но все равно, пропишем для нее ALT:

```

... Куча текста и всякого содержания,
или ничего...
<map name="karta2">
<area href="drugoy_document_2.html" shape="circle"
coords="46,48,35" alt="маленький круг">
</map>
```

Теперь при наведении на область №1 будет всплывать подсказка, что же это такое.

Теперь вы самостоятельно сделаете ссылкой второй круг (x=158, y=75, R=53), чтобы закрепить пройденное.



Задание:

Наберите страницу текста;

Примените 4 способа выравнивания абзацев;

Разделите абзацы разными линиями.

Сохраните работу под именем **list.html** в свою папку.

Практическая работа №13

Навигационные карты

Можно сделать так, чтобы при нажатии на разные области (части) одной и той же картинки, вы попадали на разные страницы, это называется - **навигационная карта**. Навигационные карты задаются тэгом **Map**. Тэг Map включает себя тэг(и) **Area**, которые определяют геометрические области внутри карты (в нашем примере это будут прямоугольники) и ссылки, связанные с каждой областью (т.е. куда вы попадете при нажатии на какую-либо часть нашей карты).

```
<map>
<area ...>
<area ...>
...
<area ...>
</map>
```

Для начала нам нужна картинка. Возьмем такую:



Картинка простая - всего лишь голубой прямоугольник (240x140 пикселей). Представьте, что прямоугольники №1 и №2, нарисованные на нем, - это изображения каких-либо кнопок, а голубое - какой-то сложный фон. Чтобы не создавать сложную таблицу, вы решили не резать картинку на много частей и не вычленять кнопки, вы решили поступить проще - создать навигационную карту, где области №1 и №2 (прямоугольники) будут ссылками.

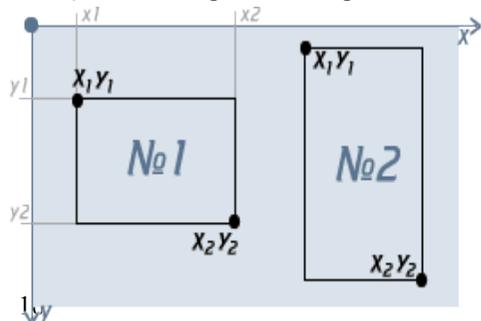
Итак, мы уже знаем, что геометрические области и то, куда пользователь попадет при нажатии на них, определяет тэг **Area**, естественно, при помощи своих параметров.

Это параметры **shape** и **coords**.

Параметр **shape** - определяет форму области (будет ли она прямоугольником (shape="rect"), кругом (shape="circle") или многоугольником (shape="poly")). Сейчас мы работаем с прямоугольной областью, поэтому:

```
<map>
<area shape="rect">
</map>
```

Параметр **coords** - определяет координаты (положение нашей геометрической формы). Число координат и порядок их значений зависят от выбранной нами формы.



Отсчет ведется от левого верхнего угла картинки, считайте его началом координат (0;0). Если мы работаем с прямоугольной областью, то нам нужны координаты верхнего-левого и нижнего-правого углов области.

Порядок записи координат для параметра coords следующий:

```
<area shape="rect" coords="x1, y1, x2, y2">
```



Практическая работа №4

Заголовки

Итак, мы уже немного умеем форматировать текст, но посмотрите на наш документ. Мне кажется, что фраза "Здравствуйте, это моя первая страница" так и просится, чтобы ее выделили. Для этого можно использовать заголовки:

```
<H1> текст </H1>
```

```
<H2> текст </H2>
```

```
<H3> текст </H3>
```

```
<H4> текст </H4>
```

```
<H5> текст </H5>
```

```
<H6> текст </H6>
```

Пришлось немного пожертвовать красотой, чтобы сразу наглядно все продемонстрировать. Зато всем видно и то, что существуют шесть уровней заголовков, и то, что соответственно буквы у каждого разного размера, и то, что тут у нас такая же ситуация как с параграфами (принудительный перенос строки). Да, заметьте, заголовок выделяется жирным текстом, это тоже одно из его свойств, очень удобно!) Заголовки предназначены для выделения небольшой части текста (строки, фразы), НО, если вы хотите выделить большой фрагмент текста, или только одно слово, при этом без переноса строки, то как быть? С заголовками такое не пройдет, поэтому вспомним о тэге ``:

```
<font size="+4"> ТЕКСТ </font>
```

```
<font size="+3"> ТЕКСТ </font>
```

```
<font size="+2"> ТЕКСТ </font>
```

```
<font size="+1"> текст </font>
```

```
<font size="+0"> текст </font>
```

```
<font size="-1"> текст </font>
```

```
<font size="-2"> текст </font>
```

Опять пришлось пожертвовать красотой:) Итак, параметр size задает размер шрифта, но, в отличие от заголовков, текст не выделяется жирным шрифтом (о курсиве, жирном шрифте и т.п. поговорим позже) и нет принудительного переноса.

Стандартный size (по умолчанию) ="+0".

Задание:

Дополните свою страницу заголовками;

Измените размеры букв разных абзацев;

Сохраните работу **list.html** в свою папку.

Практическая работа №5

Курсив, подчеркнутый текст,
полуужирный текст и фиксированный текст, спецсимволы

`` Полуужирный текст ``

`<i>` Наклонный текст (курсив) `</i>`

`<u>` Подчеркнутый текст `</u>`

Шрифт с символами одинаковой ширины.

`<tt>` fixed - моноширинный шрифт `</tt>`

Big представляется крупным шрифтом,

Small малым шрифтом относительно основного текста:

`<small>` Малый `</small>` Нормальный текст `<big>` Большой `</big>`

Тэги **Strike** и **S** представляют текст перечеркнутым шрифтом:

`<strike>` Перечеркнутый `</strike>` `<s>` Перечеркнутый `</s>`

Тэги **Sup** и **Sub** - определяют верхний и нижний индексы. **Sup** - верхний, **Sub** - нижний. Где они могут пригодиться? Ну, например при написании какой-нибудь формулы - H₂O (все, что мы выносим со школьной скамьи:).

Верхний индекс `^{`^{sup}`}`

Нижний индекс `_{`_{sub}`}`

Фиксированный текст

`<pre>` текст (куча пробелов) текст `</pre>`

Он не только фиксированный, но еще и выводится с точностью до пробела так, как вы его набили в блокноте, сей странный тэг может быть полезен, допустим, для форматирования стихотворений.

Запомните, к одному фрагменту текста может применяться сразу несколько тэгов:

`<tt><i> текст </i></tt>`

Хочу также упомянуть об одном интересном атрибуте тэга ``:

`` текст (шрифт Arial)``

Спецсимволы. Они не так часто употребляются, но тем не менее они необходимы:

(1) - `<`; - `<`

(2) - `>`; - `>`

(3) - `"`; - `"`

(4) - ` `; - пробел

(5) - `&`; - `&`

Т.к. все заключенное между `<` и `>` браузер воспринимает как тэг, то, чтобы ввести символ скобки в текст для него придумали спецсимвол (1 или 2). Кавычка (3). Символ пробела (4). Но тут существует одно **но**: (А. С. Пушкин - не должно переноситься) слово1 слово2 слово3 = слово1 слово2 слово3

Ни одно из этих слов не перенесется на другую строку без двух других, и хотя посетитель увидит обычную фразу, где между словами стоит пробел, но по идее это же одно длинное слово, а не три, т.к. ` `; - **неделимый символ пробела**

И последний символ `&`, т.к. он несчастный используется для написания спецсимволов, то для него добрые люди тоже придумали спецсимвол `&`;

Задание:

Примените к своей страничке **list.html** разные способы форматирования текста;

Вставьте специальные символы

Сохраните работу под старым именем в свою папку.

Можно избавиться от пространства между ячейками таблицы:

1x1	1x2
2x1	2x2

cellspacing=0

1x1	1x2
2x1	2x2

cellspacing=5

Такая красота достигается с помощью атрибута **cellspacing**, равного нулю:

`<table cellspacing=0>` Можно наоборот увеличить пространство между ячейками.

Как видите, и то, и другое мы можем использовать как дизайнерский прием.

Атрибут **cellpadding** добавляет свободное пространство между содержимым ячейки и ее границами. Чтобы было видно нагляднее я для начала прижимаем текст ячейки первого ряда кверху, в нижнего - к низу, используя атрибут **valign**:

1x1	1x2
2x1	2x2

cellpadding=5

1x1	1x2
2x1	2x2

cellpadding=5

Теперь зададим атрибут **cellpadding=5**:

Вот теперь всем должно стать понятно, что такое пространство между содержимым ячейки и ее границами, и что делает тэг **cellpadding**. Сравним получившееся с предыдущей таблицей: у нас добавилось пространство по бокам, снизу и сверху, причем это пространство никакими картинками и текстом заполнено быть не может, как и пространство между ячейками, когда мы задаем **cellspacing**.

Код для последнего варианта таблицы:

```
<table cellpadding=5>
<tr>
<td height="35" bgcolor="#FFCC33" colspan="2" valign="top"> <center>1x1</center> </td>
<td width="50" bgcolor="#336699" rowspan="2" valign="top"> <center>1x2</center> </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699" valign="bottom"> <center>2x1</center> </td>
<td width="50" bgcolor="#FFCC33" valign="bottom"> <center>2x2</center> </td>
</tr>
</table>
```

Задание:

Создайте 2 таблицы расписание занятий в кружках, используя приведенные примеры с объединениями; Напишите код для такой таблицы:

Используйте разные варианты пространства между ячейками, примените к таблицам разные рамки.

Сохраните работу под старыми именами в свою папку.

1x1	1x2
2x1	2x2

Практическая работа №7

Форматирование картинок

Параметр align есть и у картинок:

```

```

Это означает, что картинка будет прижата к левому краю экрана, а текст будет обтекать ее справа. Чтобы сделать наоборот (картинка справа, текст слева) надо прописать right:

```

```

Но это не все: текст может располагаться внизу картинки (это по умолчанию) - (1), по середине - (2), и сверху - (3):

```
(1) - 
```

```
(2) - 
```

```
(3) - 
```

Кроме параметра align существует еще несколько параметров:

```
(1) - 
```

```
(2) - 
```

```
(3) - 
```

```
(4) - 
```

```
(5) - 
```

```
(6) - 
```

Теперь последуют объяснения по пунктам.

(1) - параметр **vspace** - задает расстояние между текстом и рисунком (по вертикали).

(2) - параметр **hspace** - тоже задает расстояние между текстом и рисунком, но по горизонтали

(3) - параметр **alt** - краткое описание картинки. Если навести курсором мыши на рисунок, и так подержать его (курсor) несколько секунд выскочит описание картинки.

(4) - параметр **width** - ширина самой картинки (в пикселях). Если ширину не задавать специально, то по умолчанию она будет равна реальной ширине картинки (а так вы можете ее сделать или уже или шире).

(5) - параметр **height** - высота самой картинки (тоже в пикселях).

Так же как в случае с width высоту (height) картинки можно и не задавать.

(6) - параметр **border** - рамка вокруг самой картинки (в пикселях). Можно не задавать.

Все параметры могут употребляться одновременно друг с другом.

Введем следующие параметры для нашей картинки:

```

```

Картинку можно сделать фоном документа.

Тогда это прописывается в открывающем тэге боди:

```
<body text="#336699" bgcolor="#000000" background="photo.jpg">
```

Параметр Background и указывает на то, где лежит фоновая картинка, в нашем примере он указывает на то, что наша фоновая картинка лежит в той же директории (папке), что и документ.

Но зачем оставлять параметр **bgcolor**, если есть background? А вдруг фоновая картинка не загрузится (представьте, такое может быть), тогда сами поймете зачем.

Задание:

Примените разные способы форматирования картинок на странице

8 index.html; Сохраните работу под старым именем в вашу папку.

```
<table>
<tr>
<td height="35" width="50" bgcolor="#FFCC33"> 1x1 </td>
<td width="50" bgcolor="#336699"> 1x2 </td>
<td width="50" bgcolor="#FFCC33"> 1x3 </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699"> 2x1 </td>
<td width="50" bgcolor="#FFCC33"> 2x2 </td>
<td width="50" bgcolor="#336699"> 2x3 </td>
</tr>
</table>
```

Теперь нам осталось лишь выровнять содержимое внутри таблицы:

```
<table>
<tr>
<td height="35" width="50" bgcolor="#FFCC33"> <center> 1x1 </center> </td>
<td width="50" bgcolor="#336699"> <center> 1x2 </center> </td>
<td width="50" bgcolor="#FFCC33"> <center>1x3 </center> </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699"> <center> 2x1 </center> </td>
<td width="50" bgcolor="#FFCC33"> <center> 2x2 </center> </td>
<td width="50" bgcolor="#336699"> <center> 2x3 </center> </td>
</tr>
```

В каждой ячейке (столбце) могут находиться и картинки, и текст, и даже таблицы (в этом случае они называются - **вложенные таблицы**). Тэги, которые мы применяем

Вертикальное выравнивание содержимого таблицы, т.е. как можно сделать так, чтобы содержимое ячейки не только располагалось ровно посередине ее (как по умолчанию), а еще сверху или внизу. Вертикальное выравнивание задается следующим атрибутом - **valign="middle"** (top, bottom) - содержимое конкретной ячейки будет находиться в середине ячейки (наверху или внизу):

Вот вам и наглядный пример, а теперь как это пишется:

```
<table>
<tr>
<td height="35" width="50" bgcolor="#FFCC33"
valign="top"> <center>1x1</center> </td>
<td width="50" bgcolor="#336699"> <center>1x2</center> </td>
<td width="50" bgcolor="#FFCC33" valign="bottom"> <center>1x3</center> </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699" valign="bottom"> <center>2x1</center> </td>
<td width="50" bgcolor="#FFCC33"> <center>2x2</center> </td>
<td width="50" bgcolor="#336699" valign="top"> <center>2x3</center> </td>
</tr>
</table>
```

1x1	1x2	1x3
2x1	2x2	2x3

Задание:

Создайте таблицу расписание уроков на одной из ваших страниц;

Практическая работа №11

Таблицы

Таблица задается тэгом:

```
<table></table>
```

Увы, это далеко не все: таблица состоит из строк и столбцов (ячеек), поэтому нам надо еще и указать их.

`<tr></tr>` - строка таблицы

`<td></td>` - столбец (ячейка) таблицы

Итак, перед вами таблица из двух строк и трех столбцов (ячеек). Для наглядности я выделила ячейки таблицы разными цветами. Границы таблицы не заданы, поэтому вы их не видите.

Сначала задаем строки. В нашем примере их две.

```
<table>
<tr></tr>
<tr></tr>
</table>
```

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

Теперь в каждой строке зададим по три столбца (ячейки):

Итак, теперь нам надо заполнить получившийся каркас:

```
<table>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</table>
```

Увы, фона (голубенького или желтенького, как в примере) еще не видно. Фон задается параметром `bgcolor="цвет_фона"`. Фон можно задать для таблицы в целом, для ряда, для столбца (в пределе одного ряда). В нашем случае мы задаем фон для каждого столбца.

```
<table>
<tr>
<td bgcolor="#FFCC33">1x1</td>
<td bgcolor="#336699">1x2</td>
<td bgcolor="#FFCC33">1x3</td>
</tr>
<tr>
<td bgcolor="#336699">2x1</td>
<td bgcolor="#FFCC33">2x2</td>
<td bgcolor="#336699">2x3</td>
</tr>
</table>
```

В ТАБЛИЦЕ существуют параметры **height** и **width** - вы можете их задать для всей таблицы, для одного ряда, для ячейки (столбца). Вспомним, что высота и ширина могут задаваться как в пикселях, так и процентах. В нашем случае мы зададим ширину и высоту в пикселях для столбцов (ячеек).

Практическая работа №8

Гиперссылки на странице

Наш сайт может состоять из нескольких документов. Один из них главный (index.html или main.html) - он открывается первым и должен обязательно лежать на вашем сайте в интернете.

Остальные документы вы можете называть как угодно (photos.html, about_me.html, my_pets.html, friends.html, gh516hgd.html). Они все могут лежать в одной директории (папке), а могут в разных.

Ссылкой на эти другие документы (части нашей странички) может быть текст (фраза, слово), а может быть и картинка.

Для начала проверим наличие в нашей папке документа list.html, в той же директории (папке), где находится наш главный документ index.html. Тогда мы можем без зазрения совести фразу "прочитать мой рассказ" сделать ссылкой на list.html:

```
<a href="list.html">прочитать мой рассказ</a>
```

Тэг `<a>` делает ссылкой заключенную в него картинку или фразу (текст).

Принципы прописывания пути здесь такие же как в случае с картинками:

(1) - `мои фотографии`

(2) - `мои фотографии`

(3) - `мои фотографии`

В случае (1) документ лежит в той же директории (папке), что и документ, в котором мы ссылаемся на prf.html, в случае (2) документ лежит в поддиректории /photos, в случае (3) мы ссылаемся на сайт http://www.homepage.ru, где лежит нужный нам документ. Как видите все просто.

Но есть один фактор, который нам нужно учесть: **цвет ссылки** нужно прописывать, а иначе по умолчанию он будет просто безобразным, поэтому вспомним об открывающем тэге боди:

```
<body text="#336699" bgcolor="#000000" link="#339999"
alink="#339999" vlink="#339999">
```

Разберемся что к чему: **link** - цвет ссылки, **alink** - цвет активной ссылки (нажатой), **vlink** - цвет уже посещенной ссылки. В нашем примере цвета одинаковые, но они могут быть разными, также не карается законом, если цвет ссылки такой же как цвет текста документа (ссылка же выделяется чертой).

Ссылка на ваш почтовый ящик прописывается так:

```
<a href="mailto:pochta@mail.ru"> pochta@mail.ru - пишите письма </a>
```

Запомните это раз и навсегда и не пытайтесь ее прописать по другому. Ссылка на документ, картинку или файл - одно дело, а ссылка на почтовый ящик - другое (да, но цвета всех ссылок в документе одинаковые:).

Задание:

Создайте переходы между страницами index.html и list.html

Сохраните свои страницы под старыми именами в свою папку. 9

Практическая работа №9

Картинка как гиперссылка

Ссылкой может быть и картинка. Принцип ссылки тот же, что и в случае с текстом, только между тэгами вставляется не текст, а картинка:

```
<a href="prf.html">  </a>
```

Эта картинка будет ссылкой на страничку с фотографиями. Вокруг картинки появится рамочка. Эту рамочку можно оставить, если вам она нравится, а можно убрать, если вы зададите картинке параметр **border="0"**

```

```

Если вы уже достаточно попутешествовали по Интернету, то вы должны были заметить, что ссылка может быть не только на документ с расширением *.html, но и на многие другие (*.doc, *.mp3, *.jpg, *.gif, *.txt, *.zip, *.exe и т.д.) Как это делается? Да все по тому же принципу:

```
<a href="http://www.melody.ru/music.mp3"> скачать песню </a>
```

Если адрес указан таким способом:

```
<a href="http://www.melody.ru/"> музыкальный сайт </a>
```

То это означает, что в указанном каталоге есть файл вроде index.html, который загрузится по умолчанию.

Но поговорим особо о картинках, многие из вас, наверное, сталкивались с таким явлением: когда жмешь на маленькую картинку, то загружается большая в том же или в новом окне. Как это делается? Тут никаких хитростей: делаем ссылкой картинку (допустим small.jpg) и ссылаемся на другую картинку (допустим big.jpg):

```
<a href="big.jpg">  </a>
```

В этом случае большая картинка откроется в том же окне. Но как же сделать так, чтобы картинка (или любой другой файл-документ) открылась в новом окне? Дело в том, что у тэга <a> есть параметр **target**:

```
<a href="big.jpg" target="_blank">  </a>
```

Итак, target="_blank" - указывает на то, что документ (картинка в нашем случае), на

Задание:

Создайте рисунки для кнопок (можно воспользоваться программами Paint, Flash, PhotoShop) и сохраните их с расширением *.gif. Вставьте созданные кнопки на ваши станицы ;

10 Сохраните работы под старыми именами в свою папку.

Практическая работа №10

Списки

Что такое списки? Это лучше продемонстрировать на наглядном примере:

- Текст один
- Текст другой
- Текст сякой
- Текст эдакий
- Скорее всего вы с этим сталкивались. Все это удобно в определенных ситуациях. Теперь разберемся, какими же тэгами обозначаются списки:

```
<li>Текст один</li>
<li>Текст другой</li>
<li>Текст сякой</li>
<li>Текст эдакий</li>
```

Да-да, все опять оказалось очень простым:) Именно тэгом они и задаются. Возникает вопрос: можно ли помимо закрашенного кружочка прописать еще чего-нибудь подобное?

- У тэга есть параметр **type**:

По умолчанию -

(1) <li type="disk"> ,

Полый кружок -

(2) <li type="circle"> ,

Не будь таким квадратным -

(3) <li type="square"> .

- Это один из вариантов. Если вы пользовались текстовыми редакторами (допустим Вордом), то вы знаете, что такое Табуляция (это когда вы жмете на кнопку Tab и весь текст сдвигается вправо). Похожая функция есть и в HTML.

Маркированный список

```
<ul>
<li>Текст один</li>
<li>Текст другой</li>
<li>Текст сякой</li>
<li>Текст эдакий</li>
</ul>
```

Нумерованный список

```
<ol type="A">
<li>Текст один</li>
<li>Текст другой</li>
<li>Текст сякой</li>
<li>Текст эдакий</li>
</ol>
```

Варианты параметра нумерованного списка:

```
Type = "A"
Type = "I"
Type = "a"
Type = "i"
```

Задание:

Создайте на любой вашей страничке списки изучаемых вами предметов

Оформите списки разными способами;

Сохраните работу под старыми именами в свою папку.