

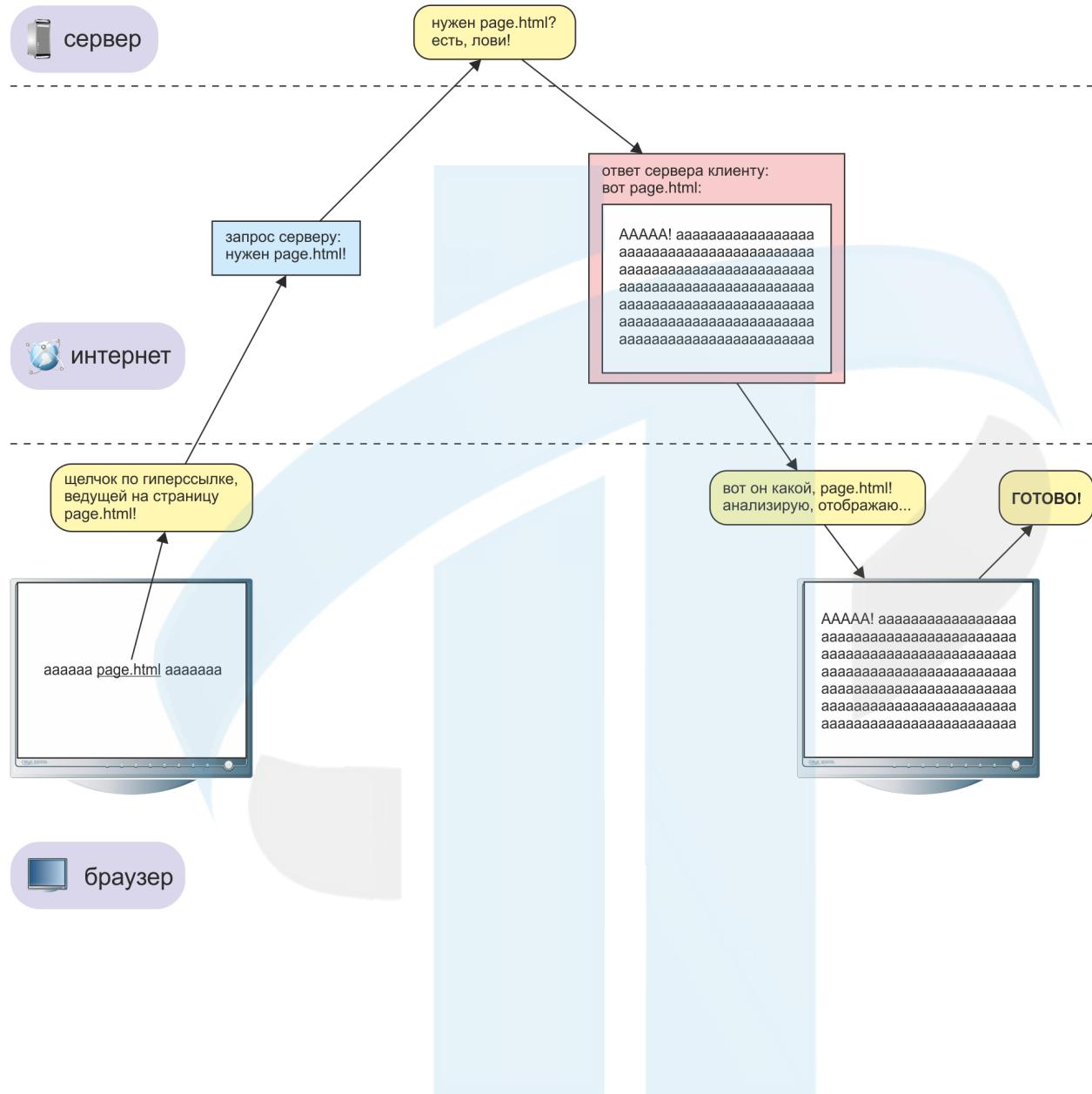
ОГЛАВЛЕНИЕ

Тема 1. Введение в HTML. Ссылки	3
(1.10) Порядок взаимодействия браузера с сервером	3
(1.45) Первые теги HTML	4
(1.55) Первые атрибуты тегов HTML	7
(1.75) Специальные символы HTML	8
(1.80) Структура URI	12
(1.90) Гиперссылки	13
Домашнее задание №1а	14
Домашнее задание №1б	14
Тема 2. Изображения. Введение в CSS	15
(2.20) Изображения	15
(2.35) Форматы изображений	16
(2.38) Проблема окантовки изображений	17
(2.40) Порядок взаимодействия браузера с сервером, если на странице используются изображения	18
(2.45) Кэширование загруженных изображений браузером	19
(2.75) Первые стилевые свойства	23
Домашнее задание №2	26
Тема 3. Цвет. Шрифт. Границы. Размеры	27
(3.20) Стилевое свойство для указания цвета текста	27
(3.25) Цветовой охват	27
(3.30) Именованные цвета	28
(3.35) Единицы измерения	29
(3.45) Стилевые свойства для управления шрифтом	30
(3.50) Имя шрифта	33
(3.52) Все реквизиты шрифта сразу	34
(3.55) Внешние шрифты	34
(3.57) Порядок взаимодействия браузера с сервером, если на странице используются изображения и шрифтовой файл	35
(3.70) Стилевые свойства для указания вида рамки	36
(3.75) Управление размерами	40
Домашнее задание №3	41
Тема 4. Отступы. Фон. Таблицы	42
(4.10) Поля и отступы	42
(4.20) Выравнивание содержимого контейнера	45
(4.25) Управление пробелами и переносами строк	47
(4.30) Стилевые свойства для управления фоном	48
(4.32) Резиновый фон	54
(4.35) Переполнение контейнера	55
(4.40) Спрайты	56
(4.70) Таблицы — основные теги и атрибуты тегов	57
(4.72) Таблицы — дополнительные теги и атрибуты тегов	59
(4.75) Управление слиянием рамок таблицы	61
(4.78) Ширина и высота таблицы и ячеек, строк, столбцов таблицы	62
(4.80) Объединение ячеек таблицы	63
Домашнее задание №4	64

Тема 5. Обтекание. Вёрстка макетов. Селекторы	67
(5.10) Обтекание	67
(5.30) Табличная и блочная вёрстка	75
(5.40) Столбцы одинаковой высоты в блочной вёрстке	77
(5.50) Выравнивание блочного содержимого	87
(5.70) Селектор элементов по классу	89
(5.75) Селектор элемента по идентификатору	90
Домашнее задание №5	90
Тема 6. Иерархические селекторы. Каскадирование	91
(6.20) Селектор элементов по тегу	91
(6.30) Универсальный селектор	91
(6.35) Селектор элементов по атрибутам	91
(6.40) Селектор элементов по нескольким критериям сразу	93
(6.45) Иерархические селекторы	93
(6.50) Селекторы псевдоклассов	95
(6.52) Селектор элементов по нескольким критериям сразу с использованием псевдоклассов	100
(6.55) Селекторы псевдоэлементов	100
(6.57)clearfix	102
(6.60) Комбинированные селекторы	103
(6.70) Веса стилевых селекторов	104
(6.80) Управление режимом отображения элементов	106
(6.85) Управление видимостью элементов	108
Домашнее задание №6a	110
Домашнее задание №6b	110
Домашнее задание №6c	111
Домашнее задание №6d	111
Тема 7. Медиазапросы. Адаптивная вёрстка	112
(7.20) Стилевая директива @media	112
(7.25) Медиазапросы	113
(7.90) Стилевые таблицы в отдельных файлах	115
(7.92) Порядок взаимодействия браузера с сервером, если на странице используются изображения и стилевой файл	116
Домашнее задание №7	117
Тема 8. Формы. Позиционирование. HTML5. Мультимедиа	118
(8.10) Порядок взаимодействия браузера с сервером при отправке формы	118
(8.15) Формы	119
(8.20) Позиционирование элементов	126
(8.35) Порядок наложения позиционированных элементов	128
(8.40) HTML5 — обзор	129
(8.42) HTML5 — семантическая вёрстка	130
(8.45) HTML5 — формы	134
Домашнее задание №8a	137
Домашнее задание №8b	138
Домашнее задание №8c	139
Тема 9. CSS3. Анимация. Flex. Grid layout	140
(9.15) SVG — пример	140
(9.25) SVG — краткое описание	141
(9.55) Модель flex-блоков	154
(9.57) Grid-раскладка	158
(9.65) Метаязыки описания стилей (Sass/SCSS)	167
Домашнее задание №9a,9b,9c,9d,9e,9f	172
Приложения	175

Тема 1. Введение в HTML. Ссылки

(1.10) Порядок взаимодействия браузера с сервером



Разработка веб-сайтов с использованием HTML, CSS

Учебное пособие (версия 14)

(1.45) Первые теги HTML

Парный тег **p** — параграф, или абзац.

Служит для обрамления одного абзаца текста.

Обычно отображается с некоторым отступом сверху и снизу текста.

HTML-код	Отображение в браузере
<pre><p>Маврикийский дронт, или додо (лат. <i>Raphus cucullatus</i>) - вымерший вид нелетающей птицы из монотипического рода <i>Raphus</i>. Являлся эндемиком острова Маврикий. </p> <p>Первое документальное упоминание о дронте появилось благодаря голландским мореплавателям, прибывшим на остров в 1598 году.</p></pre>	<p>Маврикийский дронт, или додо (лат. <i>Raphus cucullatus</i>) - вымерший вид нелетающей птицы из монотипического рода <i>Raphus</i>. Являлся эндемиком острова Маврикий.</p> <p>Первое документальное упоминание о дронте появилось благодаря голландским мореплавателям, прибывшим на остров в 1598 году.</p>

Парные теги **em** и **i** — курсив.

Служат для оформления текста курсивным (наклонным) шрифтом.

Оба тега отображаются одинаково. Предпочтительнее использовать **em**.

HTML-код	Отображение в браузере
<pre><p>Маврикийский дронт, или додо (лат. <i>Raphus cucullatus</i>) - птица из монотипического рода Raphus.</p></pre>	Маврикийский дронт, или додо (лат. <i>Raphus cucullatus</i>) - птица из монотипического рода <i>Raphus</i> .

Парные теги **strong** и **b** — жирность текста.

Служат для оформления текста полужирным шрифтом.

Оба тега отображаются одинаково. Предпочтительнее использовать **strong**.

HTML-код	Отображение в браузере
<pre><p>Маврикийский дронт, или додо (лат. Raphus cucullatus) - птица из монотипического рода Raphus.</p></pre>	Маврикийский дронт, или додо (лат. <i>Raphus cucullatus</i>) - птица из монотипического рода <i>Raphus</i> .

Парные теги **h1, h2, h3, h4, h5, h6** — заголовки различных уровней.

Служат для оформления заголовков текста. Обычно отображаются более крупным и/или жирным шрифтом с некоторыми отступами сверху и снизу.

HTML-код	Отображение в браузере
<pre><h1>Книга вторая. Учёба</h1> <h2>Глава первая. Начальная школа</h2> <h3>Раздел четвёртый. Так мы теряем друзей</h3> <h4>Подраздел. Первые подозрения</h4> <h5>День седьмой. Кулаки и синяки</h5> <h6>Урок второй. Математика</h6> <p>Математица совсем озверела. Сказала что я не знаю математики и поставила в дневник какую-то цифру. Ничего, вечера сейчас тёмные...</p></pre>	<p>Книга вторая. Учёба</p> <p>Глава первая. Начальная школа</p> <p>Раздел четвёртый. Так мы теряем друзей</p> <p>Подраздел. Первые подозрения</p> <p>День седьмой. Кулаки и синяки</p> <p>Урок второй. Математика</p> <p>Математица совсем озверела. Сказала что я не знаю математики и поставила в дневник какую-то цифру. Ничего, вечера сейчас тёмные...</p>

Одиночный тег **br** — переход на новую строку внутри абзаца.

Служит для переноса последующего текста на новую строку, в пределах того же абзаца.

HTML-код	Отображение в браузере
<pre><p>Однажды в студёную зимнюю пору
я из лесу вышел, был сильный мороз.
Гляжу - поднимается медленно в гору
лошадка, везущая хворосту воз.</p></pre>	Однажды в студёную зимнюю пору я из лесу вышел, был сильный мороз. Гляжу - поднимается медленно в гору лошадка, везущая хворосту воз.

Парный тег **ul** — маркированный список.

Служит для создания маркированного, т.е. неупорядоченного списка.

Его содержимым должны быть теги **li**.

Обычно отображается с отступом вправо и с маркером перед каждым элементом **li**.

Парный тег **ol** — нумерованный список.

Служит для создания нумерованного, т.е. упорядоченного списка.

Его содержимым должны быть теги **li**.

Обычно отображается с отступом вправо и с порядковым номером перед каждым элементом **li**.

Парный тег **li** — элемент маркированного или нумерованного списка.

HTML-код	Отображение в браузере
<pre><p>Мне нравится:</p> что вы больны не мной, что я больна не вами, что никогда тяжёлый шар земной не упливёт под нашими ногами. </pre>	Мне нравится: <ul style="list-style-type: none">что вы больны не мной,что я больна не вами,что никогда тяжёлый шар земной не упливёт под нашими ногами.
<pre><p>Мне нравится:</p> что вы больны не мной, что я больна не вами, что никогда тяжёлый шар земной не упливёт под нашими ногами. </pre>	Мне нравится: 1. что вы больны не мной, 2. что я больна не вами, 3. что никогда тяжёлый шар земной не упливёт под нашими ногами.

Одиночный тег **hr** — горизонтальная линия.

Служит для отображения горизонтальной линии, например, раздела между смысловыми частями текста.

HTML-код	Отображение в браузере
<pre><p>Так закончилось моё путешествие</p> <hr> <p>Прия домой и посчитав деньги...</p></pre>	Так закончилось моё путешествие <hr/> Прия домой и посчитав деньги...

(1.55) Первые атрибуты тегов HTML

Атрибут **align** — выравнивание текста.

Может принимать значения **left**, **center**, **right**, **justify**.

Служит для выравнивания содержимого абзаца или заголовка.

Может быть указан, например, для тегов **p** или **h1-h6**.

HTML-код	Отображение в браузере
<p align="left">После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.</p>	После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.
<p align="center">После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.</p>	После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.
<p align="right">После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.</p>	После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.
<p align="justify">После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.</p>	После оглушительного успеха в Голливуде Чаплин вынужден покинуть США под натиском громких скандалов.

Атрибут **type** для тега **ul** — форма маркера списка.

Может принимать значения **disc**, **square**, **circle**.

Служит для указания формы маркера в маркированном (неупорядоченном) списке.

HTML-код	Отображение в браузере
<p>В Африке:</p><ul type="disc">реки вот такой шириныгоры вот такой вышины	В Африке: <ul style="list-style-type: none">• реки вот такой ширины• горы вот такой вышины
<p>В Африке:</p><ul type="square">реки вот такой шириныгоры вот такой вышины	В Африке: <ul style="list-style-type: none">■ реки вот такой ширины■ горы вот такой вышины
<p>В Африке:</p><ul type="circle">реки вот такой шириныгоры вот такой вышины	В Африке: <ul style="list-style-type: none">○ реки вот такой ширины○ горы вот такой вышины

(1.75) Специальные символы HTML

Символы латиницы

Название	мнемоника	код10	код16	вид
неразрывный пробел	 	8#160;	8#xA0	
перевернутый восклицательный знак	!	8#161;	8#xA1	і
цент	¢	8#162;	8#xA2	¢
фунт стерлингов	£	8#163;	8#xA3	£
знак денежной единицы	¤	8#164;	8#xA4	¤
йена	¥	8#165;	8#xA5	¥
вертикальная черта	¦	8#166;	8#xA6	—
параметр	§	8#167;	8#xA7	§
диереза	¨	8#168;	8#xA8	়
знак авторского права	©	8#169;	8#xA9	©
показатель женского рода	&ogrf;	8#170;	8#xAA	ା
открывающаяся двойная угловая кавычка	«	8#171;	8#xB8	«
знак отрицания	¬	8#172;	8#xA5C	¬
мягкий перенос	­	8#173;	8#xA5D	
охраняемый знак	®	8#174;	8#xAE	®
надчеркивание	¯	8#175;	8#xAF	—
градус	°	8#176;	8#xB0	°
плюс-минус	±	8#177;	8#xB1	±
вторая степень	²	8#178;	8#xB2	²
третья степень	³	8#179;	8#xB3	³
острое ударение	´	8#180;	8#xB4	‘
знак микро	µ	8#181;	8#xB5	ୟ
конец абзаца	¶	8#182;	8#xB6	¶
средняя точка	·	8#183;	8#xB7	·
седиль	¸	8#184;	8#xB8	ׁ
единица в верхнем индексе	¹	8#185;	8#xB9	୧
показатель мужского рода	º	8#186;	8#xBA	୦
закрывающаяся двойная угловая кавычка	»	8#187;	8#xBB	»
одна четвертая	¼	8#188;	8#xBC	୧/୪
одна вторая	½	8#189;	8#xBD	୧/୨
три четверти	¾	8#190;	8#xBE	୩/୪
перевернутый вопросительный знак	?	8#191;	8#xBF	୯
А с тупым ударением	À	8#192;	8#xC0	ା
А с острым ударением	Á	8#193;	8#xC1	ଅ

Символы латиницы (продолжение)

Название	мнемоника	код10	код16	вид
А с циркумфлексом	Â	8#194;	8#xC2	ା
А с тильдой	Ã	8#195;	8#xC3	ା
А с диерезой	Ä	8#196;	8#xC4	ା
А с кружком	Å	8#197;	8#xC5	ା
лигатура АЕ	&AEig;	8#198;	8#xC6	ଫେ
С седилем	Ç	8#199;	8#xC7	ଚ୍ଛ
Е с тупым ударением	È	8#200;	8#xC8	ୟେ
Е с острым ударением	É	8#201;	8#xC9	ୟେ
Е с циркумфлексом	Ê	8#202;	8#xC4	ୟେ
Е с диерезой	Ë	8#203;	8#xCB	ୟେ
І с тупым ударением	Ì	8#204;	8#xCC	ି
І с острым ударением	Í	8#205;	8#xCD	ି
І с циркумфлексом	Î	8#206;	8#xC5	ି
І с диерезой	Ï	8#207;	8#xCF	ି
ЕТН	Ð	8#208;	8#xD0	ଡ
Н с тильдой	Ñ	8#209;	8#xD1	ନ
О с тупым ударением	Ò	8#210;	8#xD2	ଓ
О с острым ударением	Ó	8#211;	8#xD3	ଓ
О с циркумфлексом	Ô	8#212;	8#xD4	ଓ
О с тильдой	&Otild;	8#213;	8#xD5	ଓ
О с диерезой	Ö	8#214;	8#xD6	ଓ
знак умножения	×	8#215;	8#xD7	ଗ୍ରେ
О перечеркнутое	Ø	8#216;	8#xD8	ଘ
І с тупым ударением	Ù	8#217;	8#xD9	ୟୁ
І с острым ударением	Ú	8#218;	8#xD4	ୟୁ
І с циркумфлексом	Û	8#219;	8#xD5	ୟୁ
І с диерезой	Ü	8#220;	8#xDC	ୟୁ
Ү с острым ударением	Ý	8#221;	8#xDD	ୟୁ
THORN	Þ	8#222;	8#xDE	ର
двойное s	ß	8#223;	8#xDF	ବ୍ୟ
а с тупым ударением	`	8#224;	8#xE0	ା
а с острым ударением	á	8#225;	8#xE1	ା
а с циркумфлексом	â	8#226;	8#xE2	ା
а с тильдой	ã	8#227;	8#xE3	ା

Специальные символы

Название	Мнемоника	код10	код16	вид	код10	код16	вид
а с диерезой	ä	ä	ä	ä	&qot;	"	"
а с кружком	å	å	å	å	&at;	&	&
лигатура ae	æ	æ	æ	æ	<	<	<
с седиллем	ç	ç	ç	ȝ	>	>	>
е с тупым ударением	è	è	è	े	Œ	Œ	Œ
е с острый ударением	é	é	é	é	&oeig;	œ	œ
е с циркумфлексом	ê	ê	ê	ê	Š	Š	Š
е с диерезой	ë	ë	ë	ë	š	š	š
í с тупым ударением	ì	ì	á	í	Ÿ	Ÿ	Ÿ
í с острый ударением	í	í	à	í	ˆ	ˆ	ˆ
í с циркумфлексом	î	î	á	î	˜	˜	˜
í с диерезой	ï	ï	à	í	 	 	 
eth	ð	ð	ð	ð	 	 	 
п с тильдой	ñ	ñ	ñ	ń	 	 	 
о с тупым ударением	ò	ò	ò	ò	‌	‌	‌
о с острый ударением	ó	ó	ó	ó	‌	‍	‍
о с циркумфлексом	ô	ô	ô	ô	&irm;	‎	‎
о с тильдой	õ	õ	õ	ö	–	‏	‏
о с диерезой	ö	ö	ö	ö	‐	–	–
знак деления	÷	÷	÷	÷	—	—	—
о перечеркнутое	ø	ø	ø	ø	открытывающая одинарная кавычка	‘	‘
и с тупым ударением	ù	ù	ù	ù	закрывающая одинарная кавычка	’	’
и с острый ударением	í	ú	ú	ú	нижняя одинарная кавычка	‚	‚
и с циркумфлексом	î	û	û	ő	открывающая двойная кавычка	“	“
и с диерезой	ï	ü	ü	ő	закрывающая двойная кавычка	”	”
У с острый ударением	ý	ý	ý	ў	нижняя двойная кавычка	&bddquo;	„
thorn	þ	þ	þ	þ	кинжал	&daggar;	†
у с диерезой	ü	ÿ	ÿ	ӱ	двойной кинжал	‡	‡
					знак промилле	‰	‰
					открывающая угловая кавычка	‘	‹
					закрывающая угловая кавычка	’	›
					евро	€	€

Символы латиницы (продолжение)

Название	Мнемоника	код10	код16	вид
а с диерезой	ä	ä	ä	ä
а с кружком	å	å	å	å
лигатура ae	æ	æ	æ	æ
с седиллем	ç	ç	ç	ȝ
е с тупым ударением	è	è	è	े
е с острый ударением	é	é	é	é
е с циркумфлексом	ê	ê	ê	ê
е с диерезой	ë	ë	ë	ë
í с тупым ударением	ì	ì	á	í
í с острый ударением	í	í	à	í
í с циркумфлексом	î	î	á	î
í с диерезой	ï	ï	à	í
eth	ð	ð	ð	ð
п с тильдой	ñ	ñ	ñ	ń
о с тупым ударением	ò	ò	ò	ò
о с острый ударением	ó	ó	ó	ó
о с циркумфлексом	ô	ô	ô	ô
о с тильдой	õ	õ	õ	ö
о с диерезой	ö	ö	ö	ö
знак деления	÷	÷	÷	÷
о перечеркнутое	ø	ø	ø	ø
и с тупым ударением	ù	ù	ù	ù
и с острый ударением	í	ú	ú	ú
и с циркумфлексом	î	û	û	ő
и с диерезой	ï	ü	ü	ő
У с острый ударением	ý	ý	ý	ў
thorn	þ	þ	þ	þ
у с диерезой	ü	ÿ	ÿ	ӱ

Математические символы и греческие буквы

Математические символы и греческие буквы (продолжение)

Название	Мнемоника	код ₁₀	код ₁₆	вид	Название	Мнемоника	код ₁₀	код ₁₆	вид
курсивное f	ƒ	&##402;	ƒ	f	строчная мю	π	μ	μ	μ
протисная альфа	Α	Α	Α	Α	строчная ню	π	ν	ν	ν
протисная бета	Β	Β	Β	Β	строчная кси	ξ	ξ	ξ	ξ
протисная гамма	Γ	Γ	Γ	Γ	строчный омикрон	ο	ο	ο	ο
протисная дельта	Δ	Δ	Δ	Δ	строчная пи	π	π	π	π
протисной эпсилон	Ε	Ε	Ε	Ε	строчная ро	ρ	ρ	ρ	ρ
протисная дзета	Ζ	Ζ	Ζ	Ζ	строчная сигма конечная	ς	ς	ς	ς
протисная эта	Η	Η	Η	Η	строчная сигма	σ	σ	σ	σ
протисная тета	Θ	Θ	Θ	Θ	строчная тау	τ	τ	τ	τ
протисная иота	Ι	Ι	Ι	Ι	строчная ипсилон	υ	υ	υ	υ
протисная каппа	Κ	Κ	Κ	Κ	строчная фи	φ	φ	φ	φ
протисная ламбда	Λ	Λ	Λ	Λ	строчная хи	χ	χ	χ	χ
протисная мю	Μ	Μ	Μ	Μ	строчная пси	ψ	ψ	ψ	ψ
протисная ню	Ν	Ν	Ν	Ν	строчная омега	ω	ω	ω	ω
протисная кси	Ξ	Ξ	Ξ	Ξ	символ строчная тета	ϑ	ϑ	ϑ	ϑ
протисной омикрон	Ο	Ο	Ο	Ο	ипсилон с крючком	υ	ϒ	ϒ	ϒ
протисная пи	Π	Π	Π	Π	символ пи	π	ϖ	ϖ	π
протисная ро	Ρ	Ρ	Ρ	Ρ	маркер списка	•	•	•	•
протисная сигма	Σ	Σ	Σ	Σ	многоточие	…	…	…	…
протисная тау	Τ	Τ	Τ	Τ	знак прим	′	′	′	́
протисная ипсилон	Υ	Υ	Υ	Υ	знак двойной прим	″	″	″	”
протисная фи	Φ	Φ	Φ	Φ	надчеркивание	‾	‾	‾	–
протисная хи	Χ	Χ	Χ	Χ	дробная черта	⁄	⁄	⁄	/
протисная пси	Ψ	Ψ	Ψ	Ψ	рукописная Р	℘	℘	℘	϶
протисная омега	Ω	Ω	Ω	Ω	мнимая часть числа	ℑ	ℑ	ℑ	϶
строчная альфа	α	α	α	α	действительная часть числа	ℜ	ℜ	ℜ	϶
строчная бета	β	β	β	β	торговая марка	™	™	™	™
строчная гамма	γ	γ	γ	γ	алеф	ℵ	ℵ	ℵ	ℵ
строчная дельта	δ	δ	δ	δ	стрелка влево	←	←	←	←
строчная эпсилон	ε	ε	ε	ε	стрелка вверх	↑	↑	↑	↑
строчная дзета	ζ	ζ	ζ	ζ	стрелка вправо	→	→	→	→
строчная эта	η	η	η	η	стрелка вниз	↓	↓	↓	↓
строчная тета	θ	θ	θ	θ	стрелка влево-вправо	↔	↔	↔	↔
строчная иота	ι	ι	ι	ι	возврат каретки	↵	↵	↵	↖
строчная каппа	κ	κ	κ	κ					
строчная ламбда	λ	λ	λ	λ					

Математические символы и греческие буквы (продолжение)

Математические символы и греческие буквы (продолжение)

Название	мнемоника	код10	код16	вид	Название	мнемоника	код10	код16	вид
двойная стрелка влево	&Ilt;	⇐	⇐	←	не равно	≠	≠	≠	≠
двойная стрелка вверх	&iAgt;	⇑	⇑	↑	тождественно равно	≡	≡	≡	≡
двойная стрелка вправо	&rAgt;	⇒	⇒	⇒	меньше или равно	≤	≤	≤	≤
двойная стрелка вниз	&dAgt;	⇓	⇓	↓	больше или равно	≥	≥	≥	≥
двойная стрелка влево-вправо	⇔	⇔	⇔	↔	подмножество	⊂	⊂	⊂	⊂
квантор всеобщности	∀	∀	∀	∀	надмножество	⊃	⊃	⊃	⊃
знак дифференциала	∂	∂	∂	∂	не подмножество	⊄	⊄	⊄	⊄
квантор существования	∃	∃	∃	∃	подмножество или равно	⊆	⊆	⊆	⊆
пустое множество	∅	∅	∅	∅	надмножество или равно	⊇	⊇	⊇	⊇
набла	∇	∇	∇	∇	прямая сумма	⊕	⊕	⊕	⊕
принадлежит множеству	∈	∈	∈	∈	векторное произведение	⊗	⊗	⊗	⊗
не принадлежит множеству	∉	∉	∉	∉	перпендикулярно	⊥	⊥	⊥	⊥
является членом	∋	∋	∋	∊	оператор точка	⋅	⋅	⋅	·
п-арное произведение	∏	∏	∏	∏	левый верхний угол	&iceil;	⌈	⌈	⌈
п-арная сумма	∑	∑	∑	Σ	правый верхний угол	&ceil;	⌉	⌉	⌋
знак минус	−	−	−	-	левый нижний угол	⌊	⌊	⌊	⌊
оператор звездочка	∗	∗	∗	*	правый нижний угол	⌋	⌋	⌋	⌋
радикал	√	√	√	√	левая угловая скобка	⟨	〈	〈	{
пропорционально	∝	∝	∝	∝	правая угловая скобка	⟩	〉	〉	}
бесконечность	∞	∞	∞	∞	ромб	◊	◊	◊	◇
угол	∠	∠	∠	∠	пики	♠	♠	♠	♠
логическое И	∧	∧	∧	∧	трофы	♣	♣	♣	♣
логическое ИЛИ	∨	∨	∨	∨	черви	♥	♥	♥	♥
пересечение	∩	∩	∩	∩	бубы	⋄	♦	♦	♦
объединение	∪	∪	∪	∪					
интеграл	∫	∫	∫	∫					
следовательно	∴	∴	∴	∴					
оператор тильда	∼	∼	∼	˜					
приблизительно равно	≅	≅	≅	≈					
асимптотически равно	≈	≈	≈	≈					

(1.80) Структура URI

Структура URI веб-страниц

протокол://домен/страница?запрос#закладка

Примеры:

<http://news.tut.by/society/361194.html>

абсолютная ссылка; здесь http — протокол, news.tut.by — домен, /society/361194.html — страница

[/society/361195.html](#)

относительная ссылка — протокол и домен не указаны, то есть, протокол и домен **не изменились**, остались теми же что в том документе в котором была ссылка; /society/361194.html — страница

<http://ru.wikipedia.org/wiki/Beatles.phtml#Cavern>

абсолютная ссылка; здесь http — протокол, ru.wikipedia.org — домен, /wiki/Beatles.phtml — страница, Cavern — закладка

[#Quarrymen](#)

относительная ссылка — протокол, домен и страница не указаны, то есть, они **не изменились**; Quarrymen — закладка

<http://yandex.by/yandsearch?lr=157&text=жаба>

абсолютная ссылка; здесь http — протокол, yandex.by — домен, yandsearch — страница, lr=157&text=жаба — запрос

Другие протоколы

<mailto:loktev.alex.74@gmail.com>

абсолютная ссылка на почтовый ящик;
здесь mailto — протокол, loktev.alex.74@gmail.com — адрес почтового ящика

<skype:loktev-alexey>

абсолютная ссылка на учётную запись Skype;
здесь skype — протокол, loktev-alexey — учётная запись Skype

[tel: +233 322 233](tel:+233 322 233)

абсолютная ссылка на телефон;
здесь tel — протокол, 233 322 233 — номер телефона.

(1.90) Гиперссылки

Парный тег **a** — гиперссылка.

Атрибут **href** тега **a** — адрес гиперссылки.

Атрибут **title** тега **a** — подсказка, всплывающая при поднесении мыши к гиперссылке.

Служит для создания гиперссылки на другую (или эту же) веб-страницу или на некий документ.

Содержимое тега будет **текстом** гиперссылки, а в атрибуте **href** — URL, куда ведёт гиперссылка.

Обычно изображается с подчёркиванием, а при поднесении мыши — меняется форма курсора.

HTML-код	Отображение в браузере
<pre><p>Если мы хотим что-то найти в Интернете, мы пользуемся поисковыми системами, например, гугл или яндекс.</p></pre>	Если мы хотим что-то найти в Интернете, мы пользуемся поисковыми системами, например, гугл или яндекс .

Атрибут **target** тега **a** — окно (вкладка) браузера для гиперссылки.

Указывает вкладку браузера, в которой нужно открыть веб-страницу, на которую ведёт гиперссылка.

Если **target** не указан, веб-страница открывается в этой же вкладке браузера.

Если указан **target=_blank**, веб-страница открывается в новой вкладке браузера.

Если в качестве значения **target** указано некое произвольное имя, веб-страница открывается в новой вкладке браузера, и все последующие гиперссылки с таким же значением **target** будут открываться в ней же.

Атрибут **name** тега **a** превращает гиперссылку в закладку.

Тег **a** при наличии атрибута **name** является уже не гиперссылкой, а закладкой. То есть, на это место данной страницы можно будет делать гиперссылки.

Указывать атрибут **href** в этом случае не следует.

Домашнее задание №1a

Сверстать проект DIARY по образцу, приведённому ниже.
Ссылки должны открываться в новой вкладке браузера и могут вести куда угодно;
следует заполнить атрибуты title для ссылок.

Дневник партизана

1. **Вторник.** Выбили [немцев](#) с опушки. Засели в доме лесника.
2. **Среда.** Мощная контратака немцев вынудила нас оставить занятые ранее позиции.
3. **Четверг.** Под покровом ночи смогли снова засесть в доме лесника. Курим «Беломор».
4. **Пятница.** Вернулся лесник и всех выгнал.

© бородатый [анекдот](#)

Домашнее задание №1b

Создать проект VIZBOR. Проект будет состоять из двух HTML-страниц.

Страница №1: Добавить из библиотеки текст vechno_student.txt. Создать HTML-страничку vechno_student.html, перенести в неё текст статьи, разметить абзацы, расставить (где нужно) закладки.

Страница №2: Создать страничку vizbor.html по образцу, приведённому ниже. Ссылки должны вести на первую страницу (vechno_student.html), прокрутку к нужному месту страницы реализовать любым из двух изученных способов.

Юрий Визбор

Вечно студент

[читать сначала](#)

[читать с последнего абзаца](#)

IT-Academy.by

Тема 2. Изображения. Введение в CSS

(2.20) Изображения

Одиночный тег **img** — изображение.

Атрибут **src** тега **img** — URL изображения.

Служит для вставки изображения, URL которого указан в атрибуте **src**.

HTML-код	Отображение в браузере
<pre><p>Это зубака:
</p></pre>	Это зубака: 
<pre><p>А это — воробей:
</p></pre>	А это — воробей: 

Атрибуты **width** и **height** тега **img** — размер изображения.

Устанавливают размер изображения на странице браузера. Необходимы, если мы хотим, чтобы изображение на странице имело размер, отличающийся от оригинального.

HTML-код	Отображение в браузере
<pre><p>Это — воробей:
</p></pre>	Это — воробей: 
<pre><p>Это — воробей:
</p></pre>	Это — воробей: 

Атрибут **alt** тега **img** — альтернативный текст.

Задаёт текст, который появляется вместо изображения, если изображение почему-либо не может быть отображено.

Атрибут **title** тега **img** — всплывающая подсказка.

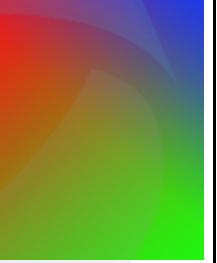
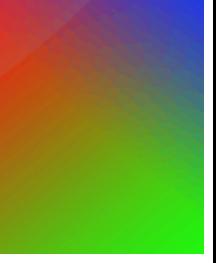
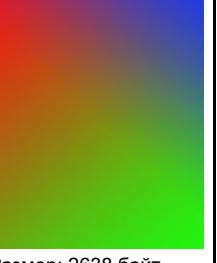
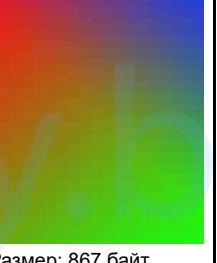
Задаёт текст, который всплывает при поднесении курсора мыши к изображению.

(2.35) Форматы изображений

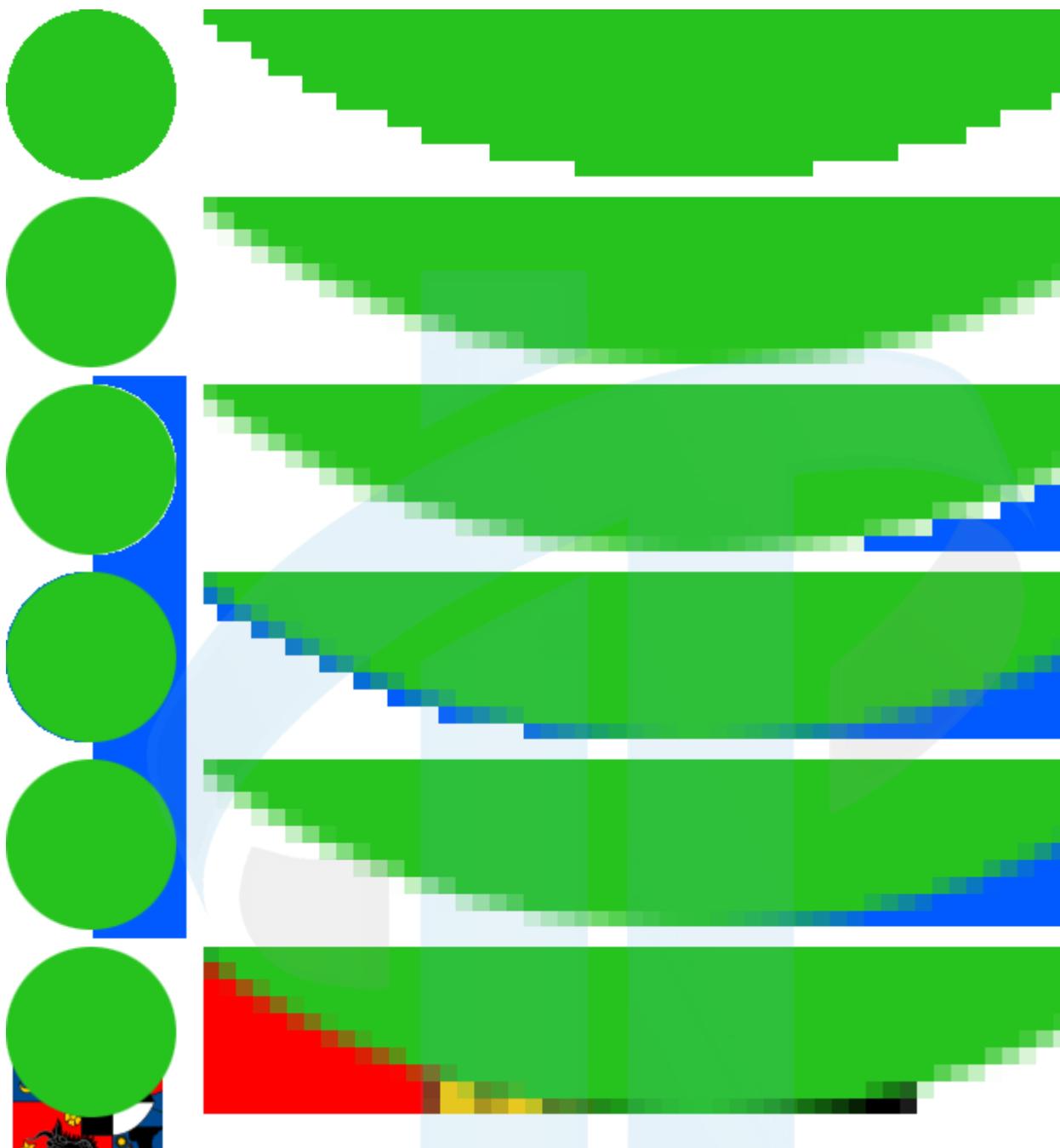
Свойства форматов изображений

формат	кол-во цветов	иска-жения	прозрачность	назначение
PNG	16 М	нет	256 уровней прозрачности	полноцветные изображения, потери качества недопустимы; изображения с полупрозрачностью
GIF	256	нет	2 уровня прозрачности (т.е. каждая точка или прозрачна, или нет)	малоцветные изображения
JPEG	16 М	да	нет	естественные полноцветные изображения, потери качества допустимы

Примеры изображений в различных форматах

	дельфиниум	Синатра	герб	двуцветный градиент	трёхцветный градиент
PNG					
	Размер: 52687 байт	Размер: 29670 байт	Размер: 34315 байт	Размер: 17344 байт	Размер: 22546 байт
GIF					
	Размер: 20910 байт	Размер: 25075 байт	Размер: 16144 байт	Размер: 13612 байт	Размер: 9572 байт
JPEG q=70					
	Размер: 11829 байт	Размер: 7936 байт	Размер: 19121 байт	Размер: 2364 байт	Размер: 2638 байт
JPEG q=10					
	Размер: 3088 байт	Размер: 2793 байт	Размер: 5406 байт	Размер: 726 байт	Размер: 867 байт

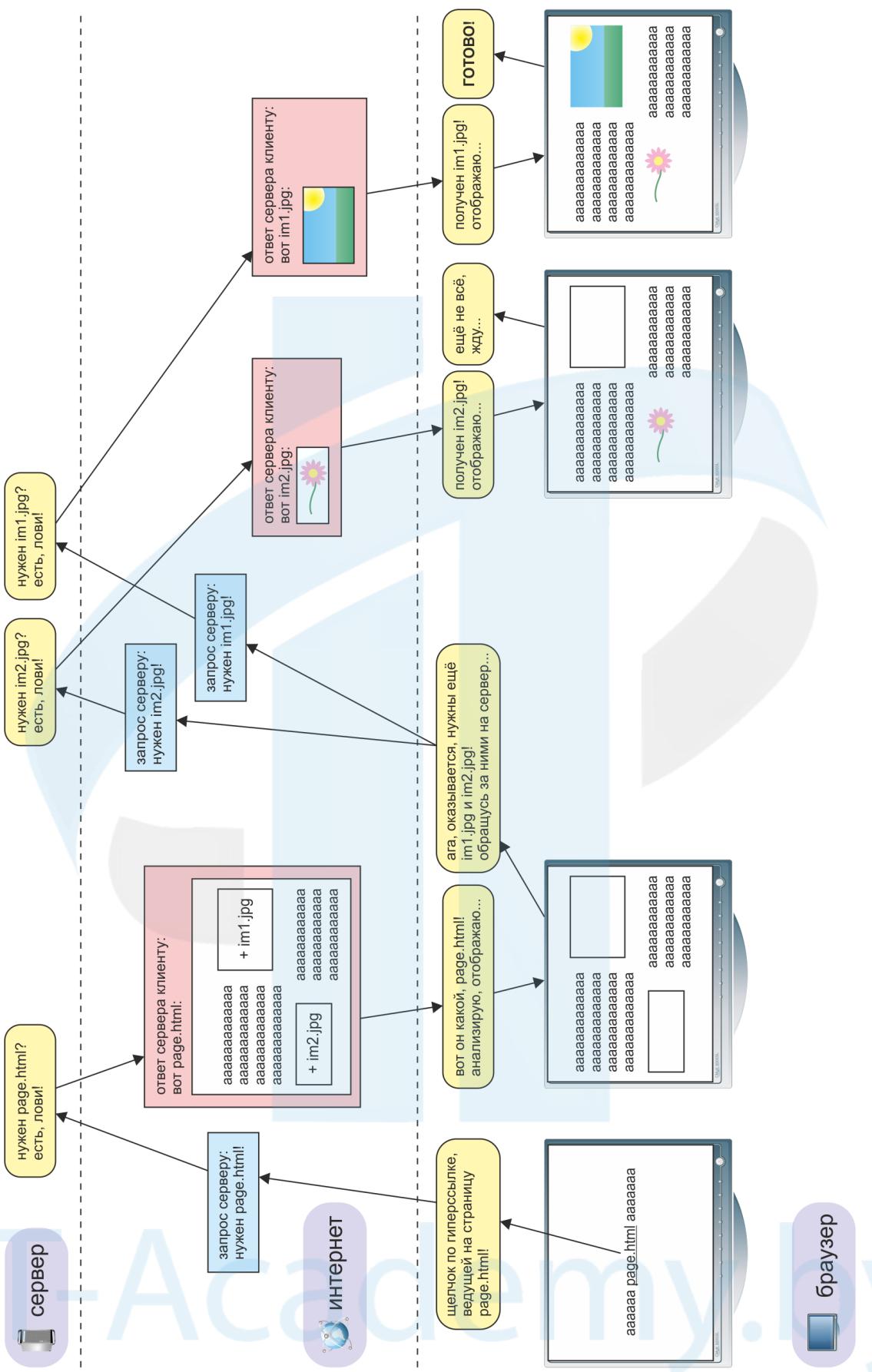
(2.38) Проблема окантовки изображений



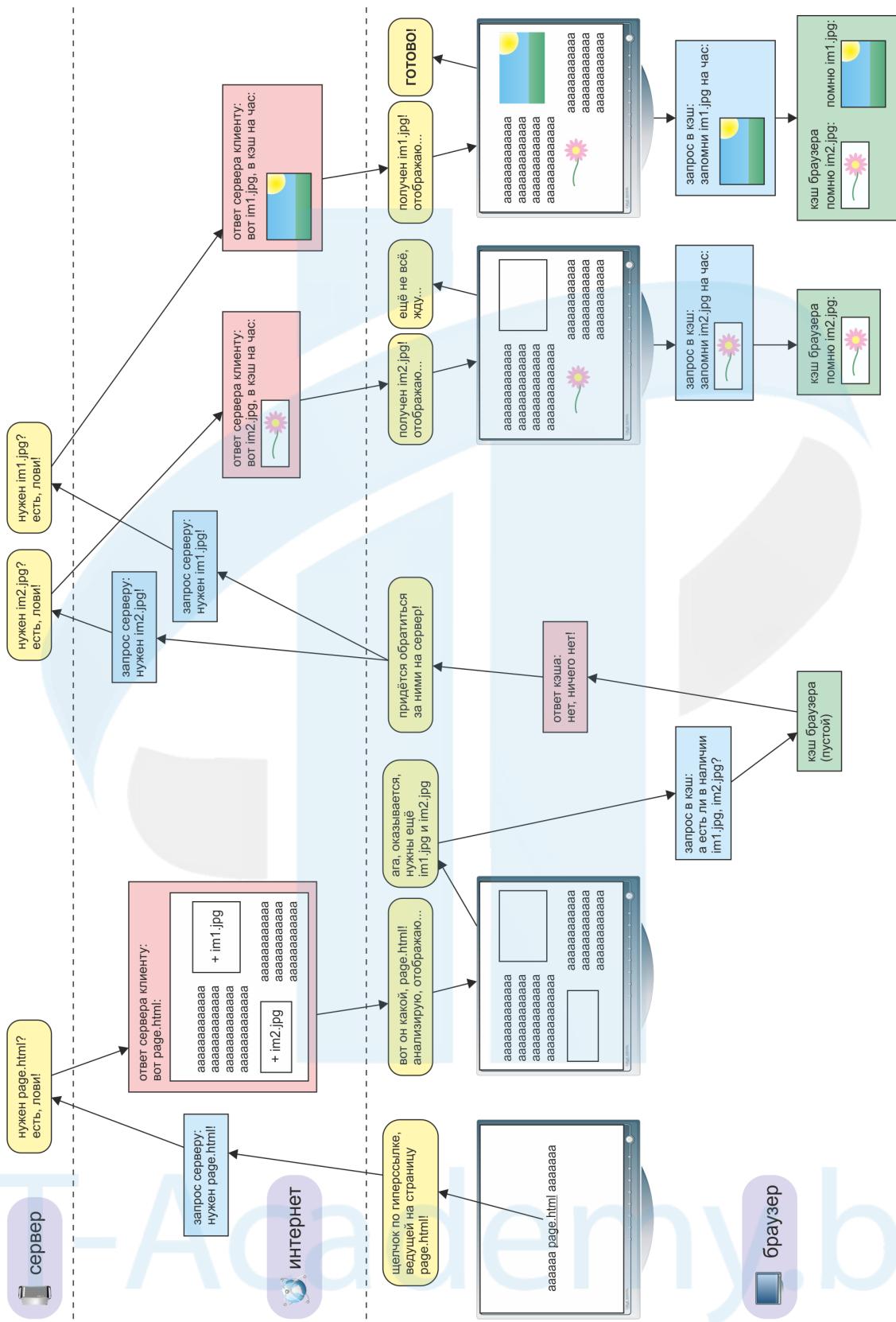
1. Круг с «рваной» окантовкой, белые углы, прозрачности нет.
2. Используется антиалиасинг — на окантовке полутоночные пиксели, имеющие промежуточный цвет между цветом круга и цветом углов, прозрачности нет.
3. Если нужно чтобы вместо белых углов проглядывал фон заднего плана, мы можем белые пиксели углов сделать прозрачными, но полутоночные пиксели окантовки будут видны на любом небелом фоне, т.к. являются полутоночными между белым и зелёным.
4. Мы можем окантовку сделать не из зелёно-белых, а из зелёно-синих пикселей, тогда на синем фоне круг будет выглядеть хорошо, но на белом фоне будет видна сине-зелёная окантовка.
5. Если окантовка сделана не зелёно-белыми, а зелёно-полупрозрачными пикселями, то такой круг будет хорошо выглядеть на фоне любого цвета.
6. Также, такой круг будет выглядеть хорошо на фоне любой сложности.

Полупрозрачность пикселей поддерживается ТОЛЬКО форматом PNG.

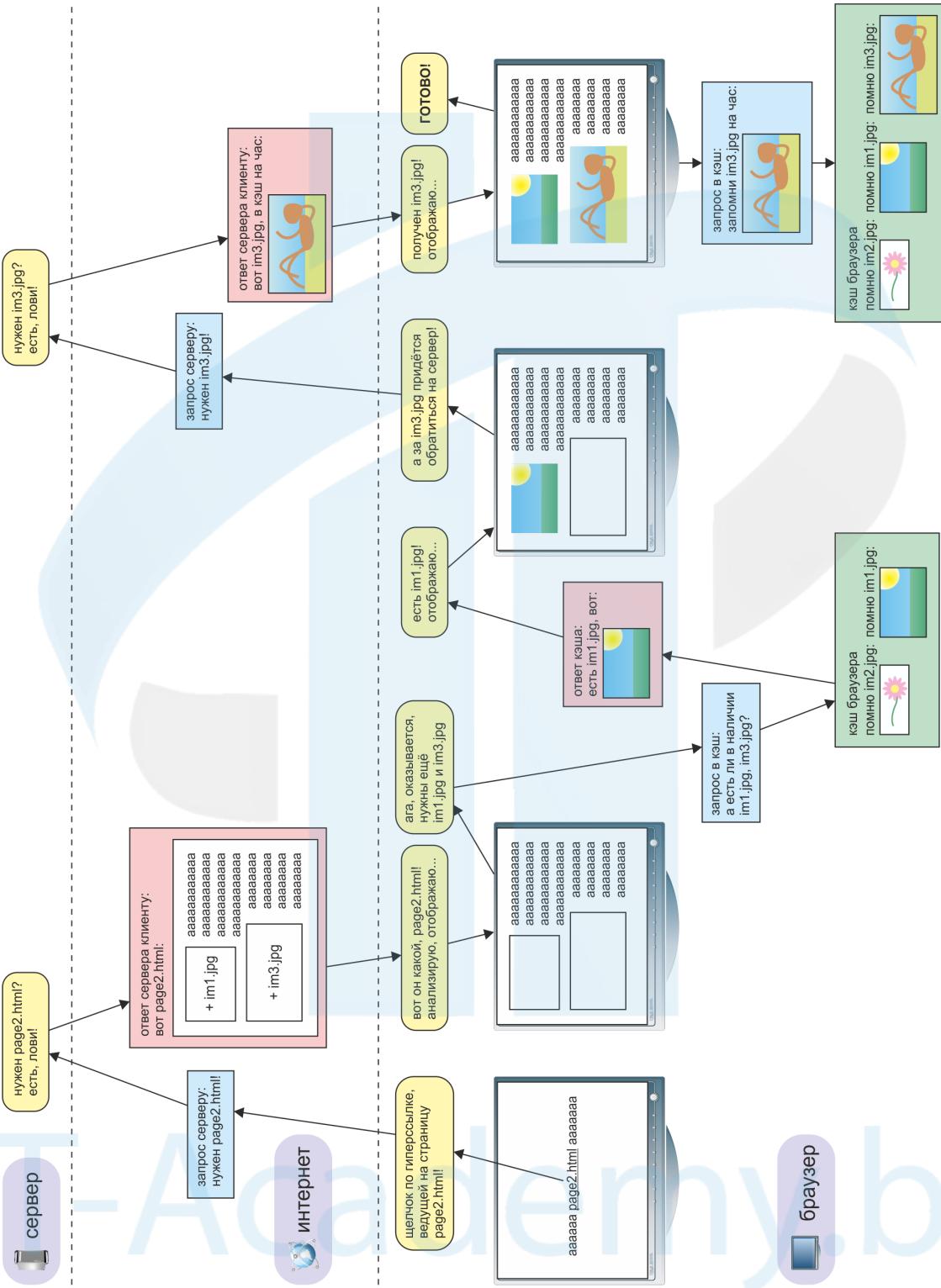
(2.40) Порядок взаимодействия браузера с сервером, если на странице используется изображения



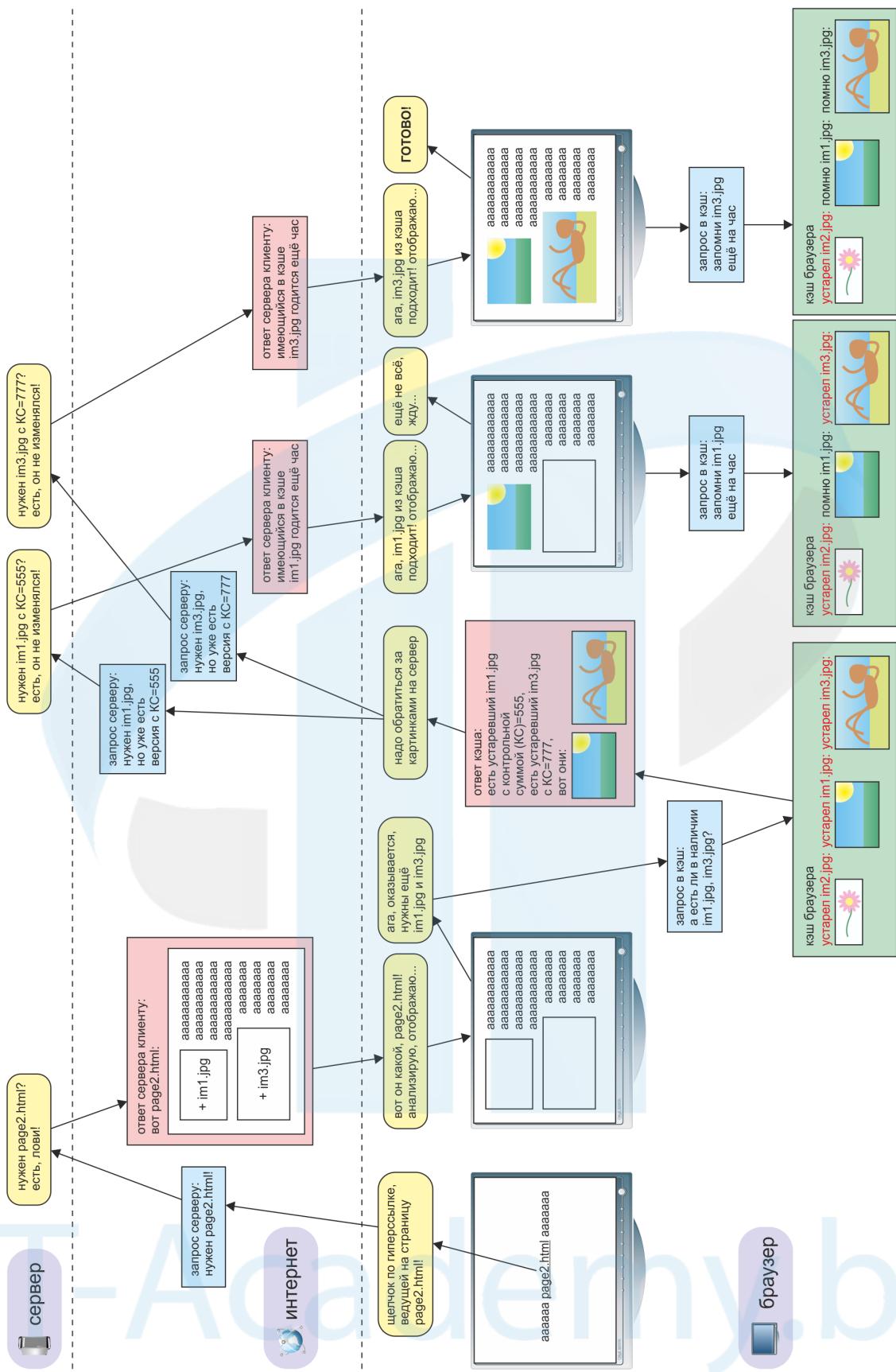
(2.45а) Порядок взаимодействия браузера с сервером, если на странице используется изображение (кэш пустой)



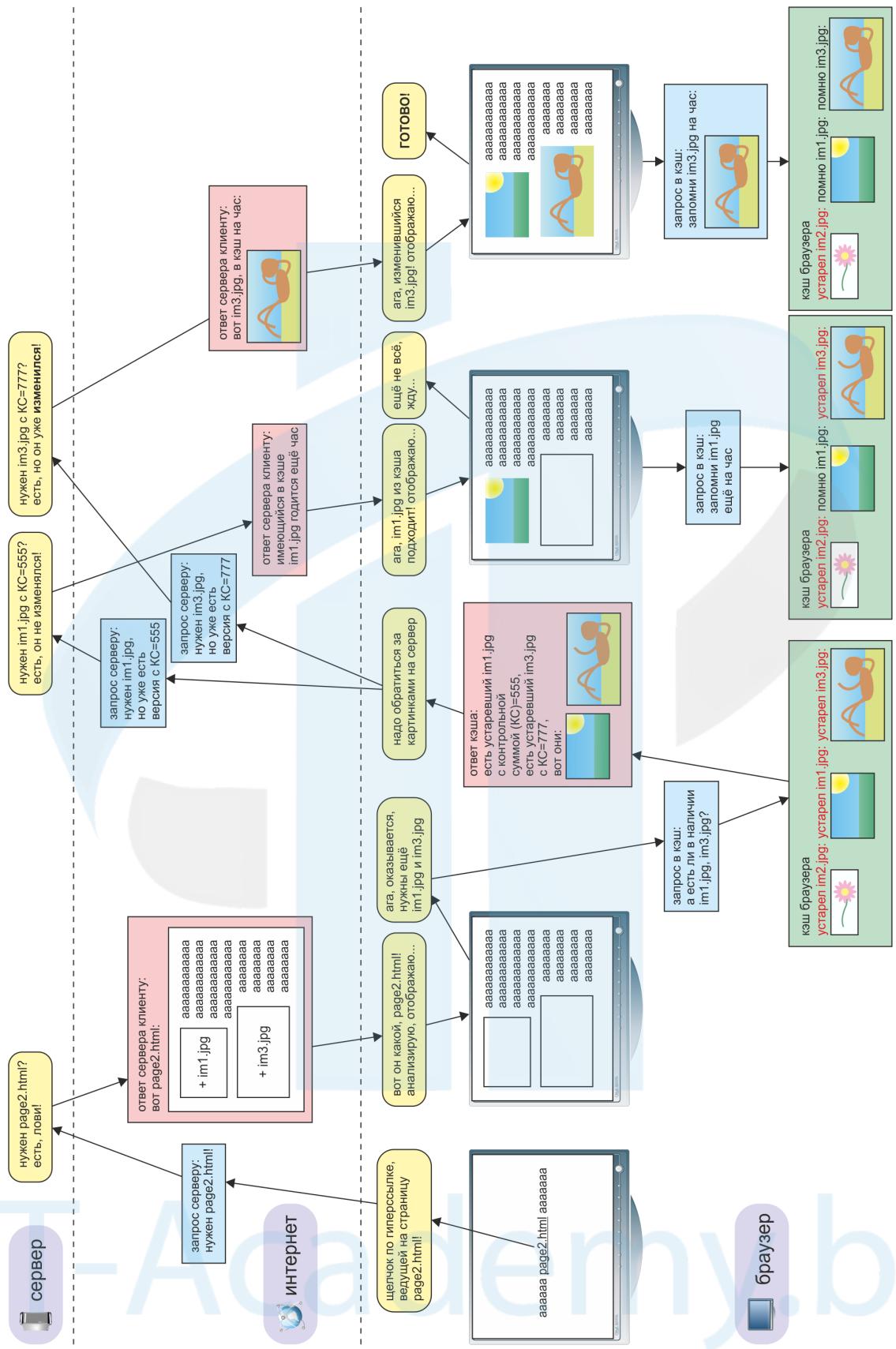
(2.45b) Порядок взаимодействия браузера с сервером, если на странице используется изображения (прошло 5 минут)



(2.45с) Порядок взаимодействия браузера с сервером, если на странице используется изображения (прошло 2 часа)



(2.45d) Порядок взаимодействия браузера с сервером, если на странице используется изображения (прошло 5 часов, файл изменился)



(2.75) Первые стилевые свойства

Стилевое свойство **text-decoration** — украшение текста.

Может применяться практически к любому элементу, в котором содержится текст.

Возможные значения:

text-decoration: none — обычный текст.

text-decoration: underline — подчёркнутый текст.

text-decoration: overline — надчёркнутый текст.

text-decoration: line-through — перечёркнутый текст.

HTML-код	Отображение в браузере
<pre><p>На собеседовании свои достиныства можно подчеркнуть, а можно перечеркнуть.
<="" p><="" pre=""/></pre>	На собеседовании свои достоинства можно <u>подчеркнуть</u> , а можно перечеркнуть .

Стилевое свойство **text-transform** — преобразование текста.

Может применяться практически к любому элементу, в котором содержится текст.

Возможные значения:

text-transform: none — обычный текст.

text-transform: capitalize — делает первую букву каждого слова прописной.

text-transform: uppercase — делает все буквы прописными.

text-transform: lowercase — делает все буквы строчными.

HTML-код	Отображение в браузере
<pre><p> Вот ПЕРВЫЙ пример,
 ВТОРОЙ пример,
 ТРЕТИЙ пример
 и ЧЕТВЁРТЫЙ размер!
 </p></pre>	Вот ПЕРВЫЙ пример, ВТОРОЙ Пример, третий пример и ЧЕТВЁРТЫЙ РАЗМЕР!

Стилевое свойство **list-style-type** — вид маркера списка.

Может применяться к элементам **ul** и **ol**, а также к **li**.

Некоторые возможные значения:

none — нет маркера

disc — чёрный кружок

circle — светлый кружок

square — чёрный квадратик

decimal — десятичные числа (1, 2, 3, ...)

decimal-leading-zero — десятичные числа с ведущим нулем (01, 02, 03, ... 98, 99)

lower-roman — строчные римские числа (i, ii, iii, iv, v, ...)

upper-roman — прописные римские числа (I, II, III, IV, V, ...)

lower-greek — строчные греческие буквы

lower-latin — строчные латинские буквы (a, b, c, ..., z)

upper-latin — прописные латинские буквы (A, B, C, ..., Z)

HTML-код	Отображение в браузере
<pre><ol style="list-style-type: square"> учиться, учиться, и ещё раз учиться! </pre>	<ul style="list-style-type: none">■ учиться,■ учиться,■ и ещё раз учиться!
<pre><ol style="list-style-type: circle"> учиться, <li style="list-style-type: disc">учиться, и ещё раз учиться! </pre>	<ul style="list-style-type: none">○ учиться,● учиться,○ и ещё раз учиться!

Стилевое свойство **list-style-image** — изображение вместо маркера списка.

Может применяться к элементам **ul** и **ol**.

HTML-код	Отображение в браузере
<pre><ol style="list-style-image: url('images/smile2.gif')"> учиться, учиться, и ещё раз учиться! </pre>	<ul style="list-style-type: none">🟡 учиться,🟡 учиться,🟡 и ещё раз учиться!

Стилевое свойство **cursor** — форма курсора мыши.

Задаёт форму, которую принимает курсор мыши при поднесении к данному элементу.

Может применяться практически к любому элементу.

Возможные значения:

Запись	Вид	Описание вида курсора, традиционное назначение курсора
<code>cursor: auto</code>		форма курсора определяется браузером в зависимости от типа элемента, автоматически
<code>cursor: default</code>		курсор операционной системы по умолчанию; в Windows — стрелка
<code>cursor: crosshair</code>		ожидается точное попадание мышью внутри элемента
<code>cursor: help</code>		для данного элемента можно получить справку
<code>cursor: move</code>		элемент можно перемещать либо элемент уже перемещается
<code>cursor: pointer</code>		на элемент можно нажимать/щёлкать
<code>cursor: progress</code>		элемент выполняет некоторую работу, но, возможно, будет реагировать на нажатие
<code>cursor: text</code>		элемент содержит текст, который можно выделять и/или изменять
<code>cursor: wait</code>		элемент очень занят и не может реагировать на нажатие
<code>cursor: n-resize</code>		верхнюю границу элемента можно перетягивать или она уже перетягивается
<code>cursor: s-resize</code>		нижнюю границу элемента можно перетягивать или она уже перетягивается
<code>cursor: e-resize</code>		правую границу элемента можно перетягивать или она уже перетягивается
<code>cursor: w-resize</code>		левую границу элемента можно перетягивать или она уже перетягивается
<code>cursor: ne-resize</code>		правый верхний угол элемента можно перетягивать или он уже перетягивается
<code>cursor: sw-resize</code>		левый нижний угол элемента можно перетягивать или он уже перетягивается
<code>cursor: se-resize</code>		правый нижний угол элемента можно перетягивать или он уже перетягивается
<code>cursor: nw-resize</code>		левый верхний угол элемента можно перетягивать или он уже перетягивается
<code>cursor: url(URL)</code> <code>cursor: url(URL) X Y</code>		вместо курсора отображается указанное изображение; можно также через пробелы добавить два числа — координаты «горячей точки» — той точки, которая является «остриём» курсора.

Можно перечислять несколько видов курсора через запятую; тогда, если браузер не сможет воспользоваться первым курсором, он воспользуется вторым, и т.д.

Для курсора-изображения *Internet Explorer* поддерживает форматы CUR и ANI;
Firefox, *Chrome* и *Safari* поддерживают форматы CUR, PNG, GIF, JPG;
Opera не поддерживает курсоры-изображения вообще.

После курсора-изображения обязательно через запятую указать какой-либо из базовых курсоров; он будет использован, если изображение курсора почему-либо использовать нельзя.

Домашнее задание №2

Создать проект CHILDBOOK.

Добавить из библиотеки файл childbook.png (его содержимое также приведено ниже); вырезать из него изображения, преобразовать их в оптимальный формат.

Сверстать газету с использованием этих изображений.

В HTML-комментариях обосновать выбранные форматы изображений.

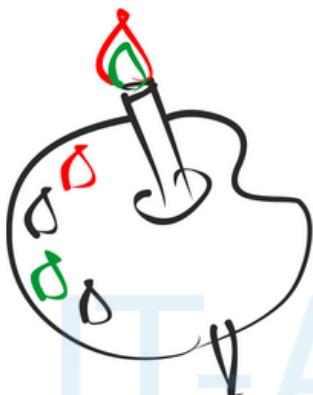
Подумать, почему для файла самого задания (childbook.png) использован формат PNG.



За рулём сижу довольный —
Ездить просто и легко!
Только руль чуть-чуть огромный
И педали далеко.



Мне купили **паровозик**,
Сразу два вагона возит.
Я включаю его, и он
За собой везет вагон.
Вот бы мне в нём прокатиться,
Только жаль — не поместиться.



Нарисую я в альбоме
Голубое море,
Белый парус над волною
Бьётся на просторе.

Тема 3. Цвет. Шрифт. Границы. Размеры

(3.20) Стилевое свойство для указания цвета текста

Стилевое свойство **color** — цвет текста.

Может применяться практически к любому элементу, в котором содержится текст.

Возможные форматы значений:

color: red — указывается название цвета.

color: #FF4400 — указывается шестнадцатиричный код цвета

(в шестнадцатиричном коде цвета хорошо видны значения красной, зелёной и синей составляющих — **#FF4400** — каждая от 00 до FF).

color: rgb(80,50,40) или **color: rgb(50%,80%,5%)** — указываются красная, зелёная и синяя составляющие цвета — **rgb(128,255,40)** — каждая от 0 до 255 или от 0% до 100%.

color: rgba(128,255,40,0.5) или **color: rgba(10%,50%,20%,0.8)** — указываются красная, зелёная и синяя составляющие цвета и непрозрачность (дробное число от 0 до 1).

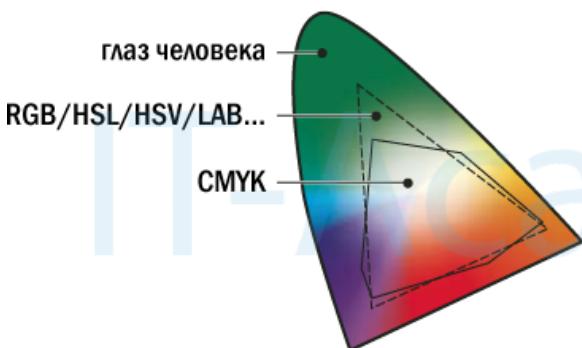
color: hsl(45,50%,20%) — указываются тон (угол на цветовом круге, от 0 до 360), насыщенность (в процентах от 0 до 100) и светлота (в процентах от 0 до 100).

color: hsla(250,50%,20%,0.5) — указываются тон, насыщенность и светлота, а также непрозрачность (дробное число от 0 до 1).

HTML-код	Отображение в браузере
<pre><p style="color: green">В 1933 г. первые за историю кинематографа публике была показана эротическая сцена &mdash; обнажённое купание в лесном озере в исполнении Хэди Ламарр, за что Адольф Гитлер назвал её врагом Третьего рейха.
А в 1941 году Хэди Ламарр запатентовала секретное средство связи, динамически изменяющее частоту вещания, которое в 1991 году легло в основу <b style="color: blue">Wi-Fi.</p></pre>	<p>В 1933 г. первые за историю кинематографа публике была показана эротическая сцена — обнажённое купание в лесном озере в исполнении Хэди Ламарр, за что Адольф Гитлер назвал её врагом Третьего рейха. А в 1941 году Хэди Ламарр запатентовала секретное средство связи, динамически изменяющее частоту вещания, которое в 1991 году легло в основу Wi-Fi.</p>

(3.25) Цветовой охват

цветовые модели



Источник: [Adobe Illustrator / О цвете](#)
<http://helpx.adobe.com/ru/illustrator/using/color.html>

устройства отображения



Источник: [Википедия / Цветовая модель](#)
https://ru.wikipedia.org/wiki/Цветовая_модель

(3.30) Именованные цвета

Основные цвета (CSS1 и CSS2):

black #000000 rgb(0,0,0)	silver #C0C0C0 rgb(192,192,192)	gray (grey) #808080 rgb(128,128,128)	white #FFFFFF rgb(255,255,255)	maroon #800000 rgb(128,0,0)	red #FF0000 rgb(255,0,0)	purple #800080 rgb(128,0,128)
fuchsia #FF00FF rgb(255,0,255)	green #008000 rgb(0,128,0)	lime #00FF00 rgb(0,255,0)	olive #808000 rgb(128,128,0)	yellow #FFFF00 rgb(255,255,0)	navy #000080 rgb(0,0,128)	blue #0000FF rgb(0,0,255)
teal #008080 rgb(0,128,128)	aqua #00FFFF rgb(0,255,255)	orange #FFA500 rgb(255,165,0)				

Цвета X11:

cyan #00FFFF rgb(0,255,255)	magenta #FF00FF rgb(255,0,255)	darkblue #00008B rgb(0,0,139)	darkcyan #008B8B rgb(0,139,139)	darkgreen #006400 rgb(0,100,0)	darkgray #A9A9A9 rgb(169,169,169)	darkmagenta #8B008B rgb(139,0,139)
darkorange #FF8C00 rgb(255,140,0)	darkred #8B0000 rgb(139,0,0)	lightblue #ADD8E6 rgb(173,216,230)	lightcyan #E0FFFF rgb(224,255,255)	lightgreen #90EE90 rgb(144,238,144)	lightgray #D3D3D3 rgb(211,211,211)	lightyellow #FFFFE0 rgb(255,255,224)
mediumblue #0000CD rgb(0,0,205)	mediumpurple #9370DB rgb(147,112,219)	aliceblue #F0F8FF rgb(240,248,255)	antiquewhite #FAEBD7 rgb(250,235,215)	aquamarine #7FFFDD rgb(127,255,212)	azure #F0FFFF rgb(240,255,255)	beige #F5F5DC rgb(245,245,220)
bisque #FFE4C4 rgb(255,228,196)	blanchedalmond #FFEBBC rgb(255,235,205)	blueviolet #8A2BE2 rgb(138,43,226)	brown #A52A2A rgb(165,42,42)	burlywood #DEB887 rgb(222,184,135)	cadetblue #5F9EA0 rgb(95,158,160)	chartreuse #7FFF00 rgb(127,255,0)
chocolate #D2691E rgb(210,105,30)	coral #FF7F50 rgb(255,127,80)	cornflowerblue #6495ED rgb(100,149,237)	cornsilk #FFF8DC rgb(255,248,220)	crimson #DC143C rgb(220,20,60)	darkgoldenrod #B8860B rgb(184,134,11)	darkkhaki #BDB76B rgb(189,183,107)
darkolivegreen #556B2F rgb(85,107,47)	darkorchid #9932CC rgb(153,50,204)	darksalmon #E9967A rgb(233,150,122)	darkseagreen #8FB8C8 rgb(143,188,143)	darkslateblue #483D8B rgb(72,61,139)	darkslategray #2F4F4F rgb(47,79,79)	darkturquoise #00CED1 rgb(0,206,209)
darkviolet #9400D3 rgb(148,0,211)	deeppink #FF1493 rgb(255,20,147)	deepskyblue #00BFFF rgb(0,191,255)	dimgray #696969 rgb(105,105,105)	dodgerblue #1E90FF rgb(30,144,255)	firebrick #B22222 rgb(178,34,34)	floralwhite #FFFFA0 rgb(255,250,240)
forestgreen #228B22 rgb(34,139,34)	gainsboro #DCDCDC rgb(220,220,220)	ghostwhite #F8F8FF rgb(248,248,255)	gold #FFD700 rgb(255,215,0)	goldenrod #DAA520 rgb(218,165,32)	greenyellow #ADFF2F rgb(173,255,47)	honeydew #F0FFF0 rgb(240,255,240)
hotpink #FF69B4 rgb(255,105,180)	indianred #CD5C5C rgb(205,92,92)	indigo #4B0082 rgb(75,0,130)	ivory #FFFFFF rgb(255,255,240)	khaki #F0E68C rgb(240,230,140)	lavender #E6E6FA rgb(230,230,250)	lavenderblush #FFF0F5 rgb(255,240,245)
lawngreen #7CFC00 rgb(124,252,0)	lemonchiffon #FFFFACD rgb(255,250,205)	lightcoral #F08080 rgb(240,128,128)	lightgoldenrodyellow #FAFAD2 rgb(250,250,210)	lightpink #FFB6C1 rgb(255,182,193)	lightsalmon #FFA07A rgb(255,160,122)	lightseagreen #20B2AA rgb(32,178,170)
lightskyblue #87CEFA rgb(135,206,250)	lightslategray #778899 rgb(119,136,153)	lightsteelblue #B0C4DE rgb(176,196,222)	limegreen #32CD32 rgb(50,205,50)	linen #FAF0E6 rgb(250,240,230)	mediumaquamarine #66CDA rgb(102,205,170)	mediumorchid #BA55D3 rgb(186,85,211)
mediumseagreen #3CB371 rgb(60,179,113)	mediumslateblue #7B68EE rgb(123,104,238)	mediumspringgreen #00FA9A rgb(0,250,154)	mediumturquoise #48D1CC rgb(72,209,204)	mediumvioletred #C71585 rgb(199,21,133)	midnightblue #191970 rgb(25,25,112)	mintcream #F5FFFA rgb(245,255,250)
mistyrose #FFE4E1 rgb(255,228,255)	moccasin #FFE4B5 rgb(255,228,181)	navajowhite #FFFDEAD rgb(255,222,183)	oldlace #FDF5E6 rgb(253,245,230)	olivedrab #6B8E23 rgb(107,142,35)	orangered #FF4500 rgb(255,69,0)	orchid #DA70D6 rgb(218,112,214)
palegoldenrod #EEE8AA rgb(238,232,170)	palegreen #98FB98 rgb(152,251,152)	paleturquoise #AFEEEE rgb(175,238,238)	palevioletred #DB7093 rgb(219,112,147)	papayawhip #FFEF05 rgb(255,239,213)	peachpuff #FFDAB9 rgb(255,218,185)	peru #CD853F rgb(205,133,63)
pink #FFC0CB rgb(255,192,203)	plum #DDA0DD rgb(221,160,221)	powderblue #B0E0E6 rgb(176,224,230)	rosybrown #BC8F8F rgb(188,143,143)	royalblue #4169E1 rgb(65,105,225)	saddlebrown #8B4513 rgb(139,69,19)	salmon #FA8072 rgb(250,128,114)
sandybrown #F4A460 rgb(244,164,96)	seagreen #2E8B57 rgb(46,139,87)	seashell #FFFF5EE rgb(255,245,238)	sienna #A0522D rgb(160,82,45)	skyblue #87CEEB rgb(135,206,235)	slateblue #6A5ACD rgb(106,90,205)	slategray #708090 rgb(112,128,144)
snow #FFFFFA rgb(255,250,250)	springgreen #00FF7F rgb(0,255,127)	steelblue #4682B4 rgb(70,130,180)	tan #D2B48C rgb(210,180,140)	thistle #D8BFD8 rgb(216,191,216)	tomato #FF6347 rgb(255,99,71)	turquoise #40E0D0 rgb(64,224,208)
violet #EE82EE rgb(238,130,238)	wheat #F5DEB3 rgb(245,222,179)	whitesmoke #F5F5F5 rgb(245,245,245)	yellowgreen #9ACD32 rgb(154,205,50)			

(3.35) Единицы измерения

Абсолютные единицы измерения

Задают точный физический размер. Применяются, как правило, если известны точные физические размеры устройства отображения (например, экрана дисплея или страницы принтера).

in	дюймы	1 дюйм = 2.54 см = 25.4 мм = 72 точки = 6 пик
cm	сантиметры	1 см = 10 мм = 0.39 дюйма = 2.36 пик = 28.35 точки
mm	миллиметры	1 мм = 0.1 см = 0.039 дюйма = 0.24 пик = 2.84 точки
pt	точки	1 точка = 1/12 пик = 1/72 дюйма = 0.035 см = 0.35 мм
pc	пики	1 пика = 12 точек = 1/6 дюйма = 0.423 см = 4.23 мм

Относительные единицы измерения

Задают размер относительно какого-либо другого размера.

em	высота буквы M текущего шрифта (т.е. размер текущего шрифта)
rem	высота буквы M шрифта тега <html>
ex	высота буквы x (высота строчных букв) текущего шрифта
px	пиксели (размер зависит от устройства отображения)
%	процент (размер зависит от значения другого свойства, всегда явно указанного в документации)
vw	1vw = 1/100 ширины окна браузера
vh	1vh = 1/100 высоты окна браузера
vmax	1vmax = 1/100 ширины или высоты окна браузера (смотря что больше)
vmin	1vmin = 1/100 ширины или высоты окна браузера (смотря что меньше) (поддержка: Chrome 31+, Firefox 31+)

Как это ни странно, но в CSS3:

1in равен 96px

Стилевое свойство **text-indent** — красная строка.

Задаёт размер отступа для первой строки текста.

Может применяться практически к любому **блочному** элементу, в котором содержится текст.

HTML-код	Отображение в браузере
<pre><p style="text-indent: 1cm">Челябинские детишки настолько суровы, что в песочницы им насыпают щебень.</p></pre>	Челябинские детишки настолько суровы, что в песочницы им насыпают щебень.

(3.45) Стилевые свойства для управления шрифтом

Стилевое свойство **font-size** — размер шрифта.

Задаёт размер шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

font-size: размер — явное задание размера. Если задано значение в процентах, процент высчитывается относительно размера шрифта контейнера.

font-size: xx-small — размер шрифта «очень мелкий», конкретный размер зависит от браузера.

Аналогично, допустимы значения **x-small, small, medium, large, x-large, xx-large**.

font-size: smaller — размер шрифта меньше чем у контейнера. Аналогично, допустимо значение **larger**.

HTML-код	Отображение в браузере
<pre><p style="font-size: 8px">Всегда внимательно читайте договор! Особенно то что написано мелким шрифтом! Размер имеет значение!</p></pre>	Всегда внимательно читайте договор! Особенно то что написано мелким шрифтом! Размер имеет значение!
<pre><p style="font-size: 17px"> Мухоморы — это чудо!
 Ядовиты? Ерунда!
 Всё равно их есть не буду,
 раз такая красота!</p></pre>	Мухоморы — это чудо! Ядовиты? Ерунда! Всё равно их есть не буду, раз такая красота!

Стилевое свойство **line-height** — высота строки (интервал между строками).

Задаёт минимальную высоту строк для шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

line-height: размер — явно задаёт высоту строки.

line-height: число — задаёт высоту строки, равную размеру шрифта, умноженному на указанное число.

line-height: normal — обычная высота строки, зависящая от размера шрифта.

HTML-код	Отображение в браузере
<pre><p style="line-height: 150%">Если у тебя спрошено будет: что полезнее, солнце или месяц? — ответствуй: месяц. Ибо солнце светит днём, когда и без того светло. </p></pre>	Если у тебя спрошено будет: что полезнее, солнце или месяц? — ответствуй: месяц. Ибо солнце светит днём, когда и без того светло.
<pre><p style="line-height: 14px">Налево пойдёшь — голову потеряешь, направо пойдёшь — коня потеряешь, прямо пойдёшь — всё потеряешь. Подумал Ильюша, и пошёл направо без коня.</p></pre>	Налево пойдёшь — голову потеряешь, направо пойдёшь — коня потеряешь, прямо пойдёшь — всё потеряешь. Подумал Ильюша, и пошёл направо без коня.

Стилевое свойство **font-weight** — жирность шрифта.

Задаёт жирность шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

font-weight: bold — жирный шрифт.

font-weight: normal — нормальный (нежирный) шрифт.

font-weight: bolder — жирнее чем у контейнера. Также допустимо значение **lighter** — светлее (менее жирно) чем у контейнера. Не рекомендуется использовать.

font-weight: число — явное задание жирности. Могут быть указаны значения 100, 200, 300, 400, 500, 600, 700, 800, 900. Их трактовка зависит от браузера, поэтому их не рекомендуется использовать.

HTML-код	Отображение в браузере
<pre><p style="font-weight: bold"> Мухоморы &mdash; это чудо!
 Ядовиты? Ерунда!
 Всё равно их есть не буду,
 раз такая красота!</p></pre>	<p>Мухоморы — это чудо! Ядовиты? Ерунда! Всё равно их есть не буду, раз такая красота!</p>

Стилевое свойство **font-style** — стиль шрифта.

Задаёт стиль шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

font-style: italic — курсивный шрифт. Его начертание, в общем случае, отличается от обычного, если используемый шрифт содержит символы такого начертания.

font-style: oblique — наклонный шрифт. Его начертание совпадает с обычным, символы обычного начертания просто наклоняются вправо.

Если используемый шрифт содержит не оба начертания (**italic** и **oblique**), а только одно из них (а таково большинство шрифтов), браузер будет использовать имеющееся начертание, независимо от того, какое значение установлено стилевому свойству **font-style** — **italic** либо **oblique**.

font-style: normal — обычный шрифт.

HTML-код	Отображение в браузере
<pre><div style="font-family: Times New Roman"> <p style="font-style: oblique">Хозяйке на заметку</p> <p style="font-style: italic">Если не пришивать верхние три пуговицы на блузку, можно сэкономить на туши для глаз и помаде.</p> </div></pre>	<p><i>Хозяйке на заметку</i></p> <p><i>Если не пришивать верхние три пуговицы на блузку, можно сэкономить на туши для глаз и помаде.</i></p>

Стилевое свойство **font-variant** — устанавливает способ отображения строчных букв.

Возможные значения:

font-variant: normal — строчные буквы отображаются обычным способом.

font-variant: small-caps — вместо строчных букв отображаются заглавные, уменьшенные до размера строчных.

HTML-код	Отображение в браузере
<code><p style="font-variant: normal"> БОЛЬШИЕ и маленькие буквы</p></code>	БОЛЬШИЕ и маленькие буквы
<code><p style="font-variant: small-caps"> БОЛЬШИЕ и маленькие буквы</p></code>	БОЛЬШИЕ И МАЛЕНЬКИЕ БУКВЫ

Стилевое свойство **letter-spacing** — интервал между буквами.

Задаёт интервал между буквами для шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

letter-spacing: размер — задаёт интервал, который может быть как положительным, так и отрицательным.

letter-spacing: normal — обычный интервал.

HTML-код	Отображение в браузере
<code><p style="letter-spacing: 2px">песок сквозь пальцы</p></code>	песок сквозь пальцы
<code><p style="letter-spacing: -1px">в тесноте, да не в обиде</p></code>	в тесноте, да не в обиде

Стилевое свойство **word-spacing** — интервал между словами.

Задаёт интервал между словами для шрифта, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

word-spacing: размер — задаёт интервал, который может быть как положительным, так и отрицательным.

word-spacing: normal — обычный интервал.

HTML-код	Отображение в браузере
<code><p style="word-spacing: 10px">песок сквозь пальцы</p></code>	песок сквозь пальцы
<code><p style="word-spacing: -4px">в тесноте, да не в обиде</p></code>	втесноте,даневобиде

(3.50) Имя шрифта

Стилевое свойство **font-family** — имя шрифта (шрифтов) и/или родовое имя шрифта.

Задаёт шрифт, которым будет отображаться содержимое элемента.

Может применяться практически к любому элементу, содержащему текст.

Возможные значения:

font-family: имя, имя, ..., родовое имя — как правило указывается одно-два имени шрифта, и затем — родовое имя шрифта.

Для каждого родового имени на компьютере пользователя обязательно найдётся шрифт; какой именно — зависит от браузера и используемой операционной системы.

Например:

имя	описание	вид
font-family: serif	шрифт с засечками	В моём браузере родовое имя serif отображается вот таким шрифтом.
font-family: sans-serif	шрифт без засечек	В моём браузере родовое имя sans-serif отображается вот таким шрифтом.
font-family: cursive	каллиграфический шрифт	В моём браузере родовое имя cursive отображается вот таким шрифтом.
font-family: fantasy	декоративный шрифт	В моём браузере родовое имя fantasy отображается вот таким шрифтом.
font-family: monospace	моноширинный шрифт	В моём браузере родовое имя monospace отображается вот таким шрифтом.

HTML-код	Отображение в браузере
<pre><p style="font-family: Times, serif">Чебурашка пишет письмо Гене:</p> <p style="font-family: Cheburashka, cursive">Гена, хотел тебе послать 10 долларов, но уже конверт запечатал...</p></pre>	Чебурашка пишет письмо Гене: Гена, хотел тебе послать 10 долларов, но уже конверт запечатал...

(3.52) Все реквизиты шрифта сразу

Стилевое свойство **font** позволяет задать несколько реквизитов шрифта сразу, одной записью. Размер шрифта и имя шрифта должны быть заданы обязательно.

Синтаксис:

font: [font-style] [font-variant] [font-weight] font-size [/line-height] font-family

HTML-код	Отображение в браузере
<pre><p style="font: italic bold 20px/200% Arial"> краткость &mdash; сестра таланта! </p></pre>	краткость — сестра таланта!

(3.55) Внешние шрифты

Одиночный тег **link** служит для организации различных взаимосвязей между документами.

Как правило, он включается в качестве дочернего в тег **head**.

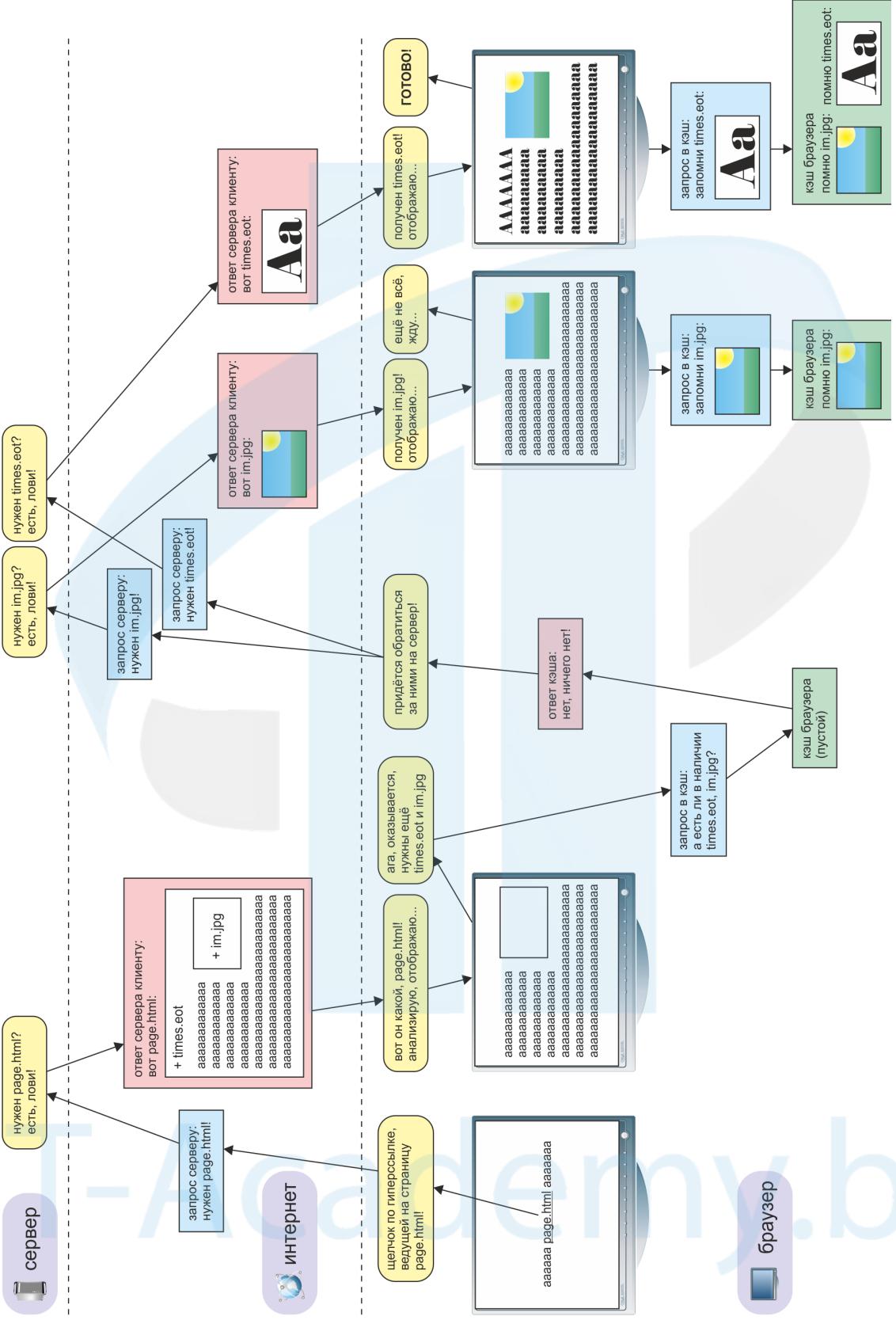
В случае подключения внешнего шрифта, данный тег используется, чтобы подключить внешние стилевые описания, позволяющие использовать внешний шрифт. Мы не включаем эти стилевые описания прямо на свою веб-страницу, т.к. они громоздкие.

HTML-код	Отображение в браузере
<pre><link href="http://fonts.googleapis.com/css?family=Yesteryear" rel="stylesheet" type="text/css"> <p style="font-family: Yesteryear, serif; font-size: 38px"> english letters русские буквы </p></pre>	english letters русские букивы

Поддержка браузерами различных форматов шрифтов:

	MSIE	Mozilla	Safari	Chrome	Opera	Android Browser	iOs Safari	Blackberry Browser
TTF (true type font)	9	+	3.1	+	10.1	2.2	4.3	7
OTF (open type font)	9	+	3.1	+	10.1	2.2	4.3	7
EOT (embedded open type)	6	+	3.1	+	9.6	3	3.2	7
WOFF (web open font format)	9	3.6	5.1	+	11.1	4.4	5.1	7
SVG (scalable vector graphics)	-	-	3.2	+	9.6	3	3.2	7

(3.57) Порядок взаимодействия браузера с сервером, если на странице используется изображения и шрифтовой файл



(3.70) Стилевые свойства для указания вида рамки

Стилевое свойство **border-style** — стиль рамки.

Стилевое свойство **border-top-style** — стиль верхней линии рамки.

Стилевое свойство **border-right-style** — стиль правой линии рамки.

Стилевое свойство **border-bottom-style** — стиль нижней линии рамки.

Стилевое свойство **border-left-style** — стиль левой линии рамки.

Задаёт стиль всей рамки или отдельной линии рамки.

Может применяться практически к любому элементу.

Возможные значения:

стиль	описание	вид
border-style: solid	сплошная рамка	
border-style: double	двойная рамка	
border-style: dotted	пунктирная рамка	
border-style: dashed	штриховая рамка	
border-style: groove	трёхмерная выемка	
border-style: ridge	трёхмерный выступ	
border-style: inset	трёхмерная врезка	
border-style: outset	трёхмерная вырезка	
border-style: hidden	скрытая рамка	
border-style: none	нет рамки	

HTML-код	Отображение в браузере
<pre><p style="border-style: double">Приём ведёт
доктор Айболит</p></pre>	

Может быть задано сразу четыре значения через пробел — в этом случае задаются отдельно стили верхней, правой, нижней и левой линий рамки.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid double solid double"> уважаемые пациенты!
 взяток не беру, конфеты не пью.</p></pre>	

По умолчанию стиль рамки — *none*.

HTML-код	Отображение в браузере
<pre><p style="border-left-style: dotted; border-top-style: dashed"> Время работы:
8:30-12:30</p></pre>	

Стилевое свойство **border-color** — цвет рамки.

Стилевое свойство **border-top-color** — цвет верхней линии рамки.

Стилевое свойство **border-right-color** — цвет правой линии рамки.

Стилевое свойство **border-bottom-color** — цвет нижней линии рамки.

Стилевое свойство **border-left-color** — цвет левой линии рамки.

Задаёт цвет всей рамки или отдельной линии рамки.

Может применяться практически к любому элементу.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-color: red">Приём ведёт
доктор Айболит</p></pre>	Приём ведёт доктор Айболит

Может быть задано сразу четыре значения через пробел — в этом случае задаются отдельно цвета верхней, правой, нижней и левой линий рамки.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-color: green blue red yellow"> уважаемые пациенты!
входите только по одному.

взяток не беру,
конфеты не пью.</p></pre>	уважаемые пациенты! входите только по одному. взяток не беру, конфеты не пью.

По умолчанию цвет рамки — чёрный.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-left-color: red"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00

Важно: установленный цвет рамки отобразится только в том случае, когда рамка видима, т.е. имеет стиль не **none** и не **hidden**.

HTML-код	Отображение в браузере
<pre><p style="border-color: red"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00
<pre><p style="border-left-style: solid; border-color: red"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00

Стилевое свойство **border-width** — толщина рамки.

Стилевое свойство **border-top-width** — толщина верхней линии рамки.

Стилевое свойство **border-right-width** — толщина правой линии рамки.

Стилевое свойство **border-bottom-width** — толщина нижней линии рамки.

Стилевое свойство **border-left-width** — толщина левой линии рамки.

Задаёт толщину рамки или отдельной линии рамки.

Может применяться практически к любому элементу.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-width: 1px">Приём ведёт
доктор Айболит</p></pre>	Приём ведёт доктор Айболит
<pre><p style="border-style: solid; border-width: 4px">Приём ведёт
доктор Айболит</p></pre>	Приём ведёт доктор Айболит

Может быть задано сразу четыре значения через пробел — в этом случае задаются отдельно толщина верхней, правой, нижней и левой линий рамки.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-width: 1px 3px 1px 3px"> уважаемые пациенты!
входите только по одному.

взяток не беру,
конфеты не пью.</p></pre>	уважаемые пациенты! входите только по одному. взяток не беру, конфеты не пью.

По умолчанию толщина рамки — 3px.

HTML-код	Отображение в браузере
<pre><p style="border-style: solid; border-bottom-width: 1px"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00

Важно: установленная толщина рамки отобразится только в том случае, когда рамка видима, т.е. имеет стиль не **none** и не **hidden**.

HTML-код	Отображение в браузере
<pre><p style="border-width: 1px"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00
<pre><p style="border-left-style: solid; border-width: 3px"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00

Стилевое свойство **border** — вид рамки.

Стилевое свойство **border-top** — вид верхней линии рамки.

Стилевое свойство **border-right** — вид правой линии рамки.

Стилевое свойство **border-bottom** — вид нижней линии рамки.

Стилевое свойство **border-left** — вид левой линии рамки.

Задаёт сразу стиль, цвет и толщину рамки или отдельной линии рамки.

Может применяться практически к любому элементу.

HTML-код	Отображение в браузере
<pre><p style="border: solid red 1px">Приём ведёт
доктор Айболит</p></pre>	Приём ведёт доктор Айболит
<pre><p style="border-left: dashed red 2px">уважаемые пациенты!
входите только по одному.

взяток не беру,
конфеты не пью.</p></pre>	уважаемые пациенты! входите только по одному. взяток не беру, конфеты не пью.
<pre><p style="border-left: dotted red 2px; border-top: dashed green 2px"> уважаемые пациенты!
входите только по одному.

взяток не беру,
конфеты не пью.</p></pre>	уважаемые пациенты! входите только по одному. взяток не беру, конфеты не пью.

Можно задать стиль, цвет, толщину или сразу все свойства вида **всей** рамки, а потом для отдельных линий переопределить некоторые свойства.

HTML-код	Отображение в браузере
<pre><p style="border: solid green 2px; border-top-style: dotted; border-bottom-color: yellow"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00
<pre><p style="border-style: solid; border-left-style: dotted; border-color: red; border-right-color: green"> Время работы:
8:30-12:30
13:30-17:00</p></pre>	Время работы: 8:30-12:30 13:30-17:00

Стилевое свойство **color** задаёт не только цвет текста, но и умолчательный цвет рамки, а также цвет маркеров списка.

HTML-код	Отображение в браузере
<pre><div style="color: red; border-style: solid"> Время работы: 8:30-12:3013:30-17:00
<="" div><="" pre=""/></pre>	Время работы: <ul style="list-style-type: none">• 8:30-12:30• 13:30-17:00

(3.75) Управление размерами

Стилевое свойство **width** — внутренняя ширина тега.

Стилевое свойство **height** — внутренняя высота тега.

Могут применяться практически к любому блочному элементу.

Задают внутреннюю ширину и высоту блочного элемента.

*Если значение ширины и/или высоты не задано, оно считается заданным как **auto**.*

Возможные значения:

width: размер — ширина явно задана. При указании размера в процентах, процент вычисляется относительно ширины контейнера.

width: auto — ширина подбирается автоматически.

HTML-код	Отображение в браузере
<pre><p style="border: solid red 1px; width: 150px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.</p></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.
<pre><p style="border: solid red 1px; width: 220px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.</p></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.
<pre><p style="border: solid red 1px; width: 220px; height: 150px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.</p></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.

IT-Academy.by

Стилевое свойство **min-width** — минимальная внутренняя ширина.

Стилевое свойство **max-width** — максимальная внутренняя ширина.

Стилевое свойство **min-height** — минимальная внутренняя высота.

Стилевое свойство **max-height** — максимальная внутренняя высота.

*Могут применяться практически к любому **блочному элементу**.*

*Задают максимальную или минимальную **внутреннюю** ширину и высоту блочного элемента.*

Возможные значения:

min-width: размер — минимальная ширина явно задана.

HTML-код	Отображение в браузере
<pre><p style="border: solid red 1px; min-height: 40px"> Колл-центр Velcom</p></pre>	Колл-центр Velcom
<pre><p style="border: solid red 1px; min-height: 40px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.</p></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Образовался он в 2000 году со штатом в два сотрудника у одного телефонного аппарата.

Домашнее задание №3

Создать новый проект FONT_GWF.

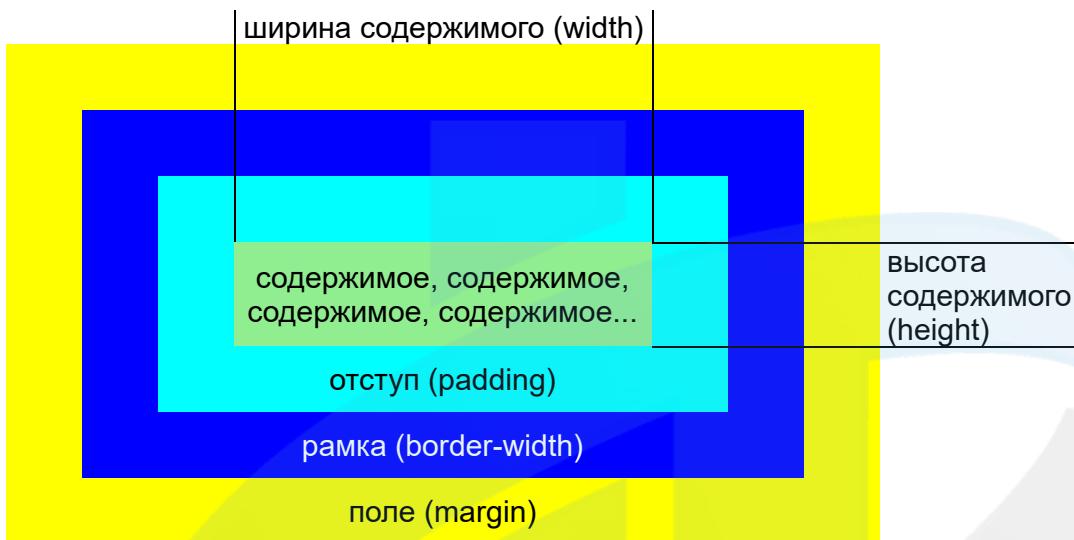
В Google Fonts выбрать и настроить два любых красивых **кириллических** шрифта, сверстать этими шрифтами любой текст (стихотворение, текст песни, отрывок книги).

IT-Academy.by

Тема 4. Отступы. Фон. Таблицы

(4.10) Поля и отступы

Схема расположения полей, рамок и отступов:



padding — отступ от рамки элемента (*border*) до содержимого элемента.

margin — наружное поле от рамки элемента (*border*) до рамок соседних элементов или до рамок контейнера.

Стилевое свойство **padding** — отступы.

Стилевое свойство **padding-top** — отступ сверху.

Стилевое свойство **padding-right** — отступ справа.

Стилевое свойство **padding-bottom** — отступ снизу.

Стилевое свойство **padding-left** — отступ слева.

Могут применяться практически к любому **блочному** элементу.

Может быть задано четыре значения через пробел, в этом случае задаются отступы верхний, правый, нижний, левый соответственно.

HTML-код	Отображение в браузере
<pre><p style="padding-left: 38px; border: solid red 1px"> Колл-центр оператора velcom &mdash; один из крупных в Беларуси.</p></pre>	Колл-центр оператора velcom — один из крупных в Беларуси.
<pre><p style="padding: 20px; border: solid red 1px"> Колл-центр оператора velcom &mdash; один из крупных в Беларуси.</p></pre>	Колл-центр оператора velcom — один из крупных в Беларуси.
<pre><p style="padding: 10px 40px 10px 10px; border: solid red 1px"> Колл-центр оператора velcom &mdash; один из крупных в Беларуси.</p></pre>	Колл-центр оператора velcom — один из крупных в Беларуси.

Стилевое свойство **margin** — поля.

Стилевое свойство **margin-top** — поле сверху.

Стилевое свойство **margin-right** — поле справа.

Стилевое свойство **margin-bottom** — поле снизу.

Стилевое свойство **margin-left** — поле слева.

Могут применяться практически к любому **блочному** элементу;

margin-left и margin-right могут также применяться к строчным элементам.

Может быть задано четыре значения через пробел, в этом случае задаются поля верхнее, правое, нижнее, левое соответственно.

HTML-код	Отображение в браузере
<pre><div style="margin-left: 38px; border: solid red 1px"> Колл-центр velcom был одним из первых крупных в Беларуси.</div></pre>	Колл-центр velcom был одним из первых крупных в Беларуси.
<pre><div style="margin: 10px; border: solid red 1px"> Колл-центр velcom был одним из первых крупных в Беларуси.</div></pre>	Колл-центр velcom был одним из первых крупных в Беларуси.

Поля могут быть отрицательными.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; margin-bottom: -10px"> Колл-центр velcom был одним из первых крупных в Беларуси.</div> <div style="border: solid blue 1px"> Публичный договор об оказании услуги электросвязи...</div></pre>	Колл-центр velcom был одним из первых крупных в Беларуси. Публичный договор об оказании услуги электросвязи...

Поле — это минимальное расстояние от рамки данного элемента до рамок других элементов, поэтому, поля соседних по вертикали элементов схлестываются, насколько возможно.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; margin-bottom: 10px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси.</div> <div style="border: solid blue 1px"> Публичный договор об оказании услуги электросвязи...</div></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Публичный договор об оказании услуги электросвязи...
<pre><div style="border: solid red 1px; margin-bottom: 10px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси.</div> <div style="border: solid blue 1px; margin-top: 3mm"> Публичный договор об оказании услуги электросвязи...</div></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Публичный договор об оказании услуги электросвязи...
<pre><div style="border: solid red 1px; margin-bottom: 10px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси.</div> <div style="border: solid blue 1px; margin-top: 5mm"> Публичный договор об оказании услуги электросвязи...</div></pre>	Колл-центр оператора velcom был одним из первых крупных в Беларуси. Публичный договор об оказании услуги электросвязи...

Поле (margin) элемента не схлестывается с отступом (padding) его контейнера.

HTML-код	Отображение в браузере
<pre><div style="padding: 15px; border: solid green 1px"> <div style="border: solid red 1px; margin-bottom: 15px"> Колл-центр velcom </div> </div></pre>	<p>Колл-центр velcom</p>

Если контейнер элемента с соответствующей стороны имеет рамку (border) или отступ (padding), то поле (margin) элемента отсчитывается до рамки контейнера.

HTML-код	Отображение в браузере
<pre><div style="border: solid green 3px"> <div style="border: solid red 1px; margin-bottom: 20px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. </div> </div> <div> <div style="border: solid blue 1px; margin-top: 20px"> Публичный договор об оказании услуги электросвязи является соглашением </div> </div></pre>	<p>Колл-центр оператора velcom был одним из первых крупных в Беларуси.</p> <p>Публичный договор об оказании услуги электросвязи является соглашением</p>
<pre><div style="padding: 1px"> <div style="border: solid red 1px; margin-bottom: 20px"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. </div> </div> <div> <div style="border: solid blue 1px; margin-top: 20px"> Публичный договор об оказании услуги электросвязи является соглашением </div> </div></pre>	<p>Колл-центр оператора velcom был одним из первых крупных в Беларуси.</p> <p>Публичный договор об оказании услуги электросвязи является соглашением</p>

Если контейнер элемента с соответствующей стороны не имеет ни рамки (border), ни отступа (padding), то поля (margin) элемента выходят за пределы контейнера, схлестываются с полями (margin) самого контейнера, и схлестываются с полями (margin) соседних элементов.

HTML-код	Отображение в браузере
<pre><div style="margin-bottom: 5mm"> <p style="border: solid red 1px; margin-bottom: 5mm"> Колл-центр оператора velcom был одним из первых крупных в Беларуси. </p> </div> <div style="margin-top: 5mm"> <p style="border: solid blue 1px; margin-top: 5mm"> Публичный договор об оказании услуги электросвязи является соглашением </p> </div></pre>	<p>Колл-центр оператора velcom был одним из первых крупных в Беларуси.</p> <p>Публичный договор об оказании услуги электросвязи является соглашением</p>

(4.20) Выравнивание содержимого контейнера

Стилевое свойство **text-align** — горизонтальное выравнивание содержимого.

Может применяться практически к любому **блочному** элементу.

Возможные значения:

text-align: left — выравнивание по левой стороне.

text-align: right — выравнивание по правой стороне.

text-align: center — выравнивание по центру.

text-align: justify — выравнивание по ширине.

Если содержимое **является текстом**, задаёт выравнивание строк текста.

HTML-код	Отображение в браузере
<pre><p style="border: solid red 1px; text-align: left"> Первая стадия ношения носков — когда носки ставят рядом с ботинками.</p></pre>	Первая стадия ношения носков — когда носки ставят рядом с ботинками.
<pre><p style="border: solid red 1px; text-align: center"> Вторая стадия ношения носков — когда стригут ногти, не снимая носков.</p></pre>	Вторая стадия ношения носков — когда стригут ногти, не снимая носков.
<pre><p style="border: solid red 1px; text-align: right"> Третья стадия ношения носков — когда снимают носки, не снимая ботинок.</p></pre>	Третья стадия ношения носков — когда снимают носки, не снимая ботинок.
<pre><p style="border: solid red 1px; text-align: justify"> Чистоплотные люди не меняют носки. Им противно руками трогать грязную вещь. И не надо их трогать, постепенно носки тихо сами истлеют и от ног отпадут.
(Г.Остер)</p></pre>	Чистоплотные люди не меняют носки. Им противно руками трогать грязную вещь. И не надо их трогать, постепенно носки тихо сами истлеют и от ног отпадут. (Г.Остер)

Если содержимое **не является текстом**, задаёт расположение содержимого внутри данного элемента.

HTML-код	Отображение в браузере
<pre><div style="border: solid green 1px; text-align: left"> </div></pre>	
<pre><div style="border: solid green 1px; text-align: center"> </div></pre>	
<pre><div style="border: solid green 1px; text-align: right"> </div></pre>	

Стилевое свойство **vertical-align** при применении к строчному элементу задаёт его вертикальное выравнивание внутри строки текста, содержащей его.

Некоторые возможные значения:

vertical-align: top — выравнивание к верхней стороне контейнера.

vertical-align: bottom — выравнивание к нижней стороне контейнера.

vertical-align: baseline — выравнивание к базовой линии текста контейнера (если текста нет, базовая линия есть нижняя сторона контейнера).

vertical-align: отступ — отступ вверх от базовой линии контейнера.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	
<pre><div style="border: solid red 1px"> бу </div></pre>	

Стилевое свойство **vertical-align** при применении к блочному элементу задаёт вертикальное выравнивание его содержимого (при этом высота блочного элемента не должна быть задана явно, иначе стилевое свойство не сработает).

Некоторые возможные значения:

vertical-align: top — выравнивание содержимого к верхней стороне.

vertical-align: middle — выравнивание содержимого по центру.

vertical-align: bottom — выравнивание содержимого к нижней стороне.

vertical-align: top	vertical-align: middle	vertical-align: bottom
		

IT-Academy.by

(4.25) Управление пробелами и переносами строк

Стилевое свойство **white-space** — вид обработки пробелов и переносов строк.

Может применяться практически к любому **блочному элементу**.

Возможные значения:

white-space: normal — пробелы и переносы строк отображаются обычным образом, то есть:

- несколько пробелов в HTML-коде объединяются в один;
- переносы строк в HTML-коде аналогичны пробелам и не вызывают перенос строки в отображаемом документе;
- автоматический перенос строк (если строка дошла до правого края доступного ей на веб-странице места — она разрывается там, где покажется браузеру оптимальным, например на месте одного из пробелов или после дефиса).

white-space: pre — пробелы и переносы строк отображаются в документе точно так как набраны в HTML-коде; автоматического переноса строк не происходит.

white-space: pre-wrap — пробелы и переносы строк отображаются в документе точно так как набраны в HTML-коде; автоматический перенос строк происходит.

white-space: nowrap — запрет переноса строк, т.е. весь текст с таким стилем отобразится в одну строку.

HTML-код	Отображение в браузере
<pre><div style="white-space: normal">Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери.</div></pre>	Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери.
<pre><div style="white-space: pre">Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери.</div></pre>	Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери.
<pre><div style="white-space: pre-wrap">Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери. </div></pre>	Сила характера заключается не в умении пробивать головой стены, а в настойчивости в поисках двери.
<pre><div style="border: solid red 1px; width: 130px"> <div style="white-space: normal"> Сила характера заключается не...</div> <div style="white-space: nowrap"> Сила характера заключается не...</div> </div></pre>	Сила характера заключается не... Сила характера заключается не...

(4.30) Стилевые свойства для управления фоном

Стилевое свойство **background-color** — цвет фона элемента.

Может применяться практически к любому элементу.

Возможные значения:

background-color: цвет — явное задание цвета.

background-color: transparent — фон прозрачный.

По умолчанию, цвет любого фона — **transparent**.

HTML-код	Отображение в браузере
<pre><p style="background-color: yellow">Кому на Руси жить хорошо, тот уж давно на Руси не живёт.</p></pre>	Кому на Руси жить хорошо, тот уж давно на Руси не живёт.

Стилевое свойство **background-image** — изображение на фоне элемента.

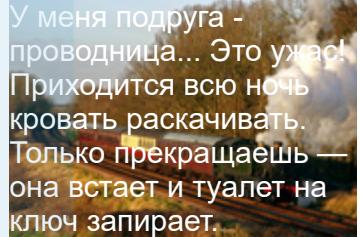
Может применяться практически к любому элементу.

Возможные значения:

background-image: url("URL") — явное задание URLа изображения для фона.

background-image: none — нет изображения на фоне.

По умолчанию, изображение фона отсутствует, т.е. **background-image: none**.

HTML-код	Отображение в браузере
<pre><p style="background-image: url('images/train.jpg'); color: white"> У меня подруга - проводница... Это ужас! Приходится всю ночь кровать раскачивать. Только прекращаешь — она встает и туалет на ключ запирает.</p></pre>	 У меня подруга - проводница... Это ужас! Приходится всю ночь кровать раскачивать. Только прекращаешь — она встает и туалет на ключ запирает.

Стилевое свойство **background-repeat** — повторение изображения на фоне элемента.

Может применяться практически к любому элементу.

Имеет смысл, только если свойство **background-image** задано.

Возможные значения:

background-repeat: no-repeat — не повторять.

background-repeat: repeat-x — повторять по горизонтали.

background-repeat: repeat-y — повторять по вертикали.

background-repeat: repeat — повторять по вертикали и горизонтали.

По умолчанию, повторение задано как **repeat**, т.е. заданное

в **background-image** изображение заполняет всю площадь фона элемента.

HTML-код	Отображение в браузере
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: no-repeat"></pre> <p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></p>	
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: repeat-x"></pre> <p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></p>	
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: repeat-y"></pre> <p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></p>	
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: repeat"></pre> <p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></p>	

Стилевое свойство **background-position** — позиция изображения на фоне элемента.

Может применяться практически к любому элементу.

Имеет смысл, только если свойство **background-image** задано.

Возможные значения:

background-position: left top — слева вверху контейнера.

background-position: center top — по центру вверху контейнера.

background-position: right top — справа вверху контейнера.

background-position: left center — слева по центру контейнера.

background-position: center center — по центру контейнера.

background-position: right center — справа по центру контейнера.

background-position: left bottom — слева внизу контейнера.

background-position: center bottom — по центру внизу контейнера.

background-position: right bottom — справа внизу контейнера.

background-position: размер размер — произвольная абсолютная позиция внутри контейнера.

background-position: процент процент — произвольная относительная позиция внутри контейнера.

По умолчанию, позиция задана как **left top**.

HTML-код	Отображение в браузере
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: no-repeat; background-position: center center"> Сэр Тимоти Джон Бернерс-Ли &mdash; британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></pre>	<p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p>
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: no-repeat; background-position: 20px 20px"> Сэр Тимоти Джон Бернерс-Ли &mdash; британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></pre>	<p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p>
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: no-repeat; background-position: 80% 30%"> Сэр Тимоти Джон Бернерс-Ли &mdash; британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p></pre>	<p>Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.</p>

В случае указания **background-position** в процентах, указанная процентами точка фонового изображения позиционируется в указанную этими же процентами точку контейнера.

Например, если указано background-position: 20% 70%, точка **фонового изображения**, находящаяся на 20% его ширины и на 70% его высоты, будет спозиционирована в точку **контейнера**, находящуюся на 20% его ширины и на 70% его высоты.

HTML-код	Отображение в браузере
<pre><div style="width: 501px; height: 501px; background: url('images/bg_blue.png') no-repeat 100% 100%"> <div style="width: 501px; height: 501px; background: url('images/bg_green.png') no-repeat 0% 20%"> <div style="width: 501px; height: 501px; background: url('images/bg_pink.png') no-repeat 20% 100%"> <div style="width: 501px; height: 501px; background: url('images/bg_yellow.png') no-repeat 80% 40%"> <div style="width: 501px; height: 501px; background: url('images/bg_big.png')"> </div></div></div></div></pre>	

Здесь для общего фона использовано крупное изображение (*bg_big.png*, синяя сетка), разбитое на сектора, чтобы можно было визуально определить, **в какую точку контейнера** указывает *background-position*.

Затем, в различных позициях, заданных процентами, расположены 4 фоновых изображения, каждое из которых также разбито на сектора, чтобы можно было визуально определить, **в какую точку фонового изображения** указывает *background-position*.

Если задано ещё и повторение фона, то ***background-position*** задаёт позицию одного из изображений, относительно которого повторяются остальные.

HTML-код	Отображение в браузере
<pre><p style="border: solid blue 1px; background-image: url('images/flower.jpg'); background-repeat: repeat-x; background-position: 40px 20px"> Сэр Тимоти Джон Бернерс- Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины и действующий глава Консорциума Всемирной паутины.</p></pre>	

Если изображение фона имеет прозрачные части, через них будет виден цвет фона — ***background-color***, если он задан.

HTML-код	Отображение в браузере
<pre><p style="background-image: url('images/gerb.png')> Считается, что прообразом графа Дракулы послужила реальная историческая личность — Влад III Цепеш, господарь средневекового княжества Валахия. </p></pre>	
<pre><p style="background-image: url('images/gerb.png'); background-color: yellow"> Считается, что прообразом графа Дракулы послужила реальная историческая личность — Влад III Цепеш, господарь средневекового княжества Валахия. </p></pre>	

Если изображение фона заполняет не весь фон (т.е. при ***background-repeat*** не равно ***repeat***), через незаполненные изображением части фона будет виден цвет фона — ***background-color***, если он задан.

HTML-код	Отображение в браузере
<pre><p style="border: solid blue 1px; background-color: yellow; background-image: url('images/flower.jpg'); background-repeat: no-repeat"> Сэр Тимоти Джон Бернерс-Ли — британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины.</p></pre>	

Стилевое свойство **background-attachment** позволяет задать поведение фона, если содержимое элемента скроллируется.

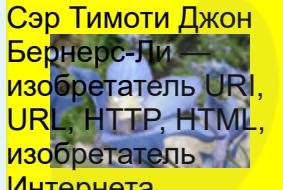
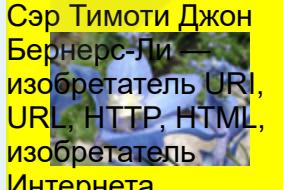
Возможные значения:

background-attachment: scroll — фон прокручивается вместе с содержимым элемента.

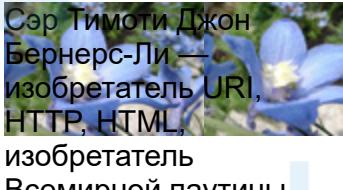
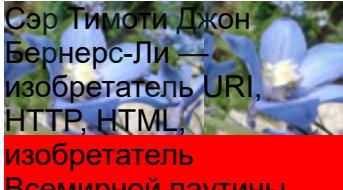
background-attachment: fixed — фон остаётся на месте, даже если содержимое элемента прокручивается.

Значение по умолчанию — **scroll**.

Стилевое свойство **background** — позволяет задать сразу все стилевые свойства фона.

HTML-код	Отображение в браузере
<pre><p style="background-image: url('images/flower.jpg'); background-repeat: no-repeat; background-position: center center; background-color: yellow"> Сэр Тимоти Джон Бернерс-Ли &mdash; изобретатель URI, URL, HTTP, HTML, изобретатель Интернета.</p></pre>	 <p>Сэр Тимоти Джон Бернерс-Ли — изобретатель URI, URL, HTTP, HTML, изобретатель Интернета.</p>
<pre><p style="background: url('images/flower.jpg') no-repeat center center yellow"> Сэр Тимоти Джон Бернерс-Ли &mdash; изобретатель URI, URL, HTTP, HTML, изобретатель Интернета.</p></pre>	 <p>Сэр Тимоти Джон Бернерс-Ли — изобретатель URI, URL, HTTP, HTML, изобретатель Интернета.</p>

Перечислять в **background** значения для всех стилевых свойств фона необязательно, но, если среди значений **background** значения каких-либо стилевых свойства фона не указаны, эти стилевые свойства получат свои умолчательные значения, а не сохранят установленные ранее значения.

HTML-код	Отображение в браузере
<pre><p style="background-color: red; background: url('images/flower.jpg') repeat-x"> Сэр Тимоти Джон Бернерс-Ли &mdash; изобретатель URI, HTTP, HTML, изобретатель Всемирной паутины.</p></pre>	 <p>Сэр Тимоти Джон Бернерс-Ли — изобретатель URI, HTTP, HTML, изобретатель Всемирной паутины.</p>
<pre><p style="background: url('images/flower.jpg') repeat-x; background-color: red"> Сэр Тимоти Джон Бернерс-Ли &mdash; изобретатель URI, HTTP, HTML, изобретатель Всемирной паутины.</p></pre>	 <p>Сэр Тимоти Джон Бернерс-Ли — изобретатель URI, HTTP, HTML, изобретатель Всемирной паутины.</p>

(4.32) Резиновый фон

Резиновый фон — фон, адаптирующий свои размеры под содержимое контейнера.

The diagram shows two examples of responsive backgrounds using pencil images:

Школьник!
ЖИ и ШИ
пиши с буквой И!

Мальчики!
Не дёргайте девочек
за косички!
Они вырастают
стоматологами!

План действий:

Используются картинки:

- top.png
- middle.png
- bottom.png

HTML-код	Отображение в браузере
<pre><div style="padding-left: 70px; font-size: 15px; line-height: 180%; border: solid blue 1px"> Женщины!
Мужайтесь!
Мужчины!
Женитесь!</div></pre>	
<pre><div style="background: url('images/pen/middle.png') repeat-y"> <div style="padding-left: 70px; font-size: 15px; line-height: 180%; border: solid blue 1px"> Женщины!
Мужайтесь!
Мужчины!
Женитесь!</div> </div></pre>	
<pre><div style="background: url('images/pen/middle.png') repeat-y"> <div style="background: url('images/pen/top.png') no-repeat left top"> <div style="padding-left: 70px; font-size: 15px; line-height: 180%; border: solid blue 1px"> Женщины!
Мужайтесь!
Мужчины!
Женитесь!</div> </div></div></pre>	
<pre><div style="background: url('images/pen/middle.png') repeat-y"> <div style="background: url('images/pen/top.png') no-repeat left top"> <div style="background: url('images/pen/bottom.png') no-repeat left bottom"> <div style="padding-left: 70px; font-size: 15px; line-height: 180%; border: solid blue 1px"> Женщины!
Мужайтесь!
Мужчины!
Женитесь!</div> </div></div></pre>	

(4.35) Переполнение контейнера

Стилевое свойство **overflow** — отображение переполнения.

Задаёт способ отображения содержимого контейнера, если размер контейнера ограничен (например стилевыми свойствами **width** и/или **height**), а содержимое оказывается по размеру больше чем контейнер.

Может применяться практически к любому блочному элементу.

Возможные значения:

overflow: visible — содержимое отображается за границами контейнера.

overflow: hidden — содержимое обрезается границами контейнера.

overflow: scroll — содержимое обрезается, и в контейнере появляются вертикальная и горизонтальная полосы прокрутки для просмотра всего содержимого.

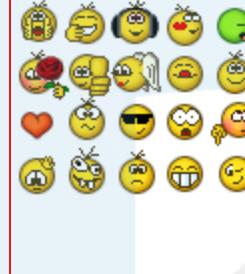
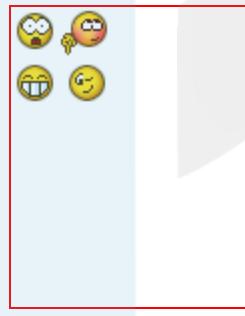
overflow: auto — содержимое обрезается, и в контейнере появляются только нужные полосы прокрутки для просмотра всего содержимого.

HTML-код	Отображение в браузере
<pre><div style="width: 150px; height: 140px; border: solid red 1px; overflow: visible"> </div></pre>	
<pre><div style="width: 150px; height: 65px; border: solid red 1px; overflow: hidden"> </div></pre>	
<pre><div style="width: 150px; height: 160px; border: solid red 1px; overflow: scroll"> </div></pre>	
<pre><div style="width: 150px; height: 160px; border: solid red 1px; overflow: auto"> </div></pre>	

В CSS3 также есть стилевые свойства **overflow-x** и **overflow-y**, позволяющие задать отображение переполнения раздельно по горизонтали и по вертикали.

(4.40) Спрайты

Пошаговая инструкция для создания спрайта.
(Спрайты на спрайт-листе расположены с шагом 25px)

HTML-код	Отображение в браузере
	
<div style="background-image: url('images/smileys.png'); background-repeat: no-repeat; width: 150px; height: 150px; border: solid red 1px"></div>	
<div style="background-image: url('images/smileys.png'); background-repeat: no-repeat; background-position: -75px -50px; width: 150px; height: 150px; border: solid red 1px"></div>	
<div style="background-image: url('images/smileys.png'); background-repeat: no-repeat; background-position: -75px -50px; width: 25px; height: 25px; border: solid red 1px"></div>	
— Вы делаете операции по увеличению глаз? — Да, стоимость — 100.000\$. <div style="background-image: url('images/smileys.png'); background-repeat: no-repeat; background-position: -75px -50px; width: 25px; height: 25px;"></div> — С вас 100.000\$.	<p>— Вы делаете операции по увеличению глаз? — Да, стоимость — 100.000\$.</p> <p>— С вас 100.000\$.</p> 

Если бы можно было увидеть сдвинутое фоновое изображение за пределами спрайта на шаге 4, оно располагалось бы так:



(4.70) Таблицы — основные теги и атрибуты тегов

Парный тег **table** — таблица.

Атрибут **border** тега **table** — толщина рамок.

Парный тег **tr** внутри тега **table** — строка таблицы.

Парный тег **td** внутри тега **tr** — ячейка в строке таблицы.

Парный тег **th** внутри тега **tr** — ячейка в строке таблицы.

Таблица (table) должна содержать одну или несколько строк (tr).

Каждая строка (tr) должна содержать одну или несколько ячеек (td или th).

Ячейка th используется для оформления заголовочных ячеек — обычно содержимое такой ячейки выводится жирным шрифтом и выровнено по центру.

HTML-код	Отображение в браузере						
<pre><table border=1> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<table border="1"> <thead> <tr><th>англ</th><th>рус</th></tr> </thead> <tbody> <tr><td>red</td><td>красный</td></tr> <tr><td>green</td><td>зелёный</td></tr> </tbody> </table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						
<pre><table border=1> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td style="width: 150px">red</td> <td>красный</td> </tr> <tr> <td>green</td> <td style="background-color: green"> зелёный </td> </tr> </table></pre>	<table border="1"> <thead> <tr><th>англ</th><th>рус</th></tr> </thead> <tbody> <tr><td>red</td><td>красный</td></tr> <tr><td>green</td><td>зелёный</td></tr> </tbody> </table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						

Таблица, строка, ячейка могут иметь независимые фоны. Браузер сначала выводит фон всей таблицы (если есть), затем накладывает фоны строк (если есть), затем — фоны ячеек (если есть).

HTML-код	Отображение в браузере
<pre><table border=1 style="background-image: url('images/Beatles2.jpg')"> <tr style="height: 50px"> <th>рус</th> <th style="width: 100px">мяч</th> </tr> <tr style="height: 100px; background-color: rgba(255,0,0,0.5)"> <td>красный</td> <td style="background-image: url('images/ball3.png')"></td> </tr> <tr style="height: 100px; background-color: rgba(0,255,0,0.5)"> <td>зелёный</td> <td style="background-image: url('images/ball3.png')"></td> </tr> </table></pre>	

(4.72) Таблицы — дополнительные теги и атрибуты тегов

Парный тег **caption** внутри тела **table** — заголовок таблицы.

Заголовок таблицы будет отображаться над таблицей.

HTML-код	Отображение в браузере						
<pre><table border=1> <caption>Учим английский язык вместе</caption> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<p>Учим английский язык вместе</p> <table border="1"><thead><tr><th>англ</th><th>рус</th></tr></thead><tbody><tr><td>red</td><td>красный</td></tr><tr><td>green</td><td>зелёный</td></tr></tbody></table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						

Стилевое свойство **caption-side** тела **caption** позволяет установить, сверху или снизу таблицы отображается заголовок.

Возможные значения:

caption-side: top (значение по умолчанию) — заголовок таблицы отображается над таблицей.

caption-side: bottom — заголовок таблицы отображается под таблицей.

HTML-код	Отображение в браузере						
<pre><table border=1> <caption style="caption-side: bottom; text-align: left"> Учим английский язык вместе </caption> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<table border="1"><thead><tr><th>англ</th><th>рус</th></tr></thead><tbody><tr><td>red</td><td>красный</td></tr><tr><td>green</td><td>зелёный</td></tr></tbody></table> <p>Учим английский язык вместе</p>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						

Одиночный тег **col** внутри тела **table** — колонка таблицы.

Не отображается.

Служит для установки стилевых свойств всему столбцу ячеек таблицы.

Поддерживаются только стилевое свойство *width* и группа стилевых свойств по управлению фоном (*background-**).

HTML-код	Отображение в браузере						
<pre><table border=1> <col><col style="background-color: yellow"> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<table border="1"> <tr><td>англ</td><td>рус</td></tr> <tr><td>red</td><td>красный</td></tr> <tr><td>green</td><td>зелёный</td></tr> </table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						

Парный тег **colgroup** внутри тела **table** позволяет логически сгруппировать вместе несколько колонок (тегов **col**), например, для установки им общих стилевых свойств.

Парные теги **thead**, **tbody**, **tfoot** внутри тела **table** позволяют сгруппировать строки таблицы (теги **tr**), например, для установки им общих стилевых свойств:

- тег **thead** позволяет выделить строки шапки таблицы; он может встретиться в таблице только один раз; если он есть, до него не должно быть других строк или групп строк;
- тег **tbody** позволяет выделить строки тела таблицы; таких тегов внутри одной таблицы может быть несколько;
- тег **tfoot** позволяет выделить строки подвала таблицы; он может встретиться в таблице только один раз; строки, входящие в него, будут отображены внизу таблицы, даже если после **tfoot** есть другие строки или группы строк.

(4.75) Управление слиянием рамок таблицы

Стилевое свойство **border-collapse** тега **table** — вид слияния рамок таблицы.

Устанавливает способ, каким сливаются рамки соседних ячеек таблицы.

Возможные значения:

border-collapse: collapse — рамки соседних ячеек сливаются вместе.

border-collapse: separate — рамки соседних ячеек отображаются отдельно.

HTML-код	Отображение в браузере						
<pre><table border=1> <caption>Учим английский язык вместе</caption> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<p>Учим английский язык вместе</p> <table border="1"><thead><tr><th>англ</th><th>рус</th></tr></thead><tbody><tr><td>red</td><td>красный</td></tr><tr><td>green</td><td>зелёный</td></tr></tbody></table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						
<pre><table border=1 style="border-collapse: collapse"> <caption>Учим английский язык вместе</caption> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	<p>Учим английский язык вместе</p> <table border="1"><thead><tr><th>англ</th><th>рус</th></tr></thead><tbody><tr><td>red</td><td>красный</td></tr><tr><td>green</td><td>зелёный</td></tr></tbody></table>	англ	рус	red	красный	green	зелёный
англ	рус						
red	красный						
green	зелёный						

(4.78) Ширина и высота таблицы и ячеек, строк, столбцов таблицы

Стилевые свойства **width**, **min-width**, **max-width** тега **table** — управляют шириной таблицы.

Стилевые свойства **height**, **min-height**, **max-height** тега **table** — управляют высотой таблицы.

Стилевые свойства **width**, **min-width**, **max-width** тега **col** — управляют шириной столбца таблицы, а то есть шириной всех ячеек в столбце.

Стилевые свойства **height**, **min-height**, **max-height** тега **tr** — управляют высотой строки таблицы, а то есть высотой всех ячеек в столбце.

Стилевые свойства **width**, **min-width**, **max-width** тегов **td**, **th** — управляют шириной ячейки таблицы, а то есть могут повлиять на ширину всего столбца.

Стилевые свойства **height**, **min-height**, **max-height** тегов **td**, **th** — управляют высотой ячейки таблицы, а то есть могут повлиять на высоту всей строки.

В любом случае:

- *ширина и высота таблицы будут такими, чтобы вместилось всё её содержимое;*
- *ширина столбца таблицы будет такой, чтобы вместилось содержимое всех ячеек столбца;*
- *высота строки таблицы будет такой, чтобы вместилось содержимое всех ячеек строки.*

Стилевое свойство **table-layout** тега **table** — управляет алгоритмом подбора ширины столбцов.

Возможные значения:

table-layout: auto — ширина столбцов подбирается автоматически по всему содержимому таблицы.

table-layout: fixed — ширина столбцов подбирается автоматически по элементам **col** (если они есть) или по содержимому ячеек только первой строки таблицы.

(4.80) Объединение ячеек таблицы

Атрибут **colspan** тегов **td, th** — объединение ячеек таблицы по горизонтали.

Атрибут **rowspan** тегов **td, th** — объединение ячеек таблицы по вертикали.

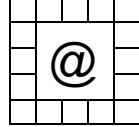
В случае указания **colspan=N**, данная ячейка занимает пространство сразу *N* ячеек по горизонтали. В данной строке таблицы не нужно указывать те элементы ячеек (**td, th**), которые «поглощены» данной ячейкой.

HTML-код		Отображение в браузере			
браузер	Netscape	Firefox	Opera	MSIE	
возраст	старый	новенький		так себе	
версия	9.0	23.0	15	11	

В случае указания **rowspan=N**, данная ячейка занимает пространство сразу *N* ячеек по вертикали. В последующих строках таблицы не нужно указывать те элементы ячеек (**td, th**), которые «поглощены» данной ячейкой.

HTML-код		Отображение в браузере		
		валюта	язык	
Испания		евро	испанский	
			итальянский	
Швеция		крона	шведский	

Можно объединить несколько ячеек по вертикали и горизонтали одновременно.

HTML-код	Отображение в браузере
<pre><table style="border-collapse: collapse" border=1> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td> <td colspan=3 rowspan=3 style="font-size: 24px">@</td> <td></td> </tr> <tr> <td></td><td></td> </tr> <tr> <td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table></pre>	

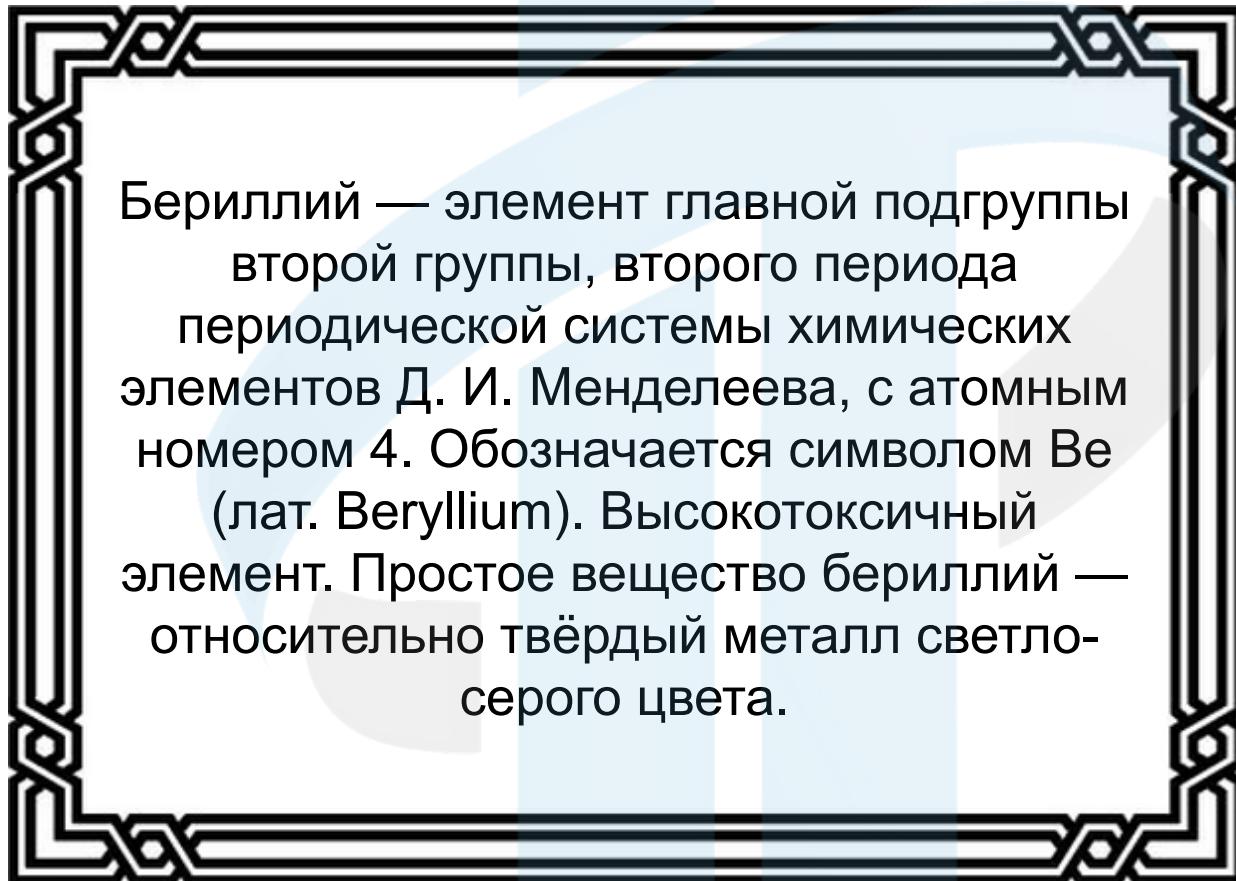
Домашнее задание №4

Создать проект DIV_FRAME. Добавить в проект из библиотеки картинку ramka.png.

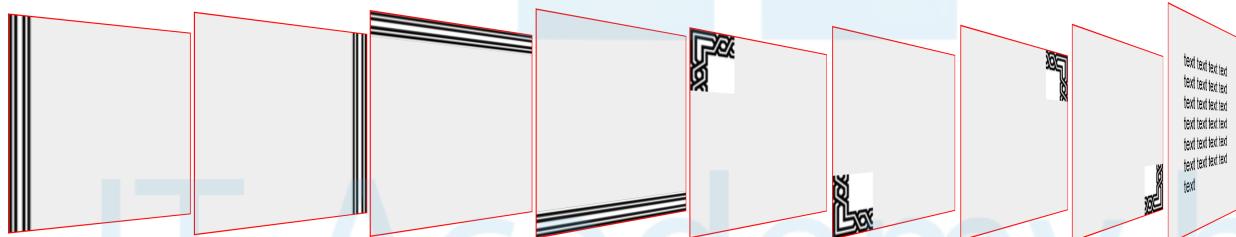
Разрезать ramka.png и сверстать резиновый фон, какой приведён ниже.

Текст можно использовать любой, но без жёстких переносов строк, чтобы мог растягиваться на произвольную ширину.

Размеры рамки должны зависеть от количества текста в ней и от размеров окна браузера (т.е. рамка по ширине — на всю ширину окна браузера, и имеет такую высоту, чтобы вместить содержащийся в ней текст).



Подсказка — план действий:



Тема 5. Обтекание. Вёрстка макетов. Селекторы

(5.10) Обтекание

Стилевое свойство **float** устанавливает для элемента режим **плавающий**.

Это стилевое свойство указывает, что элемент является плавающим и прижимается к левой или правой границе своего контейнера.

Элементы, идущие в HTML-коде после плавающего элемента, обтекают плавающий элемент.

Может применяться практически к любому элементу.

Возможные значения:

float: left — элемент плавающий, прижат влево.

float: right — элемент плавающий, прижат вправо.

float: none — элемент обычный, не плавающий.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>
<pre><div style="border: solid red 1px"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>
<pre><div style="border: solid red 1px"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>

Если после плавающего элемента в HTML-коде следует ещё один плавающий элемент, второй плавающий элемент обтекает первый плавающий элемент.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; text-align: justify"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>
<pre><div style="border: solid red 1px; text-align: justify"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; text-align: justify"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p> 

Если несколько обтекаемых элементов расположены подряд, причём второй по высоте больше чем первый, под первым образуется область, в которую обтекающий текст не попадает.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; text-align: justify"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>

Если несколько обтекаемых элементов расположены подряд и в доступную им ширину не помещаются, невместившиеся переносятся ниже, по возможности продолжая обтекать более ранние элементы.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства нераспустившегося цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы.</p>

Стилевое свойство **clear** отменяет для элемента режим обтекания плавающих элементов.

Это стилевое свойство запрещает данному элементу обтекать плавающие элементы, с одной или двух сторон.

Может применяться практически к любому элементу.

Возможные значения:

clear: left — этот элемент не может обтекать плавающие элементы, расположенные слева.

clear: right — этот элемент не может обтекать плавающие элементы, расположенные справа.

clear: both — этот элемент не может обтекать плавающие элементы вообще.

clear: none — этот элемент может обтекать плавающие элементы с любой стороны.

HTML-код	Отображение в браузере
<pre> <p>Это — дельфиниум.</p> <p style="clear: none"> Есть версия, что имя растения происходит от названия греческого города Дельфы.</p></pre>	 Это — дельфиниум. Есть версия, что имя растения происходит от названия греческого города Дельфы.
<pre> <p>Это — дельфиниум.</p> <p style="clear: left"> Есть версия, что имя растения происходит от названия греческого города Дельфы.</p></pre>	 Это — дельфиниум. Есть версия, что имя растения происходит от названия греческого города Дельфы.
<pre> <p>Это — дельфиниум.</p> <p style="clear: right"> Есть версия, что имя растения происходит от названия греческого города Дельфы.</p></pre>	 Это — дельфиниум. Есть версия, что имя растения происходит от названия греческого города Дельфы.
<pre> <p>Это — дельфиниум.</p> <p style="clear: both"> Есть версия, что имя растения происходит от названия греческого города Дельфы.</p></pre>	 Это — дельфиниум. Есть версия, что имя растения происходит от названия греческого города Дельфы.

Если у элемента отменено обтекание, то он не обтекает плавающие элементы, на каком бы уровне вложенности ни находились плавающие элементы.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> </div> <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p> <p style="clear: left">По другой версии, имя растения происходит от названия греческого города Дельфы.</p> </div> </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p> <p>По другой версии, имя растения происходит от названия греческого города Дельфы.</p>

Если у элемента отменено обтекание, то он не обтекает другие элементы, но это не мешает другим элементам обтекать его, если он плавающий.

Обтекаемым может быть любой элемент (*тег*). Для обтекаемых блочных элементов важно явно указывать ширину.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> <div style="width: 100px; background-color: yellow; float: left; clear: both; margin: 10px 20px 10px 10px"> покупайте
дельфиниум
только у нас! </div> <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. По другой версии, имя растения происходит от названия греческого города Дельфы. <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. </div></pre>	 <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p> <p>По другой версии, имя растения происходит от названия греческого города Дельфы. Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p>  <p>покупайте дельфиниум только у нас!</p>

Размеры обтекающих элементов включают в себя площадь, закрытую обтекаемыми элементами.

HTML-код	Отображение в браузере
<pre> <div style="border: solid blue 1px"> Фрэнк Синатра
1915-1998 </div></pre>	 <p>Фрэнк Синатра 1915-1998</p>

Соответственно, и фон, и рамки обтекающих элементов распространяются на площадь, закрытую обтекаемыми элементами.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> <h3 style="background-color: yellow; border-bottom: solid black 1px"> Дельфениум</h3> Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. </div></pre>	 <p>Дельфениум</p> <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p>

Если установить для обтекающего элемента стилевое свойство **overflow: hidden**, можно скрыть те части фона и рамок обтекающих элементов, которые закрыты обтекаемыми элементами.

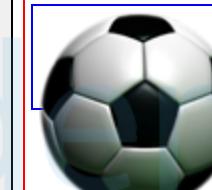
HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> <h3 style="background-color: yellow; border-bottom: solid black 1px; overflow: hidden"> Дельфениум</h3> Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина. </div></pre>	 <p>Дельфениум</p> <p>Своё научное латинское название растение получило, возможно, из-за сходства цветка с формой головы и тела дельфина.</p>

Размеры контейнера не включают в себя площадь, закрытую дочерними плавающими элементами.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; padding: 3px"> <div style="border: solid blue 1px"> Мяч футбольный.
 Покупай, не проиграешь! </div> </div></pre>	 <p>Мяч футбольный. Покупай, не проиграешь!</p>

Если необходимо включить в размеры контейнера площадь, закрытую дочерними плавающими элементами, можно пойти 3 путями:

- в контейнер включить пустой div с отменой обтекания (так называемый *clearfix*)
- установить контейнеру стилевое свойство *overflow: hidden*
- сам контейнер сделать плавающим.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; padding: 3px"> <div style="border: solid blue 1px"> Мяч футбольный.
 Покупай, не проиграешь! </div> </div></pre>	 <p>Мяч футбольный. Покупай, не проиграешь!</p>
<pre><div style="border: solid red 1px; padding: 3px; overflow: hidden"> <div style="border: solid blue 1px"> Мяч футбольный.
 Покупай, не проиграешь! </div> </div></pre>	 <p>Мяч футбольный. Покупай, не проиграешь!</p>
<pre><div style="border: solid red 1px; padding: 3px; float: left; width: 246px"> <div style="border: solid blue 1px"> Мяч футбольный.
 Покупай, не проиграешь! </div> </div></pre>	 <p>Мяч футбольный. Покупай, не проиграешь!</p>

Внешние поля (margin) и внутренние отступы (padding) обтекающих элементов нахлѣстываются на плавающие элементы, насколько возможно.

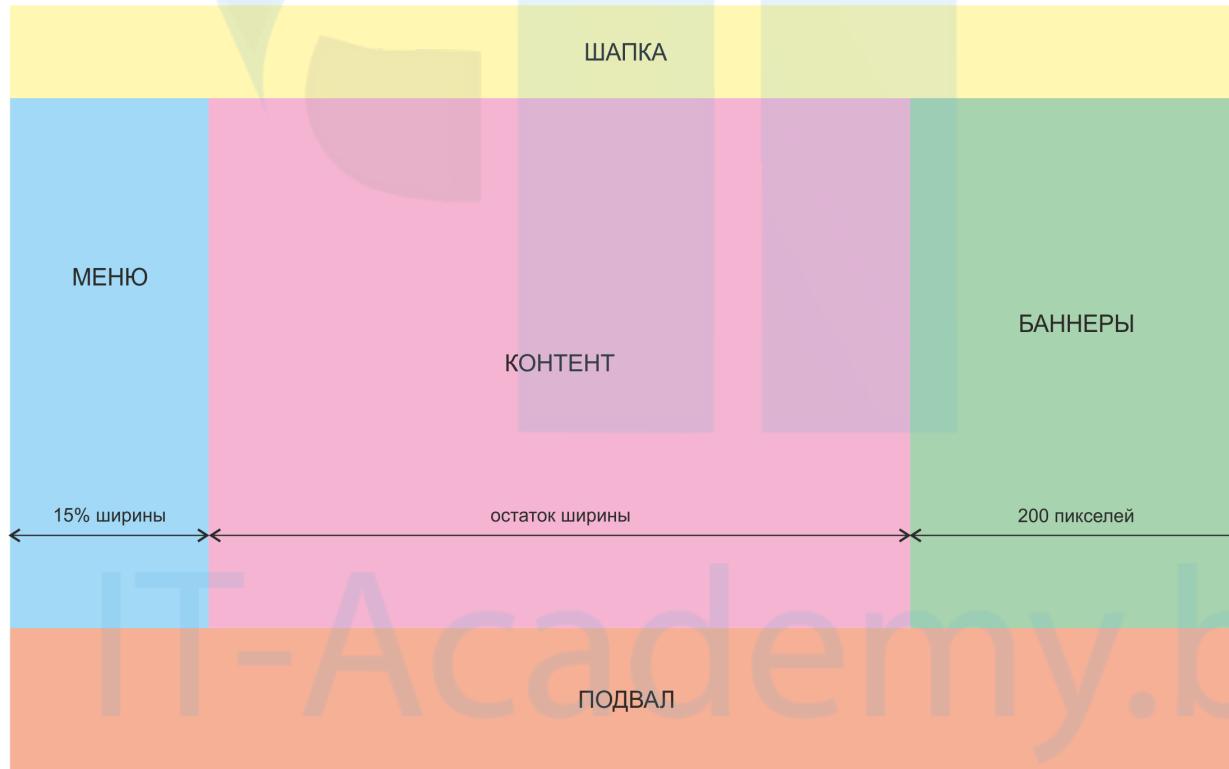
HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; padding: 3px"> <div style="border: solid blue 1px; margin-left: 170px"> Фрэнк Синатра
1915-1998
Девять раз становился лауреатом премии &laquo;Грэмми&raquo;. Славился стилем исполнения песен и тембром голоса. </div> </div></pre>	 <p>Фрэнк Синатра 1915-1998 Девять раз становился лауреатом премии «Грэмми». Славился стилем исполнения песен и тембром голоса.</p>
<pre><div style="border: solid red 1px; padding: 3px"> <div style="border: solid blue 1px; margin-left: 80px"> Фрэнк Синатра
1915-1998
Девять раз становился лауреатом премии &laquo;Грэмми&raquo;. Славился стилем исполнения песен и &laquo;бархатным&raquo; тембром голоса. </div> </div></pre>	 <p>Фрэнк Синатра 1915-1998 Девять раз становился лауреатом премии «Грэмми». Славился стилем исполнения песен и «бархатным» тембром голоса.</p>
<pre><div style="border: solid red 1px; padding: 3px"> <div style="border: solid blue 1px; padding-left: 80px"> Фрэнк Синатра
1915-1998
Девять раз становился лауреатом премии &laquo;Грэмми&raquo;. Славился стилем исполнения песен и &laquo;бархатным&raquo; тембром голоса. </div> </div></pre>	 <p>Фрэнк Синатра 1915-1998 Девять раз становился лауреатом премии «Грэмми». Славился стилем исполнения песен и «бархатным» тембром голоса.</p>

(5.30) Табличная и блочная вёрстка

Макет сайта:



Разбивка макета на функциональные блоки с примечаниями дизайнера:



Табличная вёрстка:

ЧЕРЕПАШКИ - ЛУЧШИЙ САЙТ ИНТЕРНЕТА!		
<p>Главная О нас Виды черепах Уход Кормление Дрессировка Обратная связь</p>	<p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.</p> <p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.</p> <p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов.</p>	<p>продадим лучший корм! ... ухаживаем круче всех! ... выслушаем жалобы! ... наши друзья</p>
<p><td colspan=3 style="background-color: brown"> сайт создан при поддержке.... хостинг для сайта предоставлен... сайт разработали... материалы для сайта предоставили...</p>		

Блочная вёрстка:

ЧЕРЕПАШКИ - ЛУЧШИЙ САЙТ ИНТЕРНЕТА!		
<p>Главная О нас Виды черепах Уход Кормление Дрессировка Обратная связь</p>	<p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.</p> <p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.</p> <p>Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов.</p>	<p>продадим лучший корм! ... ухаживаем круче всех! ... выслушаем жалобы! ... наши друзья</p>
<p><div style='clear: both; background-color: brown'> сайт создан при поддержке.... хостинг для сайта предоставлен... сайт разработали... материалы для сайта предоставили...</p>		

Важно: какие width установлены у div МЕНЮ (15%) и div РЕКЛАМА (200px), такие (или чуть больше) margin должны быть у div КОНТЕНТ (слева 15%, справа 200px).
Важно: из трёх центральных div в HTML-коде сначала должны быть записаны div МЕНЮ и div РЕКЛАМА (в любом порядке), и только после них — div КОНТЕНТ. Всегда в HTML-коде сначала идёт обтекаемое содержимое, а затем обтекающее, иначе обтекающее окажется не там где нужно.

(5.40) Столбцы одинаковой высоты в блочной вёрстке

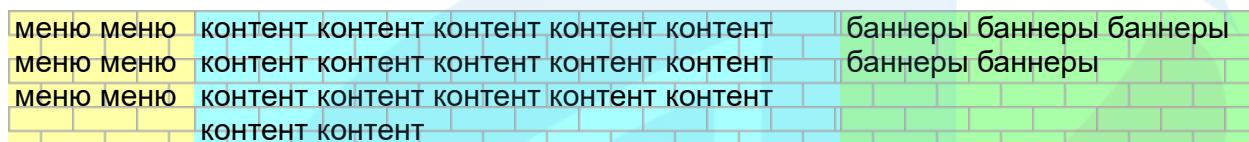
При блочной вёрстке не существует прямого способа добиться одинаковой высоты нескольких столбцов (т.е. элементов div). Каждый элемент div имеет высоту, определяемую его содержимым, и не зависящую от высоты других элементов div.

Однако, часто можно сверстать макет так, чтобы визуально столбцы **казались** имеющими одинаковую высоту.

Подход №1. Подготовка специальных фоновых изображений для контейнера

Можно добиться визуально одинаковых столбцов, используя специально подготовленные фоновые изображения, применяемые к контейнеру, в котором располагаются все столбцы. Т.е. элементы `div` имеют различную высоту, но не имеют фона; контейнер же как раз имеет такой фон, который совпадает по размеру и поведению с шириной столбцов, поэтому **кажется**, что столбец имеет ту же высоту что и контейнер.

Предположим, что на сайте про черепашек нужно сверстать следующий фон для столбцов (для простоты опущены блоки ШАПКА и ПОДВАЛ):



1. Верстаем столбцы без фона. Временно добавляем рамочки, чтобы видеть границы столбцов.

2. Добавляем контейнер для всех столбцов (розовая рамка).

Контейнер должен охватывать все колонки. Однако, элемент div охватывает всё своё содержимое, но игнорирует высоту float-элементов. Если блок МЕНЮ или блок БАННЕРЫ окажется длиннее блока КОНТЕНТ, контейнер не сработает так как нам нужно:

HTML-код		
Отображение в браузере		
меню меню меню меню меню меню	контент контент контент контент контент контент контент контент контент контент	баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры

Чтобы решить эту проблему, в контейнер можно включить ещё один (можно пустой) элемент div, имеющий стилевое свойство clear: both, т.е. гарантированно не обтекающий ни один float-элемент; он расположится ниже любых float-элементов, и он будет корректно охвачен контейнером:

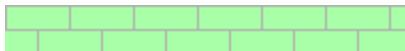
HTML-код		
Отображение в браузере		
меню меню меню меню меню меню (пустой)	контент контент контент контент контент контент контент контент контент контент	баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры

Убираем содержимое и рамку этого пустого div — раз он не имеет содержимого, он будет иметь нулевую высоту и не будет виден:

HTML-код		
Отображение в браузере		
меню меню меню меню меню меню	контент контент контент контент контент контент контент контент контент контент	баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры баннеры

Теперь контейнер всегда охватывает все три колонки, какую бы высоту они не имели.

3. Опишем фон для правой колонки. Колонка имеет фиксированную ширину — 200px, поэтому, подготовим нужное фоновое изображение ширины 200px и минимальной высоты (wall_g200.png):



Задаём это изображение в качестве фона **контейнера** справа:

HTML-код
<pre><div style="border: solid magenta 1px; background: url('images/wall_g200.png') right top repeat-y"> <div style="border: solid blue 1px; width: 15%; float: left">меню меню меню меню меню</div> <div style="border: solid green 1px; width: 200px; float: right">баннеры баннеры баннеры баннеры</div> <div style="border: solid red 1px; margin-left: 17%; margin-right: 210px">контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент <div style="clear: both"></div> </div></pre>
Отображение в браузере

Рамку вокруг блока БАННЕРЫ можно убрать, чтобы лучше наблюдать эффект:

HTML-код
<pre><div style="border: solid magenta 1px; background: url('images/wall_g200.png') right top repeat-y"> <div style="border: solid blue 1px; width: 15%; float: left">меню меню меню меню меню</div> <div style="width: 200px; float: right">баннеры баннеры баннеры баннеры</div> <div style="border: solid red 1px; margin-left: 17%; margin-right: 210px">контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент <div style="clear: both"></div> </div></pre>
Отображение в браузере

Поскольку мы задали фоновое изображение **контейнера**, оно повторяется на всю высоту контейнера; поскольку оно имеет ту же ширину и то же положение что и правая колонка — создаётся впечатление, что колонка БАННЕРЫ имеет ту же высоту что и колонка КОНТЕНТ, а зелёные кирпичи — это фон именно колонки БАННЕРЫ.

4. Опишем фон для левой и средней колонки (их можно описать отдельно, но вместе будет короче и проще). Левая колонка имеет резиновую ширину — 15%, поэтому подготовить для неё фоновое изображение так же как для правой колонки не получится.

Мы воспользуемся поведением стилевого свойства `background-position` в случае, когда положение фонового изображения задаётся в процентах.

Подготовим широкое (с расчётом на очень широкие экраны) фоновое изображение, 15% ширины которого составляют жёлтые кирпичи, и 85% — голубые (`wall_y15b85.png`):

(изображение wall_y15b85.png показано в уменьшенном виде, его ширина — 1882px)

Если задать это фоновое изображение контейнеру вместе с **background-position: 15% top**, то граница между жёлтыми и голубыми кирпичами (расположенная на **15% ширины изображения**) попадёт именно на **15% ширины контейнера**, то есть, на границу между блоком МЕНЮ и блоком КОНТЕНТ.

Обратите внимание, что ширина правой колонки — 200px — в данном случае роли не играет, т.е. неважно, что блок КОНТЕНТ имеет ширину не 85%, а меньше — важно лишь, что граница между блоком МЕНЮ и блоком КОНТЕНТ расположена на 15% ширины **всего контейнера**.

Фон для правой колонки временно уберём, чтобы лучше наблюдать происходящее. Итак:

Ширина фонового изображения (1882px) значительно больше, чем ширина контейнера (в данном случае менее 700px); фоновое изображение, выходящее за пределы контейнера, не отображается.

Рамки вокруг блоков МЕНЮ и КОНТЕНТ можно убрать, чтобы лучше наблюдать эффект:

Граница между жёлтыми и голубыми кирпичами расположилась там где мы хотим. Также, фон занимает всю высоту контейнера, поэтому создаётся впечатление, что колонка МЕНЮ имеет ту же высоту что и колонка КОНТЕНТ, а жёлтые кирпичи — это фон именно колонки МЕНЮ.

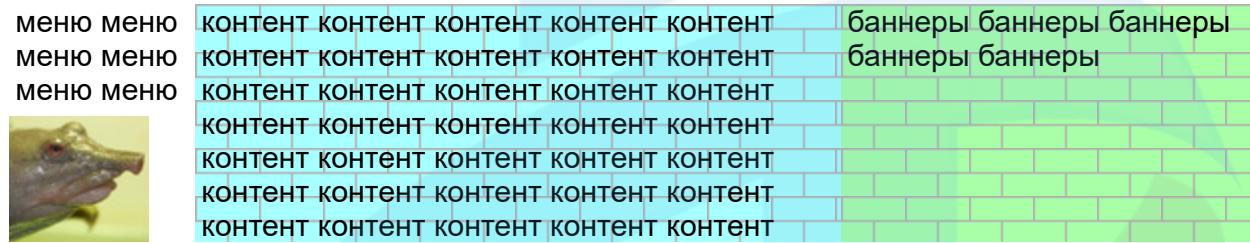
Заданный для блоков МЕНЮ и КОНТЕНТ фон заполнил также и блок БАННЕРЫ, однако, мы можем поверх этого фона расположить уже ранее проработанный фон блока БАННЕРЫ. Так как для описания фона блока БАННЕРЫ нам нужно для контейнера задать стилевое свойство background, а оно уже «занято» фоном для блоков МЕНЮ и КОНТЕНТ, мы должны описать ещё один контейнер (современные браузеры позволяют для одного HTML-элемента задавать сразу несколько фонов). Второй контейнер должен быть внутри первого контейнера (а не наоборот), т.к. нам нужно чтобы фоновое изображение второго контейнера перекрыло фоновое изображение первого контейнера:

Всё готово! Рамки обоих контейнеров можно убрать. Содержимое блоков может иметь любую высоту.

Кирпичный фон визуально не имеет начала и конца. Если в качестве фона требуется более сложное изображение, оно при способе «жёлтые и голубые кирпичи» прижмётся именно к резиновой границе между колонками, поэтому этот способ не подойдёт в случае, если фон с резиновой правой границей необходимо левой гранью прижать к левой границе колонки, либо если фон с резиновой левой границей необходимо правой гранью прижать к правой границе колонки.

Более гибкий подход подразумевает раздельную установку фона с двух сторон от резиновой границы между колонками. Суть метода та же — для резиновой границы используется специально подготовленное фоновое изображение, имеющее переход между двумя видами фона на нужном проценте ширины, но сейчас слева либо справа от границы фон будет прозрачным.

Предположим, что на сайте про черепашек нужно сверстать следующий фон для столбцов (для простоты опущены блоки ШАПКА и ПОДВАЛ):



Нужно, чтобы левая колонка в фоне имела прижатое к левому нижнему углу изображение черепахи, остальная площадь колонки не должна иметь фона.

1. Опишем фон для левой колонки.

Позиционирование изображения черепахи проблем не вызывает — её нужно просто расположить в левом нижнем углу контейнера:

HTML-код		
<div style="border: solid magenta 1px; background: url('images/turtle.jpg') left bottom no-repeat"> <div style="border: solid blue 1px; width: 15%; float: left">меню меню меню меню меню</div> <div style="border: solid green 1px; width: 200px; float: right">баннеры баннеры баннеры баннеры</div> <div style="border: solid red 1px; margin-left: 17%; margin-right: 210px">контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент</div> <div style="clear: both"></div> </div>		
Отображение в браузере		

2. Опишем фон для средней колонки.

Подготовим широкое (с расчётом на очень широкие экраны) фоновое изображение, 15% ширины которого составляют прозрачные пиксели, и 85% — голубые кирпичи (wall_t15b85.png):

(изображение wall_t15b85.png показано в уменьшенном виде, его ширина — 1882px)

Если задать это фоновое изображение контейнеру вместе с **background-position: 15% top**, то блок КОНТЕНТ получит в качестве фона голубые кирпичи, а блок МЕНЮ — прозрачные пиксели, через которые будет видна уже расположенная там черепаха.

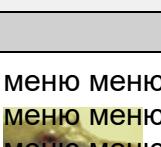
Поскольку один контейнер уже имеется (его фон — черепаха), нужно описать ещё один контейнер. Пусть необходимо чтобы при малой ширине окна браузера голубые кирпичи «накрывали» черепаху, тогда второй контейнер должен быть вложен в первый, а не наоборот:

Добавим уже известным способом зелёные кирпичи (ещё один контейнер) и уберём рамки:

HTML-код

```
<div style="background: url('images/turtle.jpg') left bottom no-repeat">
  <div style="background: url('images/wall_t15b85.png') 15% top repeat-y">
    <div style="background: url('images/wall_g200.png') right top repeat-y">
      <div style="width: 15%; float: left">меню меню меню меню меню меню</div>
      <div style="width: 200px; float: right">баннеры баннеры баннеры баннеры баннеры</div>
      <div style="margin-left: 17%; margin-right: 210px">контент контент контент контент контент контент контент
      контент контент контент контент контент контент контент контент контент контент контент контент
      контент</div>
      <div style="clear: both"></div>
    </div></div></div>
```

Отображение в браузере

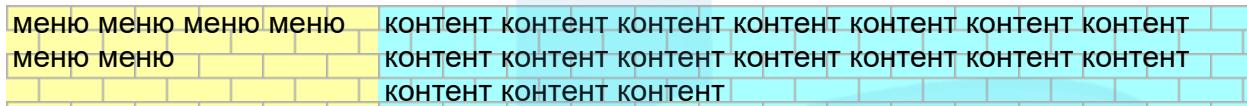
меню меню меню меню	контент контент контент контент	контент контент контент контент	баннеры баннеры баннеры баннеры
	контент контент контент контент	контент контент контент контент	баннеры баннеры баннеры баннеры
	контент контент контент контент	контент контент контент контент	баннеры баннеры баннеры баннеры

Описанные способы компоновки фона можно комбинировать в различных сочетаниях, добиваясь желаемого результата.

Подход №2. Перекрытие фона контейнера фоном столбца

Можно добиться визуально одинаковых столбцов, применяя фоновое изображение одного из столбцов не к самому столбцу, а к контейнеру, содержащему столбцы; тогда фоновое изображение другого столбца визуально перекрывает часть первого. При этом важно обеспечить, чтобы второй столбец по высоте был не меньше первого.

Предположим, что на сайте про черепашек нужно сверстать следующий фон для столбцов (для простоты опущены блоки ШАПКА и ПОДВАЛ, и пока только два столбца):



1. Верстаем столбцы без фона. Временно добавляем рамочки, чтобы видеть границы столбцов.

2. Добавляем контейнер. Помним, что для того чтобы контейнер охватывал и float-элементы тоже, следует добавить в контейнер ещё один div с clear:both.

HTML-код
<div style="border: solid magenta 1px"> <div style="border: solid blue 1px; width: 30%; float: left">меню меню меню меню меню меню меню меню</div> <div style="border: solid red 1px; margin-left: 32%">контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент</div> <div style="clear:both"></div> </div>

3. Применяем к контейнеру то фоновое изображение, которое должно быть у обтекаемого (плавающего) элемента, т.е. того у которого установлено стилевое свойство float.

HTML-код
<div style="border: solid magenta 1px; background: url('images/brick_yellow.png')"> <div style="border: solid blue 1px; width: 30%; float: left">меню меню меню меню меню меню меню меню</div> <div style="border: solid red 1px; margin-left: 32%">контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент контент</div> <div style="clear:both"></div> </div>

4. Применяем к обтекающему элементу его фоновое изображение.

Фон контейнера перекрываетя голубыми кирпичами справа, но остаётся виден слева, причём даже ниже самого div с меню, и визуально кажется, что слева - фон столбца.

Но у этой реализации будут проблемы, если меню окажется длиннее контента:

5. Если для какого-либо элемента отменено обтекание (clear), то он не будет обтекать обтекаемые элементы, даже если они принадлежат другим контейнерам. То есть, **если внутри обтекающего текста есть элемент со стилем clear, то этот элемент опустится ниже до такого места, где уже (с указанной стороны) не будет обтекаемых элементов, а следовательно, высота обтекающего текста дойдёт минимум до конца обтекаемого (обтекаемых) блоков.**

В данном случае пустой div с синей рамкой опустился вниз до конца обтекаемого им блока меню.

6. Т.к. теперь высота контента не будет меньше высоты меню, то div с clear:both, расширяющий контейнер, уже становится ненужным — float-содержимое никак не может «вылезти» за него.

7. Убираем отладочные рамки.

HTML-код
<div style="background: url('images/brick_yellow.png')"> <div style="width: 30%; float: left">меню меню меню</div> <div style="margin-left: 32%; background: url('images/brick_blue.png')"> контент контент контент контент контент контент контент контент контент контент <div style="clear: both"></div> </div> </div>

Данная реализация сработает при любом соотношении высот меню и контента, при любой стороне расположения обтекаемого контента (float:left, float:right), и при любом способе установки размера блока меню — px, em, %, mm...

Данный подход применим только для случая, если колонок ровно две.

Если колонок больше, в общем случае такой подход не поможет.

Возможности обоих подходов могут быть скомбинированы при необходимости.

Ограничения обоих подходов:

- везде, где используется наложение фона одного элемента на другой, становится невозможным использовать полупрозрачные фоны или фоны с прозрачными областями (т.к. через прозрачные и полупрозрачные области фона первого плана будет просматриваться фон второго плана);
 - везде, где используется наложение фона одного элемента на другой, становятся неприменимы некоторые способы выравнивания фона второго плана, например, для блока меню background-repeat: repeat-y; background-position: right расположит фон не у правого края колонки, а у правого края контейнера.

(5.50) Выравнивание блочного содержимого

Обычно, блочный элемент не может быть выровнен внутри его контейнера стилевым свойством контейнера `text-align`, даже если блочному элементу явно задана ширина, меньшая ширины контейнера.

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px; text-align: center"> <div style="width: 140px; border: solid blue 1px"> Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё. </div> </div></pre>	<p>Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё.</p>

Блочный элемент, более узкий чем его контейнер, можно выровнять внутри его контейнера по горизонтали несколькими способами:

- установкой элементу стилевых свойств `margin-left` и `margin-right` в значение `auto` (центрирование);
- установкой элементу стилевого свойства `display` в значение `inline-block`, при этом на него начинает действовать стилевое свойство контейнера `text-align` (формально, элемент уже при этом не является блочным).

HTML-код	Отображение в браузере
<pre><div style="border: solid red 1px"> <div style="width: 140px; border: solid blue 1px; margin: 0 auto 0 auto"> Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё. </div> </div></pre>	<p>Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё.</p>
<pre><div style="border: solid red 1px; text-align: center"> <div style="width: 140px; border: solid blue 1px; display: inline-block"> Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё. </div> </div></pre>	<p>Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё.</p>
<pre><div style="border: solid red 1px; text-align: right"> <div style="width: 140px; border: solid blue 1px; display: inline-block; text-align: left"> Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё. </div> </div></pre>	<p>Красиво ты вошла в мою грешную жизнь, красиво ты ушла из неё.</p>

Обычно, таблица не может быть выровнена внутри её контейнера стилевым свойством контейнера `text-align`, даже если таблице явно задана ширина, меньшая ширины контейнера.

HTML-код	Отображение в браузере							
<pre><div style="border: solid red 1px; text-align: center"> <table border="1" style="border-collapse: collapse; width: 150px"> <tr><th></th><th>Netscape</th></tr> <tr><th>возраст</th><td>старый</td></tr> <tr><th>версия</th><td>9.0</td></tr> </table> </div></pre>	<table border="1"> <thead> <tr> <th></th><th>Netscape</th></tr> </thead> <tbody> <tr> <td>возраст</td><td>старый</td></tr> <tr> <td>версия</td><td>9.0</td></tr> </tbody> </table>		Netscape	возраст	старый	версия	9.0	
	Netscape							
возраст	старый							
версия	9.0							

Таблицу, более узкую чем её контейнер, можно выровнять внутри её контейнера по горизонтали несколькими способами:

- установкой таблице стилевых свойств `margin-left` и `margin-right` в значение `auto` (центрирование);
- установкой таблице стилевого свойства `display` в значение `inline-table`, при этом на неё начинает действовать стилевое свойство контейнера `text-align`;
- установкой таблице HTML-атрибута `align` равным значению `center` или `right` (соответственно центрирование и выравнивание к правому краю).

HTML-код	Отображение в браузере							
<pre><div style="border: solid red 1px"> <table border="1" style="border-collapse: collapse; margin: 0 auto 0 auto"> <tr><th></th><th>Netscape</th></tr> <tr><th>возраст</th><td>старый</td></tr> <tr><th>версия</th><td>9.0</td></tr> </table> </div></pre>	<table border="1"> <thead> <tr> <th></th><th>Netscape</th></tr> </thead> <tbody> <tr> <td>возраст</td><td>старый</td></tr> <tr> <td>версия</td><td>9.0</td></tr> </tbody> </table>		Netscape	возраст	старый	версия	9.0	
	Netscape							
возраст	старый							
версия	9.0							
<pre><div style="border: solid red 1px; text-align: center"> <table border="1" style="border-collapse: collapse; display: inline-table"> <tr><th></th><th>Netscape</th></tr> <tr><th>возраст</th><td>старый</td></tr> <tr><th>версия</th><td>9.0</td></tr> </table> </div></pre>	<table border="1"> <thead> <tr> <th></th><th>Netscape</th></tr> </thead> <tbody> <tr> <td>возраст</td><td>старый</td></tr> <tr> <td>версия</td><td>9.0</td></tr> </tbody> </table>		Netscape	возраст	старый	версия	9.0	
	Netscape							
возраст	старый							
версия	9.0							
<pre><div style="border: solid red 1px"> <table border="1" style="border-collapse: collapse" align="center"> <tr><th></th><th>Netscape</th></tr> <tr><th>возраст</th><td>старый</td></tr> <tr><th>версия</th><td>9.0</td></tr> </table> </div></pre>	<table border="1"> <thead> <tr> <th></th><th>Netscape</th></tr> </thead> <tbody> <tr> <td>возраст</td><td>старый</td></tr> <tr> <td>версия</td><td>9.0</td></tr> </tbody> </table>		Netscape	возраст	старый	версия	9.0	
	Netscape							
возраст	старый							
версия	9.0							

(5.70) Селектор элементов по классу

Стилевой селектор **.класс** — соответствует всем элементам, имеющим данный класс, т.е. тегам, у которых атрибут **class** равен **класс**.

HTML-код	Отображение в браузере
<pre><style> .main { color: red } .remark { font-style: italic } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p class="main">Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p class="main">Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p> <p class="remark">В естественных условиях в течение года население муравейника почти полностью обновляется.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p> <p>В естественных условиях в течение года население муравейника почти полностью обновляется.</p>

Для тега в атрибуте **class** может быть указано сразу несколько классов, через пробел.

HTML-код	Отображение в браузере
<pre><style> .main { color: red } .remark { font-style: italic } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p class="main main remark">Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p class="main remark">Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p> <p class="remark">В естественных условиях в течение года население муравейника почти полностью обновляется.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p> <p>В естественных условиях в течение года население муравейника почти полностью обновляется.</p>

(5.75) Селектор элемента по идентификатору

Стилевой селектор **#идентификатор** — соответствует **одному элементу**, имеющему данный идентификатор, т.е. тегу, у которого атрибут **id** равен **идентификатор**.

HTML-код	Отображение в браузере
<pre><style> #Queen { font-weight: bold } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p id="Queen">Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p>

Домашнее задание №5

Создать новый проект **МАКЕТ_TABLE**. Сверстать по макету табличной вёрсткой.

Создать новый проект **МАКЕТ_DIV**. Сверстать по макету блочной вёрсткой.

Макет — в библиотечном файле **football.png**.

Вёрстка должна быть резиновой — при изменении ширины окна браузера основная колонка должна растягиваться; ширина колонки с меню должна быть фиксированной.

Гиперссылки могут вести куда угодно. Тексты могут быть произвольными.

Количество пунктов меню должно быть достаточным для того, чтобы при широком окне браузера колонка с меню оказывалась больше по высоте чем основная колонка.

ПРИВЕТ УЧАСТИКАМ СОРЕВНОВАНИЙ!

Виды спорта:

- [конькобежный спорт](#)
- [биатлон](#)
- [футбол](#)
- [стрельба из лука](#)
- [прыжки в длину](#)
- [прыжки в высоту](#)
- [фигурное катание](#)
- [синхронное плавание](#)
- [шахматы](#)
- [шашки](#)

ФУТБОЛ



Футбол — командный вид спорта, в котором целью является забить мяч в ворота соперника ногами или другими частями тела (кроме рук) большее количество раз, чем команда соперника.

Есть 17 официальных правил игры, каждое из которых содержит список оговорок и руководящих принципов. Эти правила предназначены для применения на всех уровнях футбола, хотя есть некоторые изменения для таких групп, как юниоры, взрослые, женщины и люди с ограниченными физическими возможностями. Законы очень часто формулировались в общих чертах, которые позволяют упростить их применения в зависимости от характера игры.

ГЛАВНОЕ НЕ ПОБЕДА, А УЧАСТИЕ!

Тема 6. Иерархические селекторы. Каскадирование

(6.20) Селектор элементов по тегу

Стилевой селектор **тег** — соответствует всем элементам, имеющим данный тег.

HTML-код	Отображение в браузере
<pre><style> span { color: red } </style> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p></pre>	<p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p>

(6.30) Универсальный селектор

Стилевой селектор * — соответствует всем элементам.

HTML-код	Отображение в браузере
<pre><style> * { color: red } </style> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <div>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</div></pre>	<p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p>

(6.35) Селектор элементов по атрибутам

Стилевой селектор **[атрибут]** — соответствует всем элементам, имеющим данный атрибут, т.е. тегам, у которых данный атрибут установлен (в любое значение).

HTML-код	Отображение в браузере
<pre><style> [align] { color: red } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p align="center">Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p> <p align="right">Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз больше, чем большинство одиночных насекомых аналогичного размера.</p>

Стилевой селектор **[атрибут=значение]** — соответствует всем элементам, имеющим данный атрибут, равный данному значению.

HTML-код	Отображение в браузере
<pre><style> [align="center"] { color: red } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p align="center">Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p> <p align="right">Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p>

Стилевой селектор **[атрибут^=значение]** — соответствует всем элементам, у которых значение данного атрибута начинается с указанного значения.

HTML-код	Отображение в браузере
<pre><style> [href^="http://"] { font-weight: bold } </style> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p></pre>	<p>Королева может жить до 20 лет, рабочие особи <u>живут</u> от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p>

Стилевой селектор **[атрибут\$=значение]** — соответствует всем элементам, у которых значение данного атрибута заканчивается на указанное значение.

HTML-код	Отображение в браузере
<pre><style> [href\$=".gif"] { color: red } </style> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p></pre>	<p><u>Королева</u> может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p>

Стилевой селектор **[атрибут*=значение]** — соответствует всем элементам, у которых значение данного атрибута содержит указанное значение.

HTML-код	Отображение в браузере
<pre><style> [href*="queen"] { color: green } </style> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p></pre>	<p><u>Королева</u> может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p>

(6.40) Селектор элементов по нескольким критериям сразу

Любые виды селекторов можно объединять простым склеиванием, **без пробела**.

Например:

- стилевой селектор **div.out** соответствует тегам DIV, имеющим класс "out";
- стилевой селектор **div.out.main** соответствует тегам DIV, имеющим одновременно классы "out" и "main";
- стилевой селектор **a#main** соответствует тегу A, имеющему идентификатор "main";
- стилевой селектор **img[src\$='.jpeg']** соответствует тегам IMG, у которых атрибут SRC заканчивается на ".jpeg";
- стилевой селектор **a.back[href^='http://']** соответствует тегам A, имеющим класс "back", у которых атрибут HREF начинается на "http://".

(6.45) Иерархические селекторы

Иерархический стилевой селектор **селектор1 селектор2** — соответствует всем элементам, подходящим под **селектор2**, и в то же время являющимся **дочерними или внучатыми** к элементу, подходящему под **селектор1**.

HTML-код	Отображение в браузере
<pre><style> p span { color: red } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, <i>живут всего несколько недель</i>.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p>

Иерархический стилевой селектор **селектор1 > селектор2** — соответствует всем элементам, подходящим под **селектор2**, и в то же время являющимся **дочерними** к элементу, подходящему под **селектор1**.

HTML-код	Отображение в браузере
<pre><style> p > span { color: red } </style> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, <i>живут всего несколько недель</i>.</p></pre>	<p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p>

Иерархический стилевой селектор **селектор1 + селектор2** — соответствует всем элементам, подходящим под селектор2, которые в своём контейнере идут **сразу после** элемента, подходящего под селектор1.

HTML-код	Отображение в браузере
<pre><style> h3 + p { color: red } </style> <h3>МУРАВЬИ</h3> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p></pre>	<p>МУРАВЬИ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p>

Иерархический стилевой селектор **селектор1 ~ селектор2** — соответствует всем элементам, подходящим под селектор2, которые в своём контейнере идут **после** элемента, подходящего под селектор1.

HTML-код	Отображение в браузере
<pre><style> h3 ~ p { color: red } </style> <p>Здравствуйте, дорогие слушатели!</p> <p>Сегодня мы поговорим о муравьях.</p> <h3>МУРАВЬИ</h3> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p></pre>	<p>Здравствуйте, дорогие слушатели!</p> <p>Сегодня мы поговорим о муравьях.</p> <p>МУРАВЬИ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет, рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p>

(6.50) Селекторы псевдоклассов

Псевдоклассы — подразумеваемые стилевые классы, к которым элементы (теги) автоматически относятся на основании каких-либо признаков.

Стилевой селектор псевдокласса **:first-child** — соответствует тем элементам, которые в своём контейнере являются первыми.

Стилевой селектор псевдокласса **:last-child** — соответствует тем элементам, которые в своём контейнере являются последними.

HTML-код	Отображение в браузере
<pre><style> p:first-child { color: red } </style> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <div> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p> <p>Рабочие особи живут от 1 до 3 лет.</p> </div> <div> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p> <p>Рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p>
<pre><style> p:last-child { color: red } </style> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <div> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p> <p>Рабочие особи живут от 1 до 3 лет.</p> </div> <div> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p> <p>Рабочие особи живут от 1 до 3 лет.</p> <p>Самцы, однако, живут всего несколько недель.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем большинство одиночных насекомых аналогичного размера.</p>

Стилевой селектор псевдокласса **:first-of-type** — соответствует тем элементам, которые в своём контейнере является первыми **среди элементов того же типа**, т.е. того же тега.

Стилевой селектор псевдокласса **:last-of-type** — соответствует тем элементам, которые в своём контейнере являются последними **среди элементов того же типа**, т.е. того же тега.

Обычно комбинируются с селектором по тегу.

HTML-код	Отображение в браузере
<pre><style> p:first-of-type { color: red } </style> <div> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <h3>КОРОЛЕВА</h3> <p>Королева может жить до 20 лет.</p> <i>использованная литература...</i> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>КОРОЛЕВА</p> <p>Королева может жить до 20 лет.</p> <p>использованная литература...</p>
<pre><style> p:last-of-type { color: red } </style> <div> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <h3>КОРОЛЕВА</h3> <p>Королева может жить до 20 лет.</p> <i>использованная литература...</i> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>КОРОЛЕВА</p> <p>Королева может жить до 20 лет.</p> <p>использованная литература...</p>

IT-Academy.by

Стилевой селектор псевдокласса **:nth-child** соответствует элементам, которые в своём контейнере расположены под номерами с некоторыми интервалом.

Варианты использования:

:nth-child(odd) — соответствует элементам, которые в своём контейнере нечётные.

:nth-child(even) — соответствует элементам, которые в своём контейнере чётные.

:nth-child(*i*) — соответствует элементу, который в своём контейнере идёт под номером *i*.

:nth-child(*a*+*b*) — вместо *n* будут подставлены числа от 0 и выше; результат вычисления формулы даёт номера дочерних элементов в их контейнере; *a* и *b* должны быть целыми числами и могут быть отрицательными.

HTML-код	Отображение в браузере							
<pre><style> tr:nth-child(odd) { background-color: #B0B0B0 } </style> <table style="border-collapse: collapse"> <tr><td>первая строка</td></tr> <tr><td>вторая строка</td></tr> <tr><td>третья строка</td></tr> <tr><td>четвёртая строка</td></tr> <tr><td>пятая строка</td></tr> </table></pre>	<table border="1"><tr><td>первая строка</td></tr><tr><td>вторая строка</td></tr><tr><td>третья строка</td></tr><tr><td>четвёртая строка</td></tr><tr><td>пятая строка</td></tr></table>	первая строка	вторая строка	третья строка	четвёртая строка	пятая строка		
первая строка								
вторая строка								
третья строка								
четвёртая строка								
пятая строка								
<pre><style> tr:nth-child(3) { background-color: #B0B0B0 } </style> <table style="border-collapse: collapse"> <tr><td>первая строка</td></tr> <tr><td>вторая строка</td></tr> <tr><td>третья строка</td></tr> <tr><td>четвёртая строка</td></tr> </table></pre>	<table border="1"><tr><td>первая строка</td></tr><tr><td>вторая строка</td></tr><tr><td>третья строка</td></tr><tr><td>четвёртая строка</td></tr></table>	первая строка	вторая строка	третья строка	четвёртая строка			
первая строка								
вторая строка								
третья строка								
четвёртая строка								
<pre><style> tr:nth-child(3n+2) { background-color: #B0B0B0 } </style> <table style="border-collapse: collapse"> <tr><td>первая строка</td></tr> <tr><td>вторая строка</td></tr> <tr><td>третья строка</td></tr> <tr><td>четвёртая строка</td></tr> <tr><td>пятая строка</td></tr> <tr><td>шестая строка</td></tr> </table></pre>	<table border="1"><tr><td>первая строка</td></tr><tr><td>вторая строка</td></tr><tr><td>третья строка</td></tr><tr><td>четвёртая строка</td></tr><tr><td>пятая строка</td></tr><tr><td>шестая строка</td></tr></table>	первая строка	вторая строка	третья строка	четвёртая строка	пятая строка	шестая строка	
первая строка								
вторая строка								
третья строка								
четвёртая строка								
пятая строка								
шестая строка								
<pre><style> tr:nth-child(-n+3) { background-color: #B0B0B0 } </style> <table style="border-collapse: collapse"> <tr><td>первая строка</td></tr> <tr><td>вторая строка</td></tr> <tr><td>третья строка</td></tr> <tr><td>четвёртая строка</td></tr> <tr><td>пятая строка</td></tr> <tr><td>шестая строка</td></tr> <tr><td>седьмая строка</td></tr> </table></pre>	<table border="1"><tr><td>первая строка</td></tr><tr><td>вторая строка</td></tr><tr><td>третья строка</td></tr><tr><td>четвёртая строка</td></tr><tr><td>пятая строка</td></tr><tr><td>шестая строка</td></tr><tr><td>седьмая строка</td></tr></table>	первая строка	вторая строка	третья строка	четвёртая строка	пятая строка	шестая строка	седьмая строка
первая строка								
вторая строка								
третья строка								
четвёртая строка								
пятая строка								
шестая строка								
седьмая строка								

Стилевой селектор псевдокласса **:nth-of-type** соответствует элементам, которые в своём контейнере расположены под номерами с некоторыми интервалом **среди элементов того же типа**, т.е. того же тега. Обычно комбинируется с селектором по тегу.

Варианты использования:

:nth-of-type(odd) — соответствует элементам, которые в своём контейнере нечётные среди элементов того же типа (тега).

:nth-of-type(even) — соответствует элементам, которые в своём контейнере чётные среди элементов того же типа (тега).

:nth-of-type(*i*) — соответствует элементу, который в своём контейнере идёт под номером *i* среди элементов того же типа (тега).

:nth-of-type(*a*+*b*) — вместо *n* будут подставлены числа от 0 и выше; результат вычисления формулы даёт номера дочерних элементов в их контейнере среди элементов того же типа (тега);

a и *b* должны быть целыми числами и могут быть отрицательными.

HTML-код	Отображение в браузере
<pre><style> p { line-height: 100% } img:nth-of-type(odd) { float: right } img:nth-of-type(even) { float: left; margin-right: 10px } </style> <div> <p>Когда мы улыбаемся,</p> <p>мы реже ошибаемся.</p> <p>И чаще награждаемся</p> <p>подарками судьбы.</p> <p>Когда мы улыбаемся,</p> <p>мы жизнью наслаждаемся.</p> <p>И вмig освобождаемся</p> <p>от горестей любых.</p> <p>Улыбка-вещь бесплатная,</p> <p>простая и понятная,</p> <p>смешная и понятная,</p> <p>доступная для всех.</p> <p>Когда мы улыбаемся,</p> <p>Мы самоисцеляемся.</p> <p>И силой наполняемся,</p> <p>дарующей успех!</p> </div></pre>	<p>Когда мы улыбаемся, мы реже ошибаемся. И чаще награждаемся  подарками судьбы.</p> <p>Когда мы улыбаемся, мы жизнью наслаждаемся. </p> <p>И вмig освобождаемся от горестей любых.</p> <p>Улыбка-вещь бесплатная, простая и понятная, смешная и понятная, доступная для всех.</p> <p> Когда мы улыбаемся, Мы самоисцеляемся. И силой наполняемся, дарующей успех!</p>

Аналогично селекторам псевдоклассов **:nth-child** и **:nth-of-type**, есть селекторы псевдоклассов **:nth-last-child** и **:nth-last-of-type**, работающие полностью аналогично, только отсчёт элементов ведётся не от начала родителя, а от конца.

Стилевой селектор псевдокласса **:hover** — соответствует элементу, над которым сейчас находится курсор мыши.

Стилевой селектор псевдокласса **:active** — соответствует элементу, который сейчас активирован (например, ссылка, когда на ней нажата кнопка мыши).

Стилевой селектор псевдокласса **:focus** — соответствует элементу, который сейчас имеет фокус ввода (например, поле ввода текста, в котором сейчас находится текстовый курсор).

Стилевой селектор псевдокласса **:link** — соответствует ещё не посещённым гиперссылкам.

Стилевой селектор псевдокласса **:visited** — соответствует уже посещённым гиперссылкам.

Стилевой селектор псевдокласса **:target** — соответствует тем элементам, идентификатор (HTML-атрибут id) которых сейчас указан в закладке УРЛа страницы.

Стилевой селектор псевдокласса **:not(сел)** — соответствует элементам, **не подходящим** под указанный селектор **сел**.

Селектор **сел** не может быть любым, есть некоторые ограничения; например, в нём не допускается использование псевдоэлементов (о них ниже).

Обычно селектор **:not** комбинируется с селекторами других видов.

HTML-код	Отображение в браузере
<pre><style> div:not(.remark) { color: blue } </style> <div class="header1">Заголовок 1</div> <div class="header2">Заголовок 2</div> <div>текст текст текст</div> <div class="info">информация информация</div> <div class="remark">примечание</div> <div class="info">текст текст текст</div> <p>абзац абзац абзац</p></pre>	<p>Заголовок 1 Заголовок 2 текст текст текст информация информация примечание текст текст текст абзац абзац абзац</p>

(6.52) Селектор элементов по нескольким критериям сразу с использованием псевдоклассов

Любые виды селекторов, в том числе и селекторы псевдоклассов, можно объединять простым склеиванием, **без пробела**.

Например:

- стилевой селектор **a.NAV:visited:hover** соответствует тегам A, имеющим класс "NAV", ведущим на уже посещённую ссылку, если к ним сейчас поднесён курсор мыши;
- стилевой селектор **img[src\$='.jpg']:first-child** соответствует тегам IMG, отображающим картинку с расширением jpg, если картинка является первым элементом в своём контейнере;
- стилевой селектор **tr:not(:first-child):not(:last-child)** соответствует всем строкам таблиц, кроме первой и последней.

(6.55) Селекторы псевдоэлементов

Псевдоэлементы — подразумеваемые элементы, не являющиеся обычными HTML-элементами (тегами).

Стилевой селектор псевдоэлемента **::first-line** — соответствует первой строке элементов.

HTML-код	Отображение в браузере
<pre><style> p::first-line { color: red } </style> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <div> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем многие насекомые аналогичного размера.</p> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Муравьиные королевы живут в 100 раз дольше, чем многие насекомые аналогичного размера.</p>

Стилевой селектор псевдоэлемента **::first-letter** — соответствует первому символу элементов.

HTML-код	Отображение в браузере
<pre><style> p::first-letter { color: red } </style> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <div> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p> </div></pre>	<p>МУРАВЬИНАЯ СЕМЬЯ</p> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.</p> <p>Королева может жить до 20 лет.</p>

Стилевой селектор псевдоэлемента **::before** — соответствует (несуществующему) содержимому, расположенному перед содержимым парных элементов.
 Стилевой селектор псевдоэлемента **::after** — соответствует (несуществующему) содержимому, расположенному после содержимого парных элементов.
 Это несуществующее содержимое можно задать стилевым свойством **content**.

Стилевое свойство **content** — позволяет задать генерируемое содержимое.
 Применяется к элементам, содержимое которых изначально отсутствует в тексте веб-страницы. Обычно применяется вместе с селекторами псевдоэлементов **::before** и **::after**.

Некоторые возможные значения:

content: none — нет генерируемого содержимого.

content: 'текст' — задаёт *текст* в качестве генерируемого содержимого;

текст не может содержать HTML-тегов или HTML-сущностей (мнемоник).

content: url('ссылка') — задаёт изображение, расположенное по *ссылке*, в качестве генерируемого содержимого.

content: attr(атрибут) — задаёт значение HTML-атрибута тега в качестве генерируемого содержимого.

content: counter(счётчик) — задаёт значение счётика; CSS-счётик может быть сброшен в каких-либо тегах стилевым свойством **counter-reset: счётчик** и может быть увеличен в каких-либо тегах стилевым свойством **counter-increment: счётчик**.

content: значение1 значение2 значение3... — может быть скомбинировано любое количество значений вышеуказанных форматов.

HTML-код	Отображение в браузере
<pre> <style> b::before { content: '((('; color: red } b::after { content: '))'); color: red } p::before { content: url('images/smile.gif') } [data-bookref]::after { content: '[' attr(data-bookref) ']'; font-size: 67%; position: relative; top: -0.4em; } </style> <h3>МУРАВЬИНАЯ СЕМЬЯ</h3> <p>Муравьиные семьи могут существовать на протяжении долгого периода времени.
 Информация взята из надёжных источников.</p> </pre>	<h3>МУРАВЬИНАЯ (((СЕМЬЯ)))</h3> <p> Муравьиные семьи могут существовать на протяжении ((длого)) периода времени. Информация взята из надёжных источников<small>[БСС, стр. 25]</small>.</p>
<pre> <style> h3 { counter-reset: planet } p::before { counter-increment: planet; content: 'Планета ' counter(planet) ' ' } </style> <h3>Планеты до Земли</h3> <p>Меркурий</p><p>Венера</p> <h3>Планеты после Земли</h3> <p>Марс</p><p>Юпитер</p><p>Сатурн</p> </pre>	<h3>Планеты до Земли</h3> <p>Планета 1. Меркурий</p> <p>Планета 2. Венера</p> <h3>Планеты после Земли</h3> <p>Планета 1. Марс</p> <p>Планета 2. Юпитер</p> <p>Планета 3. Сатурн</p>

(6.57) clearfix

При вёрстке макетов часто требуется, чтобы размер контейнера включал в себя размеры всех дочерних элементов, в том числе плавающих. Обычно же размер контейнера НЕ включает в себя размеры плавающих дочерних элементы.

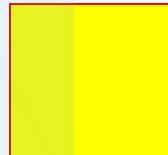
HTML-код	Отображение в браузере
<pre><style> .s-inner { float: left; width: 80px; height: 80px; background: yellow } </style> <div style="border: solid red 1px"> <div class="s-inner"></div> </div></pre>	

Проблема может быть решена выставлением стилевого свойства `overflow:hidden` или `overflow:auto` у контейнера, либо же можно сделать сам контейнер плавающим через `float:left` или `float:right`. Однако, эти способы неуниверсальны — эти стилевые свойства могут повлиять на поведение контейнера. Более универсальный способ, который мы тоже уже знаем — добавление в контейнер тега со стилевым свойством `clear:both`.

HTML-код	Отображение в браузере
<pre><style> .s-inner { float: left; width: 80px; height: 80px; background: yellow } </style> <div style="border: solid red 1px"> <div class="s-inner"></div> <div style="clear:both"></div> </div></pre>	

Недостаток такого способа, в приведённом виде — **семантический**: мы добавили HTML-тег, который НЕ имеет отображаемого смысла, НЕ задаёт ни содержание, ни логическую разметку, т.е. служебный, технический.

С помощью селектора псевдоэлемента `:after` можно сделать код семантичнее:

HTML-код	Отображение в браузере
<pre><style> .s-inner { float: left; width: 80px; height: 80px; background: yellow } .clearfix:after { content: ""; display: block; clear: both } </style> <div style="border: solid red 1px" class="clearfix"> <div class="s-inner"></div> </div></pre>	

Или, кроссбраузерный вариант:

HTML-код	Отображение в браузере
<pre><style> .s-inner { float: left; width: 80px; height: 80px; background: yellow } .clearfix:after { content: ". "; visibility: hidden; display: block; height: 0; clear: both } </style> <div style="border: solid red 1px" class="clearfix"> <div class="s-inner"></div> </div></pre>	

(6.60) Комбинированные селекторы

Можно несколько селекторов перечислить через запятую, тогда указанное стилевое оформление будет применено к тегам, подходящим под любой из перечисленных селекторов.

HTML-код	Отображение в браузере
<pre><style> a[href^='http://'], a[href^='https://'] { color: red; } a:not([href^='http://']):not([href^='https://']) { color: green; } </style> Эта ссылка абсолютная,
 и эта ссылка тоже,
 а вот эта ссылка относительная.</pre>	Эта <u>ссылка</u> абсолютная, и эта <u>ссылка</u> тоже, а вот эта <u>ссылка</u> относительная.
<pre><style> a[href*='//'] { color: red; } a:not([href*='//']) { color: green; } </style> Эта ссылка абсолютная,
 и эта ссылка тоже,
 и эта ссылка абсолютная,
 а вот эта ссылка относительная.</pre>	Эта <u>ссылка</u> абсолютная, и эта <u>ссылка</u> тоже, и эта <u>ссылка</u> абсолютная, а вот эта <u>ссылка</u> относительная.

(6.70) Веса стилевых селекторов

В случае, когда для какого-либо элемента есть сразу несколько стилевых описаний, которые к нему применяются, будут применены те стилевые свойства, стилевые описания которых имеют **больший вес**.

HTML-код	Отображение в браузере
<pre><style> p { color: red; font-style: italic } div p { color: blue } </style> <p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <div> <p>Один из самых известных и узнаваемых спортсменов в истории.</p> <p>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мухаммед Али.</p> </div></pre>	<p><i>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</i></p> <p><i>Один из самых известных и узнаваемых спортсменов в истории.</i></p> <p><i>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мухаммед Али.</i></p>

В случае указания стилевых описаний в элементе **style**, вес **каждого** стилевого описания зависит от селективности (выборочности, точности) его селектора. Чем точнее селектор, тем больше вес стилевых описаний, ему соответствующих.

Для простоты, можно считать так:

- каждый тег в селекторе добавляет к весу стилевого описания 1;
- каждый класс и каждый псевдокласс в селекторе добавляет к весу стилевого описания 10;
- каждый идентификатор в селекторе добавляет к весу стилевого описания 100.

Например, стилевое описание с селектором **p.info span** имеет вес $1+10+1 = 12$, а стилевое описание с селектором **#queen span** — вес 101.

Если стилевое описание не вынесено в элемент **style**, а указано непосредственно в элементе в атрибуте **style**, то такое стилевое описание имеет вес 1000.

HTML-код	Отображение в браузере
<pre><style> p { color: red } div p { color: blue; font-style: italic } </style> <p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <div> <p style="color: green">Один из самых известных и узнаваемых спортсменов в истории.</p> <p>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мухаммед Али.</p> </div></pre>	<p><i>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</i></p> <p><i>Один из самых известных и узнаваемых спортсменов в истории.</i></p> <p><i>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мухаммед Али.</i></p>

При указании в любом стилевом описании специального признака **!important**, вес стилевого описания увеличивается в 10.000 раз.

HTML-код	Отображение в браузере
<pre><style> p { color: red !important } div p { color: blue; font-style: italic } </style> <p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <div> <p style="color: green">Один из самых известных и узнаваемых спортсменов в истории.</p> </div></pre>	<p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <p><i>Один из самых известных и узнаваемых спортсменов в истории.</i></p>

Браузер, когда ищет для элемента значение какого-либо стилевого свойства, ищет его по следующему алгоритму:

1. Проверяет стилевые правила в теге `<style>`, селекторы которых подходят к данному элементу, а также атрибут `style` этого элемента — если данное стилевое свойство где-либо указано, берётся его значение; если указано в нескольких местах — берётся значение из описания, имеющего наибольший вес.
2. Если п.1 не дал результатов, проверяются умолчательные стилевые описания тега в браузере; если данное стилевое свойство в них указано — берётся значение оттуда.
3. Если и п.2 не дал результатов, значения **некоторых** стилевых свойств элемента берутся из аналогичных стилевых свойств его контейнера.

Данные шаги браузер предпринимает по **каждому** стилевому свойству **каждого** элемента в отдельности.

Поэтому, если для элемента любым способом установлен его собственный стиль (п.1), нет никакой возможности повлиять на этот стиль, устанавливая стиль контейнера элемента (п.3):

HTML-код	Отображение в браузере
<pre><style> div#DDD { color: red !important } span { color: blue } </style> <div id="DDD" style="color: green !important"> синий текст </div></pre>	синий текст

(6.80) Управление режимом отображения элементов

Стилевое свойство **display** — управление режимом отображения элемента.

Некоторые возможные значения:

display: block — элемент должен отображаться как блочный (он начинается с новой строки, контент после него начинается с новой строки). Такой элемент автоматически *растягивается на всю ширину контейнера, и имеет такую высоту, какая нужна, чтобы вместить всё его содержимое*.

display: inline — элемент должен отображаться как текстовый (внутристочный).

Понятия «ширина» и «высота» для него не имеют смысла.

display: inline-block — элемент должен отображаться как блочный, но не с новой строки, а в той же строке, т.е. такие элементы, будучи блочными, выстраиваются в одну строку (выровненную по базовой линии строки). Такой элемент, в отличие от блочных, не растягивается на всю доступную ему ширину, а имеет такую ширину и высоту, какие необходимы, чтобы вместить всё его содержимое.

display: inline-table — элемент должен отображаться как таблица, но в той же строке.

display: none — элемент и всё его содержимое не отображаются и не занимают места на веб-странице. Данное значение может быть применено для удаления ячеек таблицы.

Также возможны значения **display: list-item**, **display: table-row** и другие.

HTML-код	Отображение в браузере
<p>У меня живет козленок, я сама его пасу.</p><p>Я козленка в сад зеленый рано утром отнесу.</p><p>Он заблудится в саду — я в траве его найду.</p>	У меня живет козленок, я сама его пасу. Я козленка в сад зеленый рано утром отнесу. Он заблудится в саду — я в траве его найду.
<p style="display: inline">У меня живет козленок, я сама его пасу.</p><p style="display: inline">Я козленка в сад зеленый рано утром отнесу.</p><p style="display: inline">Он заблудится в саду — я в траве его найду.</p>	У меня живет козленок, я сама его пасу. Я козленка в сад зеленый рано утром отнесу. Он заблудится в саду — я в траве его найду.
У меня живет козленок, я сама его пасу.Я козленка в сад зеленый рано утром отнесу.Он заблудится в саду — я в траве его найду.	У меня живет козленок, я сама его пасу. Я козленка в сад зеленый рано утром отнесу. Он заблудится в саду — я в траве его найду.
У меня живет козленок, я сама его пасу.Я козленка в сад зеленый рано утром отнесу.Он заблудится в саду — я в траве его найду.	У меня живет козленок, я сама его пасу. Я козленка в сад зеленый рано утром отнесу. Он заблудится в саду — я в траве его найду.

HTML-код	Отображение в браузере
<pre><p style="display: inline-block; width: 110px; border: solid red 1px">У меня живет козленок.</p> <p style="display: inline-block; width: 100px; border: solid red 1px">Я козленка в сад зеленый отнесу.</p> <p style="display: inline-block; width: 120px; border: solid red 1px">Он заблудится в саду.</p></pre>	<p>У меня живет козленок.</p> <p>Я козленка в сад зеленый отнесу.</p> <p>Он заблудится в саду.</p>

Элементы с `display: inline-block` по умолчанию выравниваются по базовой линии текста контейнера, однако, это можно изменить стилевым свойством `vertical-align` элемента (не контейнера!).

HTML-код	Отображение в браузере
<pre><p style="display: inline-block; width: 50px; border: solid red 1px"> У меня</p> бу <p style="display: inline-block; width: 90px; border: solid red 1px"> Я козленка в сад зеленый отнесу.</p></pre>	<p>У меня</p> <p>бу</p> <p>Я козленка в сад зеленый отнесу.</p>
<pre><p style="display: inline-block; width: 50px; border: solid red 1px; vertical-align: top">У меня</p> <p style="display: inline-block; width: 90px; border: solid red 1px"> Я козленка в сад зеленый отнесу.</p> <p style="display: inline-block; width: 50px; border: solid red 1px; vertical-align: 20px">в саду</p></pre>	<p>У меня</p> <p>Я козленка в сад зеленый отнесу.</p> <p>в саду</p>

(6.85) Управление видимостью элементов

Стилевое свойство **visibility** — видимость элемента.

Возможные значения:

visibility: visible — видимый элемент (как обычно).

visibility: hidden — элемент и всё его содержимое невидимы, при этом они **занимают положенное им место на веб-странице**, в отличие от **display: none**.

Есть также ещё значения **collapse**, **empty-cells**.

HTML-код	Отображение в браузере
<pre>Ничего не вижу. Ничего не слышу. Никому ничего не скажу.</pre>	Ничего не вижу. Ничего не слышу. Никому ничего не скажу.
<pre>Ничего не вижу. Ничего не слышу. Никому ничего не скажу.</pre>	Ничего не вижу. Никому ничего не скажу.
<pre>Ничего не вижу. Ничего не слышу. Никому ничего не скажу.</pre>	Ничего не вижу. Никому ничего не скажу.

Но, если внутри невидимого (**visibility: hidden**) контейнера есть элемент с явно указанным **visibility: visible**, этот элемент и всё его содержимое станут видимыми.

HTML-код	Отображение в браузере
<pre><div style="visibility: hidden"> Ничего не вижу. Ничего не слышу. Никому ничего не скажу. </div></pre>	Ничего не слышу.

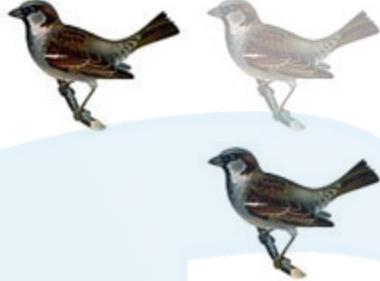
Стилевое свойство **opacity** — непрозрачность элемента и всего его содержимого.

Возможные значения:

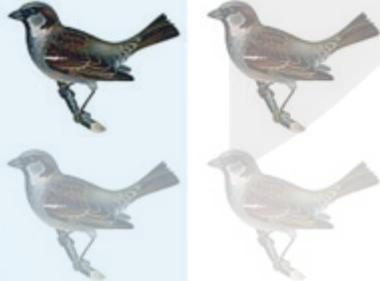
opacity: непрозрачность, где **непрозрачность** — дробное число от 0 до 1.

Значение 1 означает полностью видимый элемент,

значение 0 — полностью невидимый (аналогично `visibility: hidden`).

HTML-код	Отображение в браузере
<pre> </pre>	

Если внутри полупрозрачного контейнера есть полупрозрачные элементы, итоговая непрозрачность элементов умножается на непрозрачность контейнера.

HTML-код	Отображение в браузере
<pre> <div style="opacity: 0.4"> </div></pre>	

Домашнее задание №6

Дан макет горизонтального меню.

Придумать несколько способов сверстать данный макет.

Т.е. найти различные способы расположить несколько ссылок в ряд по горизонтали; найти различные способы сделать красные разделители.

Создать проект *LINKS*. Добавить в него данное меню, свёрстанное всеми найденными способами; каждый способ — в своём *div*-контейнере.

Контейнеру нужно указывать класс; **внутри контейнера — не использовать классов и стилей** (см. слайд 6.47 «Иерархические селекторы в вёрстке макетов»).

Т.е. меню должно появиться несколько раз подряд, с некоторым интервалом, не обязательно три раза, как показано на макете, а сколько раз, сколько способов удастся найти. Точного соответствия макету добиваться необязательно.

В HTML-комментариях пояснить, каким способом ссылки расположены в строку, каким способом сделаны разделители, а также достоинства и недостатки каждого найденного способа.

При поднесении мыши текст ссылок должен становиться красным.

Ссылки могут вести на любые URLs.



Домашнее задание №6b

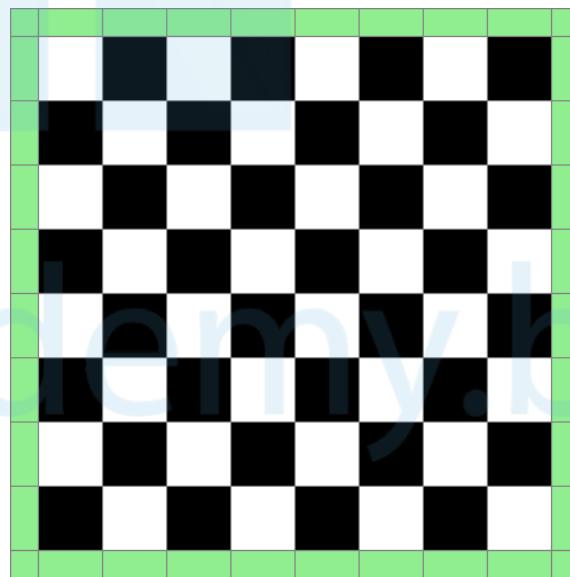
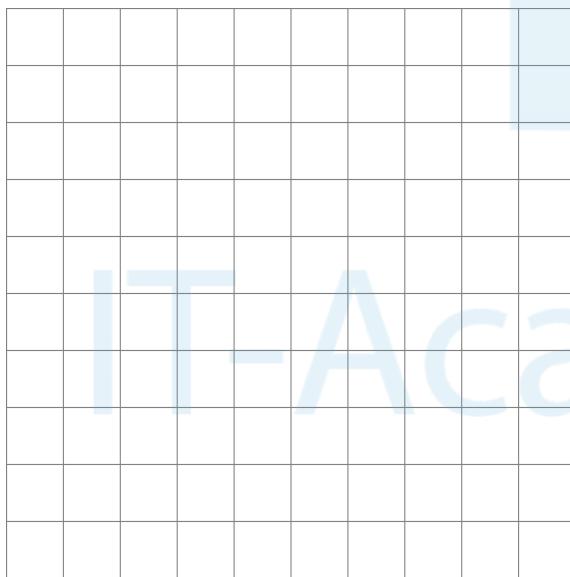
Создать проект *CSS_CHESS*. Добавить в него файл *CSS_chess.html* из библиотеки.

В файле *CSS_chess.html* — HTML-вёрстка простой таблицы 10x10 (левый рисунок).

Ничего не меняя в HTML-коде, добавляя только стилевые описания в тег *STYLE*, привести таблицу к виду, приведённому на правом рисунке.

Селекторы псевдоклассов *:not*, *:nth-*(i)* и *:nth-*(an+b)* не использовать.

Селекторы псевдоклассов *:first-**, *:last-**, *:nth-*(odd)* и *:nth-*(even)* использовать можно.



Домашнее задание №6с

Добавить в проект CHESS из библиотеки файл extref.png.

Доверстать проект по макету, указанному ниже (слева от **внешних ссылок** появились значки).

Изменения можно вносить только в тег **STYLE**.

Реализовать задание двумя способами (на двух разных страницах) — с использованием селектора псевдоэлемента `::before` и без селекторов псевдоэлементов вообще.

Гарри Каспаров

Дата рождения: 13 апреля 1963 г.

Чемпион мира: 1985-1993

Информация

Первый матч за звание чемпиона мира по шахматам Гарри Каспаров играл против чемпиона мира Анатолия Карпова. До этого они в разных официальных соревнованиях сыграли три партии, которые закончилисьничью. Для победы требовалось первым выиграть 6 партий. Такой регламент был введен в феврале 1977 года и по нему прошли два матча Карпова с Корчным. Матч начался 10 сентября 1984 года в Москве. Уже после девятой партии Карпов вел в счёте 4:0, и в последующих партиях Каспаров изменил тактику: он стал играть на ничью в каждой партии и вынуждать Карпова играть за другой цвет против своих любимых схем. Последовала серия из семнадцати ничьих, но двадцать седьмую партию снова выиграл Карпов, которому теперь оставалось одно очко до победы в матче. Каспаров «размочил» счёт в тридцать второй партии. В сорок первой партии Карпов был близок к выигрышу, но упустил его, а сорок седьмую и сорок восьмую партии выиграл Каспаров. При счёте 5:3 15 февраля 1985 года президент ФИДЕ Флоренсио Кампоманес на пресс-конференции объявил о прекращении матча, мотивировав это исчерпанием физических и психических ресурсов участников, и повторном матче между этими же соперниками в 1985 году. При этом и Карпов, и Каспаров выразили готовность продолжать матч; Каспаров на той же пресс-конференции обвинил Кампоманеса, что тот решил прервать матч только тогда, когда претендент получил шансы на победу.

[в начало страницы](#)

Домашнее задание №6d

Расположить на странице пять незакрашенных звёзд.

При поднесении мыши к любой из звёзд должна закрашиваться эта звезда и все что расположены левее неё.

Вёрстка должна быть универсальной и сработать с любым количеством звёздочек, а не только ровно с пятью.

Картинки звёздочек есть в библиотеке — файлы star0.png и star1.png.



IT-Academy.by

Тема 7. Медиазапросы. Адаптивная вёрстка

(7.20) Стилевая директива @media

Стилевая директива **@media** позволяет некоторые стилевые описания применять лишь в том случае, если отображение веб-страницы происходит на конкретном типе (типах) устройств отображения.

HTML-код	Отображение в браузере
<pre><style> p { color: blue; font-style: italic } @media print { h2 { text-align: center } p { color: black !important } } </style> <h2>Мохаммед Али</h2> <p>Мохаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <p>Один из самых известных и узнаваемых спортсменов.</p> <p>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мохаммед Али.</p></pre>	<p>Мохаммед Али</p> <p>Мохаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <p>Один из самых известных и узнаваемых спортсменов.</p> <p>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мохаммед Али.</p>
	<p>Отображение на принтере</p> <p>Мохаммед Али</p> <p>Мохаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p> <p>Один из самых известных и узнаваемых спортсменов.</p> <p>После чемпионского боя с Сонни Листоном сменил имя на Кассиус Икс, затем на Мохаммед Али.</p>

Типы устройств отображения

тип	описание
all	все типы устройств
screen	мониторы
tv	телевизоры
projection	проекторы
handheld	наладонные компьютеры
tty	устройства с фиксированным размером символов — телетайпы, терминалы... (начиная с CSS3)
print	печатающие устройства, принтеры
aural	речевые синтезаторы, речевые браузеры (до CSS3)
speech	речевые синтезаторы, речевые браузеры (начиная с CSS3)
braille	устройства отображения, основанные на системе Брайля
embossed	принтеры, основанные на системе Брайля (начиная с CSS3)

(7.25) Медиазапросы

Стилевая директива **@media** также позволяет использовать **медиазапросы**:

@media устройство and (медиаусловие)

Медиазапросы позволяют некоторые стилевые описания применять лишь в том случае, если устройство отображения имеет некоторые явно указанные характеристики, т.е. соответствует **медиаусловию** (например, имеет портретную ориентацию, имеет цветной экран, имеет ширину не менее 7 сантиметров, имеет высоту не менее 300 пикселей и т.п.).

Некоторые возможные медиаусловия:

max-device-width: размер — экран устройства отображения имеет ширину не более указанного размера (считается устаревшим).

min-device-width: размер — экран устройства отображения имеет ширину не менее указанного размера (считается устаревшим).

max-device-height: размер — экран устройства отображения имеет высоту не более указанного размера (считается устаревшим).

min-device-height: размер — экран устройства отображения имеет высоту не менее указанного размера (считается устаревшим).

(на некоторых устройствах device-width — всегда короткая сторона экрана устройства, а device-height — всегда длинная, независимо от ориентации экрана;
на некоторых устройствах device-height зависит от того, открыта ли экранная клавиатура)

max-width: размер — окно отображения (например, окно браузера) имеет ширину не более указанного размера.

min-width: размер — окно отображения имеет ширину не менее указанного размера.

max-height: размер — окно отображения имеет высоту не более указанного размера.

min-height: размер — окно отображения имеет высоту не менее указанного размера.

orientation: landscape — экран устройства имеет ландшафтную ориентацию.

orientation: portrait — экран устройства имеет портретную ориентацию.

max-device-aspect-ratio: число/число — отношение ширины устройства отображения к высоте не более указанного (считается устаревшим).

min-device-aspect-ratio: число/число — отношение ширины устройства отображения к высоте не менее указанного (считается устаревшим).

max-aspect-ratio: число/число — отношение ширины окна к высоте не более указанного.

min-aspect-ratio: число/число — отношение ширины окна к высоте не менее указанного.

min-resolution: разрешение — экран устройства имеет разрешение не менее указанного. Разрешение указывается в единицах **dpi** (пикселов CSS на дюйм экрана),

dpcm (пикселов CSS на сантиметр экрана), **dppx** (пикселов CSS на пиксел экрана).
max-resolution: разрешение — экран устройства имеет разрешение не более указанного.

(на некоторых устройствах зависит от текущего масштаба отображения страницы;
на некоторых устройствах может называться min-device-pixel-ratio и max-device-pixel-ratio)

monochrome — экран устройства является монохромным (отображает оттенки только одного цвета).

color — экран устройства является цветным.

min-color-index: количество_цветов — экран устройства поддерживает количество цветов (либо оттенков цвета) не менее указанного.

max-color-index: количество_цветов — экран устройства поддерживает количество цветов (либо оттенков цвета) не более указанного.

grid — устройство поддерживает только моноширинные шрифты.

Если для одного типа устройств требуется исполнение сразу нескольких медиаусловий, медиаусловия следует соединять через **and**:

@media устройство and (медиаусловие) and (медиаусловие) and (медиаусловие)

например:

```
@media screen and (min-width: 800px) and (color) { ... }
```

Если требуется исполнение одного из нескольких медиаусловий, медиаусловия следует соединять через **запятую**; в каждом медиаусловии следует указывать тип устройства, неуказанный тип устройства приравнивается к **all**:

@media устройство and (медиаусловие), устройство and (медиаусловие)

например:

```
@media screen and (min-width: 800px), print and (min-width: 20cm) and (color) { ... }  
@media screen and (min-width: 800px), screen and (min-height: 500px) { ... }
```

Старые браузеры, не поддерживающие медиазапросы, применяют указанные в директиве @media стилевые описания, игнорируя медиазапросы. Для избежания такого эффекта, нужно список устройств предварять ключевым словом **only**:

@media only устройство and (медиаусловие)

Стилевые описания, указанные в такой директиве, будут обработаны только браузерами, которые поддерживают медиазапросы.

(7.90) Стилевые таблицы в отдельных файлах

Подключение на веб-страницу стилевых описаний из внешнего файла:

```
<link rel="stylesheet" href="file.css" type="text/css">
```

Подключение на веб-страницу стилевых описаний, предназначенных только для некоторых устройств отображения, из внешнего файла:

```
<link rel="stylesheet" href="file.css" type="text/css" media="screen, print">
```

Подключение на веб-страницу стилевых описаний, предназначенных только для устройств отображения с конкретными параметрами, из внешнего файла:

```
<link rel="stylesheet" href="file.css" type="text/css" media="screen and (min-width: 30em)">
```

Тег **link**, как правило, должен быть дочерним для тега **head**.

HTML-код	Отображение в браузере
<pre><link rel="stylesheet" href="styles/ex735.css" type="text/css"> <h2>Мухаммед Али</h2> <p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p></pre>	<p>Мухаммед Али</p> <p><i>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</i></p>

Включение в стилевые описания дополнительных стилевых описаний из внешнего файла:

```
@import "file.css";
```

Включение в стилевые описания дополнительных стилевых описаний из внешнего файла, предназначенных только для некоторых устройств отображения:

```
@import "file.css" screen, print;
```

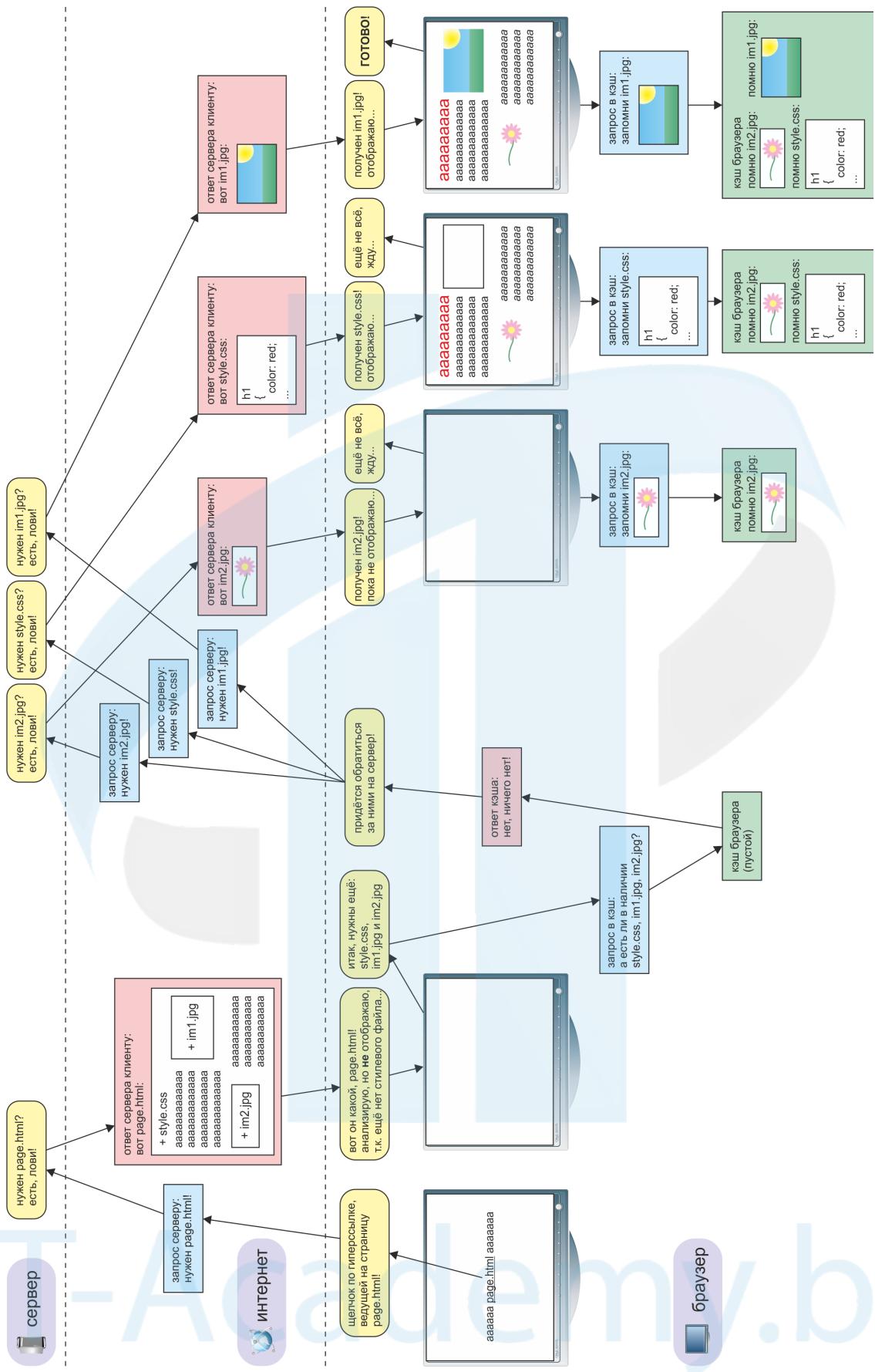
Стилевая директива **@import** должна быть первой в стилевых описаниях — до неё не должно быть никаких других стилевых описаний.

HTML-код	Отображение в браузере
<pre><style> @import "styles/ex735.css" </style> <h2>Мухаммед Али</h2> <p>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</p></pre>	<p>Мухаммед Али</p> <p><i>Мухаммед Али — американский боксёр-профессионал, выступавший в тяжёлой весовой категории.</i></p>

Стилевые файлы, указанные в тегах **link** с указанием устройства и/или медиаусловия, или указанные в директивах **@import** с указанием устройства и/или медиаусловием, будут **загружены** браузером с сервера в любом случае, даже если отображение идёт на другом устройстве и медиаусловие не выполняется (но, конечно, указанные в таких файлах стилевые описания будут **применены** только если отображение идёт на указанном устройстве и медиаусловие выполняется).

Изображения в стилевых свойствах **background-image**, расположенные в таких стилевых файлах или в директивах **@media** с указанием устройства и/или медиаусловия, будут **загружены** браузером с сервера только в том случае, если отображение идёт на указанном устройстве и медиаусловие выполняется.

(7.92) Порядок взаимодействия браузера с сервером с сервером, если на странице используется изображения и стилевого файла



Домашнее задание №7

Создать проект *MAKET_ADAPTER*.

Сверстать проект по макету из домашнего задания №5 адаптивной вёрсткой.

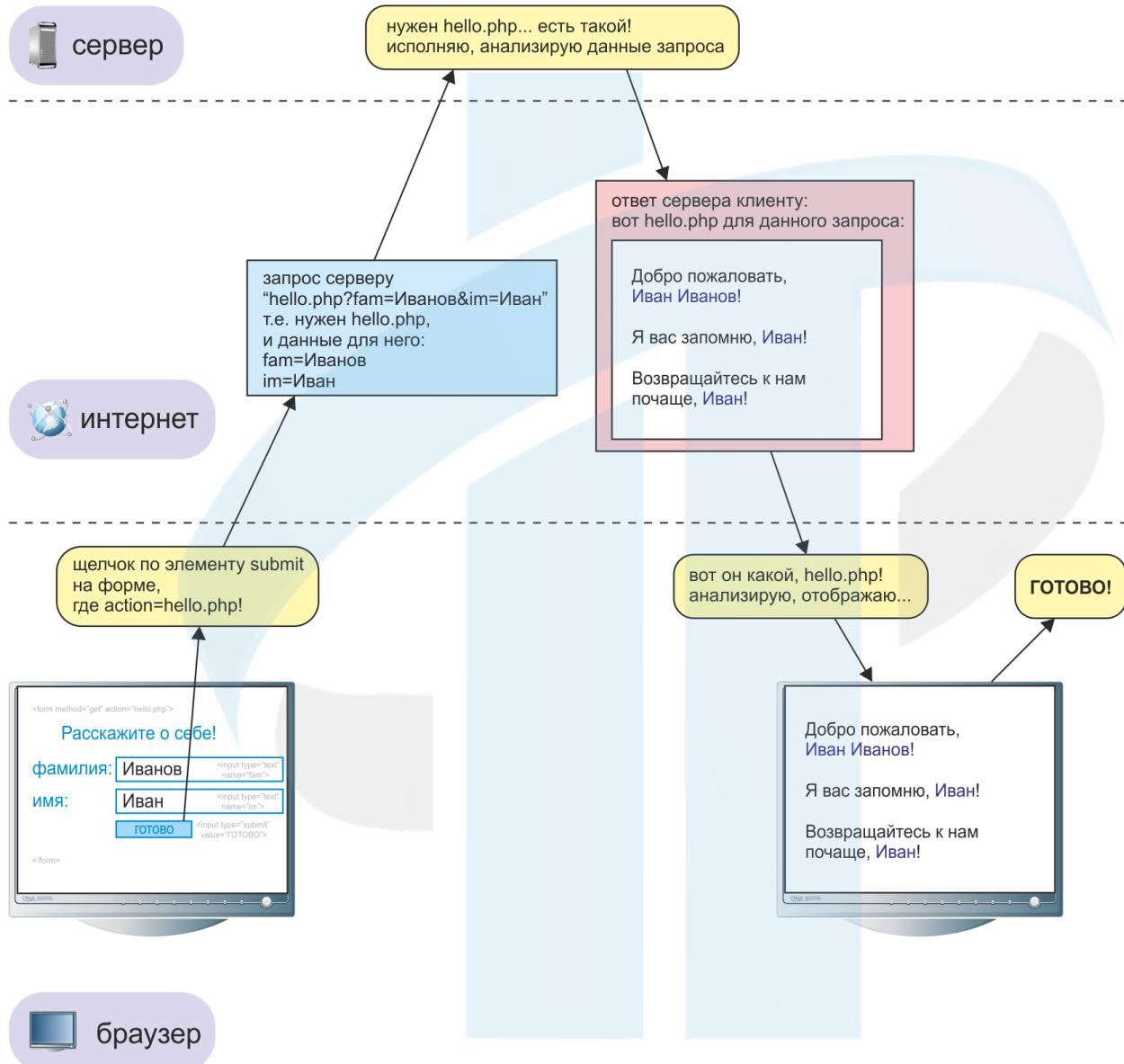
Самостоятельно определить точки перехода (при какой ширине какие изменения макета должны происходить).



IT-Academy.by

Тема 8. Формы. Позиционирование. HTML5. Мультимедиа

(8.10) Порядок взаимодействия браузера с сервером при отправке формы



(8.15) Формы

Парный тег **form** — форма.

Тег **form** используется для создания формы ввода данных — интерактивного механизма взаимодействия веб-сайта с пользователем.

Атрибут **action** тега **form** задаёт URL **скрипта-обработчика** — серверного скрипта, который будет обрабатывать введённые пользователем данные и вернёт ответ в виде новой веб-страницы, содержащей результат обработки.

После отправки данных формы на сервер всегда возвращается новая веб-страница.

По сути отправка формы на сервер аналогична переходу по гиперссылке, только переход осуществляется на страницу, содержимое которой формируется скриптом-обработчиком «на лету» на основе данных, введённых в форму.

Атрибут **target** тега **form** указывает, в какой вкладке браузера должна открываться эта новая веб-страница (аналогично атрибуту **target** у ссылок).

Атрибут **method** тега **form** задаёт способ передачи введённых данных скрипту-обработчику:

- **get** — данные передаются прямо в URL, т.е. URL состоит из имени скрипта-обработчика и перечня введённых в форму именованных значений;
- **post** — данные передаются в теле запроса, а в URL — только имя скрипта-обработчика.

Считается правильным использовать метод **get**, если скрипт-обработчик при обработке данных формы не вносит никаких изменений (например в базы данных, в файлы), а только возвращает прочитанные откуда-либо данные в виде ответной веб-страницы, т.е. можно отправлять эту форму на сервер много раз, получая один и тот же результат. Также, т.к. в URL содержатся все необходимые для работы скрипта-обработчика данные, такой URL можно отправлять не из формы, а оставлять где угодно в виде обычной гиперссылки (метод **get** — это тот метод, которым запрашивается новая страница с сервера при любом переходе по обычной гиперссылке). Также, страницу с таким URL можно освежать сколько угодно раз.

Считается правильным использовать метод **post**, если скрипт-обработчик при обработке данных формы вносит куда-либо какие-либо изменения (например в базы данных, в файлы), и потом формирует ответную веб-страницу. Также, т.к. в URL не содержатся необходимые для работы скрипта-обработчика данные (а только имя скрипта-обработчика), такой URL невозможно использовать в виде обычной гиперссылки (скрипт-обработчик не сможет работать, т.к. ожидает в теле запроса нужные ему данные, а тело запроса пустое).

Однако, в некоторых случаях метод выбирается по другим критериям. Например, существует ограничение на объём данных, которые можно передать методом **get**, т.е. ограничения на размер URL. Численно они зависят от браузера и сервера, т.е. фактически непредсказуемы. Если есть вероятность, что в поля формы будут вводиться хотя бы сотни символов, возможно, следует использовать метод **post**.

Внутри тега **form** должны содержаться элементы формы (поля формы) — теги **input**, **select**, **textarea** и другие.

Каждый элемент формы должен содержать атрибут **name** — имя элемента, это то имя, под которым скрипт-обработчик получит введённое в это поле значение.

Имя серверного скрипта-обработчика, метод передачи данных (**get** или **post**), типы (а часто и имена) элементов формы не выбираются frontend-разработчиком (верстальщиком), а являются частью технического задания на разработку веб-страницы, т.к. их же должен использовать и backend-разработчик (разработчик серверного скрипта-обработчика).

Одиночный тег **input** внутри тела **form** — элемент формы, конкретный вид которого зависит от атрибута **type**.

Некоторые возможные значения атрибута **type**:

- **text** — поле ввода текста.

Атрибут **value** задаёт начальное значение текста.

Атрибут **maxlength** задаёт максимальное количество символов, которые можно ввести в поле.

Атрибут **readonly** (без значения) делает поле доступным только для чтения (значение в поле нельзя изменить); на внешний вид элемента этот атрибут не влияет.

Атрибут **placeholder** позволяет задать текст, который будет появляться в незаполненном элементе ввода текста как подсказка о том, что именно нужно вводить в поле (этот атрибут поддерживается не везде — Chrome, Firefox, Opera 11.5+, IE 10+).

- **password** — поле ввода пароля (вводимые символы отображаются звёздочками).

Атрибут **maxlength** задаёт максимальное количество символов, которые можно ввести в поле.

- **checkbox** — птичка (может быть установленной либо сброшенной).

Атрибут **checked** (без значения) задаёт начальное состояние птички.

Атрибут **value** задаёт значение, которое будет отправлено на сервер при установке птички; по умолчанию это значение 'on'.

- **radio** — поле для выбора одного из нескольких вариантов. Каждый из вариантов должен иметь одно и то же значение **name**, и разные значения **value**.

Атрибут **checked** (без значения) указывает, какой из вариантов должен быть выбран изначально.

- **file** — поле для выбора файла с компьютера пользователя для отправки на сервер. Для правильной работы элемента в теле **form** обязательно должны быть указаны атрибуты **method=post** и **enctype="multipart/form-data"**.

- **hidden** — невидимый элемент. Предназначен для отправки на сервер дополнительных данных, которые требуются скриптом-обработчиком и известны разработчику веб-страницы, но не должны вводиться пользователем напрямую. Атрибут **value** задаёт значение элемента.

- **submit** — кнопка отправки формы (т.е. всех элементов и их значений) серверному скрипту.

Атрибут **value** задаёт подпись на кнопке.

- **reset** — кнопка сброса формы (т.е. значения всех элементов возвращаются к тому состоянию, которое было при открытии формы).

Атрибут **value** задаёт подпись на кнопке.

- **button** — кнопка без предзаданного поведения (часто используется для вызова JavaScript).

Атрибут **value** задаёт подпись на кнопке.

Атрибут **autofocus** тела **input** (без значения) автоматически фокусирует данный элемент формы при загрузке страницы (не работает в Internet Explorer до версии 10).

Атрибут **disabled** тела **input** (без значения) делает элемент формы недоступным.

Обычно такие элементы формы имеют серый цвет текста и/или фона. Этот атрибут можно использовать с любым типом элемента **input**. Значения из недоступных полей формы не передаются на сервер.

HTML-код	Отображение в браузере
<pre><form method="get" action="Search.php"> Какую машину ищете:
 <input type="text" name="carmodel">
 Только новые: <input type="checkbox" name="onlynew" checked>
 <input type="submit" value="Поиск"> </form></pre>	<p>Какую машину ищете:</p> <input type="text"/> <p>Только новые: <input checked="" type="checkbox"/></p> <p><input type="button" value="Поиск"/></p>

При вводе машины **Жигули**, установке птички «Только новые» и нажатии кнопки **Поиск** будет переход на страницу:
Search.php?carmodel=Жигули&onlynew=on

При вводе машины **Mercedes**, сбросе птички «только новые» и нажатии кнопки **Поиск** будет переход на страницу:
Search.php?carmodel=Mерcedes

HTML-код	Отображение в браузере
<pre><form method="post" action="Vote.jsp"> Проголосуйте:
 <input type="radio" name="answer" value="1">да
 <input type="radio" name="answer" value="2" checked>нет
 <input type="radio" name="answer" value="3">воздержался
 <input type="hidden" name="votenum" value="111"> <input type="submit" value="Голосовать!"> </form></pre>	<p>Проголосуйте:</p> <p><input type="radio"/> да <input checked="" type="radio"/> нет <input type="radio"/> воздержался</p> <p><input type="button" value="Голосовать!"/></p>

При вводе ответа **воздержался** и нажатии кнопки **Голосовать!** будет переход на страницу:
Vote.jsp
и в запросе к странице также будут переданы значения:
answer=3
votenum=111

HTML-код	Отображение в браузере
<pre><form method="post" enctype="multipart/form-data" action="Fot.asp"> Фото:
 <input type="file" name="myphoto">
 <input type="text" name="comment" style="width: 200px" placeholder="коммент к фото">
 Ваш пароль: <input type="password" name="pass"> <input type="submit" value="Опубликовать"> </form></pre>	<p>Фото:</p> <p><input type="file"/> Выберите файл Файл не выбран</p> <p><input type="text"/> коммент к фото</p> <p>Ваш пароль:</p> <p><input type="button" value="Опубликовать"/></p>

При выборе файла с фото, комментария а теперь волна накрыла меня ТАК, пароля **qwerty** и нажатии кнопки **Опубликовать** будет переход на страницу:
Fot.asp
и в запросе к странице также будут переданы значения:
myphoto=(содержимое файла с фотографией)
comment=а теперь волна накрыла меня ТАК
pass=qwerty

Парный тег **textarea** внутри тела **form** — элемент формы, позволяющий вводить многострочный текст. Содержимое тела **textarea** задаёт начальное значение текста. Атрибуты **disabled**, **readonly**, **maxlength**, **placeholder** работают аналогично элементу **<input type="text">**.

HTML-код	Отображение в браузере
<pre><form method="post" action="Reply.php"> Имя: <input type="text" name="myname" value="Толик">
 Вы уже большой? <input type="checkbox" name="big" value="YES">
 Ответ:
 <textarea name="mytext">Я считаю, что...</textarea>
 <input type="submit" value="Отправить"> </form></pre>	<p>Имя: Толик</p> <p>Вы уже большой? <input type="checkbox"/></p> <p>Ответ:</p> <p>Я считаю, что...</p> <p>Отправить</p>

При вводе имени **Аноним**, установке птички «большой» и вводе ответа **Всё это маловероятно** и нажатии кнопки **Отправить** будет переход на страницу:

Reply.php

и в запросе к странице также будут переданы значения:

myname=Аноним

big=YES

mytext=Всё это маловероятно

Парный тег **select** внутри тела **form** — выпадающий список. Должен содержать элементы **option** и, при необходимости, **optgroup**.

Атрибут **multiple** (без значения) позволяет пользователю выбирать сразу несколько позиций из списка; при этом список перестаёт быть выпадающим.

Атрибут **size** позволяет установить количество отображаемых в списке строк, если список невыпадающий (т.е. при наличии атрибута **multiple**).

Парный тег **option** внутри тела **select** — один из элементов выпадающего списка.

Содержимое тела **option** и будет текстом в выпадающем списке.

Атрибут **value** задаёт значение, которое получит элемент при выборе данной строки в выпадающем списке.

Атрибут **disabled** (без значения) делает элемент списка недоступным для выбора.

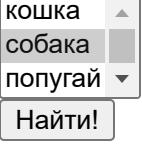
HTML-код	Отображение в браузере
<pre><form method="get" action="FindAnimal.php"> Имя вашего домашнего питомца:
 <input type="text" name="animalname">
 Что за животное:
 <select name="animalkind"> <option value="7">кошка</option> <option value="5">собака</option> <option value="8" selected>попугай</option> <option value="9" disabled>тушкан</option> </select>
 <input type="submit" value="Найти!"> </form></pre>	<p>Имя вашего домашнего питомца:</p> <p>Что за животное:</p> <p>попугай <input checked="" type="button"/></p> <p>Найти!</p>

При вводе имени **Портусик**, животного **собака** и нажатии кнопки **Найти!** будет переход на страницу:

FindAnimal.php?animalname=Портусик&animalkind=5

Атрибут **size** тега **select** (если значение атрибута больше чем 1) превращает список из выпадающего в прокрутивающийся.

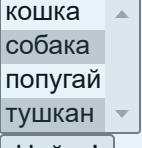
Список будет иметь такую высоту, чтобы вместить указанное количество элементов.

HTML-код	Отображение в браузере
<pre><form method="get" action="FindAnimal.php"> Что за животное:
 <select name="animalkind" size="3"> <option value="7">кошка</option> <option value="5" selected>собака</option> <option value="8">попугай</option> <option value="9">тукан</option> </select>
 <input type="submit" value="Найти!"> </form></pre>	Что за животное: 

При выборе животного **кошка** и нажатии кнопки **Найти!** будет переход на страницу: <FindAnimal.php?animalkind=7>

Атрибут **multiple** (без значения) тега **select** также превращает список из выпадающего в прокрутивающийся и позволяет выбирать сразу несколько элементов списка.

Если не задан ещё и атрибут **size**, список будет иметь достаточную высоту, чтобы вместить все элементы.

HTML-код	Отображение в браузере
<pre><form method="get" action="FindAnimal.php"> Что за животное:
 <select name="animalkind" multiple> <option value="7">кошка</option> <option value="5" selected>собака</option> <option value="8">попугай</option> <option value="9" selected>тукан</option> </select>
 <input type="submit" value="Найти!"> </form></pre>	Что за животное: 

При выборе животных **кошка** и **собака** и нажатии кнопки **Найти!** будет переход на страницу:

<FindAnimal.php?animalkind=5&animalkind=7>

Парный тег **button** позволяет расположить на форме такую кнопку, внутри которой можно разместить произвольный HTML-код.

Атрибут **type** тела **button** задаёт поведение кнопки:

- **submit** — отправка формы (т.е. всех элементов и их значений) серверному скрипту.
- **reset** — сброс формы (т.е. значения всех элементов возвращаются к тому состоянию, которое было при открытии формы).
- **button** — кнопка без предзданного поведения;

Атрибут **disabled** (без значения) запрещает нажатие кнопки.

HTML-код	Отображение в браузере
<pre><form method="get" action="FindBook.php"> Название книги: <input type="text" name="bookname">
 <button type="submit"> Поиск
 начать!
 </button> </form></pre>	<p>Название книги: <input name="bookname" type="text"/></p> <p>Поиск начать!</p> 

При вводе названия **Всё о вёрстке** и нажатии кнопки **Поиск начать!** будет переход на страницу:

FindBook.php?bookname=Всё+о+вёрстке

Парный тег **label** позволяет создавать подписи для элементов форм.

Подписи позволяют фокусировать элемент формы при щелчке по подписи либо при нажатии горячей клавиши.

Вариант №1 — элемент формы включается внутрь тела **label**:

HTML-код	Отображение в браузере
<pre><form method="post" action="Login.asp"> <label>Ваше имя: <input type="text" name="nam"> </label>
 <label>Ваша фамилия: <input type="text" name="fam"> </label> </form></pre>	<p>Ваше имя: <input name="nam" type="text"/></p> <p>Ваша фамилия: <input name="fam" type="text"/></p>

Вариант №2 — элементу формы задаётся атрибут **id**, и он же указывается в теле **label** в атрибуте **for**:

HTML-код	Отображение в браузере
<pre><form method="post" action="Login.asp"> <label for="NNN">Ваше имя:</label> <input id="NNN" type="text" name="nam">
 <label for="FFF">Ваша фамилия:</label> <input id="FFF" type="text" name="fam"> </form></pre>	<p>Ваше имя: <input id="NNN" name="nam" type="text"/></p> <p>Ваша фамилия: <input id="FFF" name="fam" type="text"/></p>

Парный тег **fieldset** позволяет визуально оформить в виде блока несколько взаимосвязанных полей формы.

Парный тег **legend**, находящийся внутри тела **fieldset**, позволяет задать заголовок для этого блока.

HTML-код	Отображение в браузере
<pre><form method="post" action="AddAnketa.php"> <fieldset> <legend>Паспортные данные</legend> <label>Ваша фамилия:
 <input type="text" name="fam" value="Иванов"> </label>
 <label>Ваше имя:
 <input type="text" name="im" value="Иван" disabled> </label>
 </fieldset> <fieldset style="border: double red 3px"> <legend style="font-style: italic">Контакты</legend> <label>Телефон:
 <input type="text" name="tel" maxlength="10"> </label>
 <label>E-mail:
 <input type="text" name="email" value="iv@mail.ru"> </label>
 <label>Skype:
 <input type="text" name="skype" value="ivanov.i" readonly> </label>
 </fieldset> </form></pre>	<p>Паспортные данные</p> <p>Ваша фамилия: Иванов</p> <p>Ваше имя: Иван</p> <p>Контакты</p> <p>Телефон:</p> <p>E-mail: iv@mail.ru</p> <p>Skype: ivanov.i</p>

8.20) Позиционирование элементов

Стилевое свойство **position** — схема позиционирования элемента.

Это стилевое свойство позволяет задать схему позиционирования элемента, т.е. способ, которым браузер вычисляет положение элемента на странице.

Может применяться практически к любому элементу.

Возможные значения:

position: static — стандартная схема позиционирования, т.е. все элементы располагаются последовательно, один за другим, слева направо, сверху вниз.

position: relative — относительная схема позиционирования, т.е. элемент занимает место в общем потоке элементов, но отображается с заданным смещением относительно своего расчётного места.

position: absolute — абсолютная схема позиционирования, т.е. элемент не занимает место в общем потоке элементов, а отображается с заданным смещением относительно **веб-страницы** либо относительно **позиционированного предка**.

position: fixed — фиксированная схема позиционирования, т.е. элемент не занимает место в общем потоке элементов, а отображается с заданным смещением относительно **окна браузера**.

Если стилевое свойство **position** элемента равно **relative**, **absolute** или **fixed**, тогда стилевые свойства **top** и **left** позволяют задать смещение верхней и левой грани элемента от точки отсчёта, определяемой значением стилевого свойства **position**.



Если стилевое свойство **position** элемента равно **absolute** или **fixed**, можно также использовать стилевые свойства **bottom** и **right**, чтобы задать положение нижней и/или правой грани элемента относительно нижней и/или правой границы контейнера, позиционированного предка, веб-страницы или окна браузера.

HTML-код	Отображение в браузере
Опытный электрик никогда не прикасается к женщине сразу двумя руками.	Опытный электрик никогда не прикасается к женщине сразу двумя руками.
Опытный электрик никогда не прикасается к женщине сразу двумя руками.	Опытный электрик никогда не прикасается к женщине сразу двумя руками.
Опытный электрик никогда не прикасается к женщине сразу двумя руками.	Опытный электрик никогда не прикасается к женщине сразу двумя руками.

Обычно задаётся или **left** или **right**, а также или **top** или **bottom**; при одновременном задании **left** и **right** задаётся одновременно положение и левой, и правой грани элемента, а следовательно — его ширина; аналогично с **top** и **bottom**.

При расчёте размера контейнера браузер:

- учитывает площадь, занятую дочерними элементами с *position: relative*, до того как элементы будут смещены;
- вообще не учитывает площадь, занятую дочерними элементами с *position: fixed* и *position: absolute*.

HTML-код	Отображение в браузере
<div style="display: inline-block; padding: 2px; border: solid red 1px"> abc <td></td>	
<div style="display: inline-block; padding: 2px; border: solid red 1px"> abc <td></td>	
<div style="display: inline-block; padding: 2px; border: solid red 1px; position: relative"> abc <td></td>	

(8.35) Порядок наложения позиционированных элементов

В случае, когда позиционированные элементы накладываются друг на друга, стилевое свойство **z-index** позволяет задать, какой элемент ближе к глазам пользователя, а какой — дальше, т.е. задать порядок наложения элементов друг на друга.

По умолчанию **z-index** всех элементов равен 0.

Чем больше значение **z-index**, тем ближе он к глазам пользователя.

HTML-код	Отображение в браузере
<pre> </pre>	
<pre> </pre>	
<pre> </pre>	

(8.40) HTML5 — обзор

HTML5 — не просто очередная версия языка разметки HTML (HTML 5.0), а ещё и совокупность многих веб-технологий, связанных с языком HTML.

Под термином **HTML5** понимаются следующие нововведения:

- новые, в т.ч. **семантические HTML-теги**;
- новые атрибуты:
 - **contenteditable** — пользователь может редактировать содержимое элемента (даже если это, например, div);
 - **contextmenu** — задаёт для элемента особое контекстное меню,
 - **draggable** — используется для реализации drag&drop на JavaScript;
 - **hidden** — элемент скрывается;
- считаются валидными любые HTML-атрибуты, имена которых начинаются с **data-**, эти атрибуты не используются браузером, но могут использоваться скриптами на JavaScript, сторонними программами, роботами;
- поддержка **WAI-ARIA** (Web Accessibility Initiative — Accessible Rich Internet Applications) — разметка для облегчения полноценного использования веб-сайтов людьми с физическими ограничениями (используемые для WAI-ARIA атрибут **role** и атрибуты, имена которых начинаются с **aria-** также считаются валидными);
- **микроразметка** — семантическая разметка для предоставления сведений о разнообразных сущностях (микроформаты, Microdata, RDF);
- новые **возможности форм** — новые элементы форм, возможности валидации форм, возможность выбора и отправки нескольких файлов;
- новые **возможности мультимедиа** — воспроизведение видео и аудио, отображение SVG, отображение математических формул (MathML), программирование графики (Canvas), программирование звука (Web Audio);
- поддержка **Offline Web applications** — сайтов, не теряющих работоспособности при долговременной потере доступа в Интернет;
- ...и другие, пока плохо поддерживаемые браузерами нововведения.

Теги, имеющие предзаданное поведение и/или отображение

Теги **progress**, **meter** — используются для визуального представления некоего числа, принадлежащего диапазону (например, температура воздуха 25 градусов на линейке от -40 до +40 градусов; конкретное представление зависит от браузера).

Тег **wbr** — указывает место, в котором браузер может сделать перенос текста при необходимости.

Тег **picture** — позволяет указать несколько изображений разных форматов и/или размеров, браузер отобразит одно из них — которое умеет отображать, и которое подходит под медиаусловия.

Семантические теги

Тег **article** — определяет основное содержание страницы (например, текст статьи, новости), такое содержание, которое имеет смысл само по себе, даже в отрыве от остальной информации на странице.

Тег **section** — определяет некий раздел документа (например, блок новостей, раздел статьи). Его содержимое должно начинаться с заголовка (тег **h1-h6**).

Тег **hgroup** — расширенный заголовок, он должен содержать теги **h1-h6** и предназначен для группировки заголовков и подзаголовков.

Тег **aside** — определяет содержимое боковой панели (например, ссылки, метки, реклама), такое содержание, которое не требуется для понимания основного содержания страницы.

Теги **details**, **summary** — определяют дополнительное содержимое, которое может быть отображено или скрыто по желанию пользователя (например, некие пояснения), и текст, щелчок по которому раскрывает дополнительное содержимое. Тег **summary** должен быть дочерним для тега **details**. Дополнительное содержимое при открытии страницы не будет отображаться, но если для тега **details** указан атрибут **open** — дополнительное содержимое будет сразу отображено.

Теги **figure**, **figcaption** — определяют соответственно контейнер для изображения и подпись к изображению. В тег **figure** вкладываются теги **img** и **figcaption**. Таким образом, устанавливается однозначная связь между изображением и подписью к нему.

Теги **header**, **footer** — определяют соответственно шапку и подвал сайта или какой-либо структурной единицы содержимого. Например, содержимое тега **header** для статьи может включать, кроме собственно названия статьи, какой-либо логотип, фамилию автора, навигационные ссылки; содержимое тега **footer** для статьи может содержать год написания статьи, фамилию автора, информацию о копирайте.

Тег **mark** — определяет неким образом выделенный текст (отображение выделения изменяется стилями).

Теги **menu**, **command** — определяют соответственно меню и пункт меню (пункт меню может быть, например, ссылкой, кнопкой).

Тег **nav** — определяет блок ссылок, предназначенных для навигации по сайту или странице.

Тег **time** — определяет некую дату и/или время. Обычно содержимым тега является дата-время в удобочитаемом виде (например, "20 ноября сего года" или "5 минут назад"), а в атрибуте **datetime** указывается дата-время в одном из машиночитаемых форматов — "ГГГГ-ММ-ДД", "ЧЧ:ММ+чч:мм" или "ГГГГ-ММ-ДД ЧЧ:ММ+чч:мм" (маленькими буквами — смещение часового пояса). Например:

Ярмарка начнётся `<time datetime="2015-03-01">`в следующую среду`</time>` и будет работать `<time datetime="18:30+03:00">`до вечера`</time>`, подведение итогов — `<time datetime="2015-03-02T09:00+03:00">`утром`</time>`.

WAI-ARIA

WAI-ARIA (*Web Accessibility Initiative — Accessible Rich Internet Applications*) — разметка для облегчения полноценного использования веб-сайтов людьми с физическими ограничениями.

У любого тега может быть указан HTML-атрибут **role**, он указывает роль данного HTML-элемента в данном документе, что позволяет программам для людей с ограниченными возможностями лучше представлять документ пользователям.

Некоторые возможные значения атрибута **role**:

- **banner** — шапка сайта; в таком элементе могут быть расположены, например, логотип, название сайта, поле поиска, название страницы.
- **contentinfo** — подвал сайта; в таком элементе могут быть расположены, например, копирайт, меню.
- **main** — основное содержание документа; на одной странице не должно быть более одного элемента с такой ролью.
- **complementary** — важная, но не основная часть контента; например, это могут быть информационные блоки типа погоды, анонсов; чаще всего отображается на боковой полосе страницы.
- **search** — поиск; например, форма, используемая для поиска по сайту.
- **navigation** — набор навигационных элементов (обычно ссылок) для навигации по странице или сайту.

<div style='text-align: center; padding: 20px; background-color: yellow' role=banner>

ЧЕРЕПАШКИ - ЛУЧШИЙ САЙТ ИНТЕРНЕТА!

<div style='width: 15%; float: left; background-color: blue' role=complementary>

Главная
О нас
Виды черепах
Уход
Кормление
Дрессировка
Обратная связь

<div style='margin-left: 15%; margin-right: 200px; background-color: red' role=main>

Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.

Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов, распространённых в тропической и умеренной климатических зонах.

Черепахи (лат. Testudines) — один из четырёх отрядов пресмыкающихся, существующий на протяжении более 220 миллионов лет. Содержит около 230 современных видов, группируемых в 12 семейств и 5 подотрядов.

<div style='width: 200px; float: right; background-color: green'>

продадим лучший корм!
...
ухаживаем круче всех!
...
выслушаем жалобы!
...
наши друзья

<div style='clear: both; background-color: brown' role=contentinfo>

сайт создан при поддержке....
хостинг для сайта предоставлен...
сайт разработали...
материалы для сайта предоставили...

У любого тега может быть указан HTML-атрибут **aria-controls**, он указывает идентификатор элемента, которым он управляет или с которым функционально связан иным образом. Например, в теге, выглядящем как закладка, атрибут **aria-controls** может содержать идентификатор того тега-контейнера, который будет отображён при клике по закладке.

У любого тега может быть указан HTML-атрибут **aria-labelledby**, он указывает идентификатор элемента, который является меткой для данного. Например, в теге-контейнере атрибут **aria-labelledby** может содержать идентификатор тега-закладки, отображающего данный контейнер.

У любого тега могут быть указаны HTML-атрибуты, обозначающие интерфейсное состояние данного тега. Например, атрибут **aria-selected** (может принимать значения 'true' или 'false') обозначает, что элемент выбран; атрибут **aria-hidden** (может принимать значения 'true' или 'false') обозначает, что элемент скрыт.

У любого тега (даже у span) может быть указан HTML-атрибут **tabindex** с целочисленным значением, который означает, что данный элемент может быть выделен нажатием клавиши Tab на клавиатуре.

Микроразметка

Микроразметка — способ семантической разметки на веб-страницах информации о некоторых сущностях (событиях, организациях, людях, товарах и т.д.) с использованием стандартных элементов языка HTML.

Пользователь-человек через обычный браузер видит страницу с микроразметкой как обычную веб-страницу, но пользователи-программы (поисковые роботы, речевые синтезаторы и т.д.) способны извлечь из такой страницы структурированную информацию о представленных сущностях.

Есть три вида микроразметки — **микроданные (Microdata)**, **микроформаты** и **RDF**.

Микроданные (Microdata)

```
<div itemscope itemtype="http://webmaster.yandex.ru/vocabularies/enc-article.xml">
    <h2 itemprop=title>о Барто Агния Львовне</h2>
    <meta itemprop=category content="Детская литература">
    <meta itemprop=category content="Советские писатели">
    <div itemprop=content>Барто Агния Львовна, русская советская поэтесса. Начала печататься в 1925. Пишет для детей — о жизни советской школы, семьи, пионерах. Автор сборников стихов.</div>
    <div itemprop=references>Лит.: Дмитриева В., Агния Барто, М.-Л., 1953; Соловьев Б., Агния Барто, М., 1967.</div>
    <div>
        <span itemprop=author>Е. А. Таратута</span>
        <span itemprop=source>БСЭ</span>, <span itemprop=source-date>1969-1978</span>.
    </div>
    
</div>
```

Микроформаты

```
<div class=vcard>
    <div>
        <span class=category>Учебный центр</span>
        <span class="fn org">ОЦ ПВТ</span>
    </div>
    <div class=adr>
        <span class=locality>г. Минск</span>
        <span class=street-address>ул. Скрыганова, 14</span>
    </div>
    <div>Телефон: <span class=tel>+375 (29) 222-24-60</span></div>
    <div>
        Мы работаем <span class=workhours>ежедневно с 9:00 до 18:00</span>
        <span class=url>
            <span class=value-title title="http://www.it-academy.by/"> </span>
        </span>
    </div>
</div>
```

RDF (Resource Description Framework, среда описания ресурса)

```
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
    Мое имя <span property="v:name">Пётр Иванов</span>,
    но вы можете звать меня <span property="v:nickname">Петя</span>.
    Моя домашняя страница &mdash; <a href="http://www.ivanov.ru" rel="v:url">здесь</a>.
    Я живу в Минске и работаю <span property="v:title">преподавателем</span>
    в <span property="v:affiliation">ОЦ ПВТ</span>.
</div>
```

(8.45) HTML5 — формы

Новые теги

Тег **datalist** — возможность задать для текстового поля формы перечень значений, отображаемых в выпадающем списке. Возможность ввести вручную любое значение остается. Пример:

```
<datalist id=animalslist>
  <option value="собака">
  <option value="кошка">
  <option value="попугай">
</datalist>
Ведите вид животного: <input type=text list=animalslist>
```

Поддержка: Chrome 20+, Firefox, Opera

Новые типы элементов ввода

<input type=search> — работает как текстовое поле, но стилизуется под поисковые поля в операционной системе.

Поддержка: Chrome, Firefox, IE 10+

<input type=email>, <input type=url>, <input type=tel> — работают как текстовые поля, но браузер может более качественно предлагать ранее вводимые значения, проверять корректность введённых данных.

Поддержка: Chrome, Firefox, Opera 10.6+, IE 10+

<input type=range> — позволяет выбрать некоторое число из диапазона, например, через интерфейс «ползунок».

Поддержка: Chrome, Firefox 23+, Opera, IE 10+

<input type=number> — позволяет ввести число и управлять им (например, изменять введённое число на единицу в большую и меньшую сторону).

Поддержка: Chrome, Firefox 29+, Opera

<input type=date>, <input type=datetime>, <input type=datetime-local>, <input type=time>, <input type=month> — позволяют ввести различные компоненты даты и/или времени.

Поддержка: Chrome 20+, Opera

<input type=color> — позволяет выбрать цвет.

Поддержка: Chrome 20+, Firefox 29+, Opera 11+

Новые атрибуты элементов ввода

Атрибут **autofocus** — устанавливает фокус ввода на данный элемент формы при открытии веб-страницы.

Поддержка: Chrome, Firefox, Opera, IE 10+

Атрибут **spellcheck** для элементов ввода текста — включает или отключает автопроверку орфографии вводимого текста. Может иметь значение "true" либо "false".

Поддержка: Chrome, Firefox, Opera 10.5+, IE 10+

Атрибут **autocomplete** для элементов ввода текста — позволяет разрешить или запретить автозаполнение текста в элементе. Может иметь значения "on" (автозаполнение разрешено) либо "off" (автозаполнение запрещено).

Поддержка: Chrome 17+, Firefox, Opera, IE 9+

Атрибут **placeholder** для элементов ввода текста — задёт текст, который будет появляться в незаполненном элементе как подсказка о том, что именно нужно вводить в поле.

Поддержка: Chrome, Firefox, Opera 11.5+, IE 10+

Атрибут **multiple** для элемента выбора файла `<input type=file>` — позволяет выбирать сразу несколько файлов для отправки. В некоторых браузерах он также применим к другим элементам формы, например, для `<input type=email>` позволяет вводить несколько адресов e-mail.

Поддержка: Chrome, Firefox, Opera 10.6+, IE 10+

Псевдоклассы CSS, связанные с состоянием элементов форм

Псевдокласс **:enabled** — соответствует доступным элементам формы, т.е. элементам, у которых не указан атрибут `disabled`.

Псевдокласс **:disabled** — соответствует недоступным элементам формы, т.е. элементам, у которых указан атрибут `disabled`.

Псевдокласс **:checked** — соответствует установленным (выбранным) элементам формы (применимо к элементам `<input type=checkbox>` или `<input type=radio>`).

Поддержка: Chrome, Firefox, IE 9+



Валидация введённых значений

Атрибут **required** указывает, что поле обязательно должно быть заполнено.

Атрибуты **min** и **max** для элементов ввода чисел `<input type=number>` и дат `<input type=date>` задают соответственно минимальное и максимальное возможное значение, введённое в элемент.

Атрибут **step** для элементов ввода чисел `<input type=number>` задаёт шаг изменения для значения, введённого в элемент.

Атрибут **pattern** для элементов ввода текста задаёт регулярное выражение, которому должен соответствовать введённый в элемент текст.

Установленные атрибутами **min**, **max**, **step**, **pattern**, **required** ограничения, а также соответствие введённого значения типу элемента (`<input type=number/email/tel...>`) автоматически проверяются браузером с выдачей соответствующих предупреждений; неверно заполненная форма не отправляется на сервер.

Поддержка: Chrome, Firefox, IE 10+

Псевдоклассы CSS, связанные с валидацией

Псевдокласс **:required** — соответствует элементам формы с атрибутом `required`, т.е. обязательным для заполнения.

Псевдокласс **:optional** — соответствует элементам формы без атрибута `required`, т.е. необязательным для заполнения.

Псевдокласс **:valid** — соответствует элементам формы с допустимыми значениями.

Псевдокласс **:invalid** — соответствует элементам формы с недопустимыми значениями.

Псевдокласс **:in-range** — соответствует элементам ввода чисел, содержащим число в допустимом диапазоне.

Псевдокласс **:out-of-range** — соответствует элементам ввода чисел, содержащим число вне допустимого диапазона.

ВНИМАНИЕ: Валидация введённых в элементы форм значений обычно происходит только в момент отправки формы на сервер, соответственно псевдоклассы `:valid`, `:invalid`, `:in-range`, `:out-of-range` применяются к элементам формы только в этот момент.

Поддержка: Chrome, Firefox, IE 10+

Домашнее задание №8а

Переверстать проект *MAKET_FORM*. Форма должна быть адаптивной:

- на узком экране подписи полей должны быть выше полей, радиокнопки должны выстраиваться в столбик, текстовые поля должны растягиваться на всю доступную ширину;
- на широком экране подписи полей должны быть фиксированной ширины (т.е. ширина подразумеваемого «столбца» с подписями полей должна быть фиксированной), текстовые поля должны растягиваться на всю оставшуюся ширину окна.

Вёрстка блочная, резиновая. CSS-функцию *calc* (кто знает о ней) не использовать.

Макеты для широкого и для узкого окна приложены. Красная двойная рамка на макетах символизирует размер окна браузера (для ориентировки, её верстать не надо).

Количество и положение точек перехода определить самостоятельно.

Для публикации вашей статьи заполните, пожалуйста, следующую форму:

Название статьи:

Рубрика:
 программирование

Публикация:
 на сайте
 в журнале
 на сайте и в журнале

Разрешить комментарии:

Текст статьи:

Для публикации вашей статьи заполните, пожалуйста, следующую форму:

Название статьи:

Рубрика:
 программирование

Публикация:
 на сайте в журнале на сайте и в журнале

Разрешить комментарии:

Текст статьи:

Домашнее задание №8b

Создать проект MAKET_ABS. Добавить из библиотеки изображения biohazard.png, Man-and-woman.png, No_smoking.png, Yin_and_Yang.png.

Сверстать проект по макету.

При поднесении мыши к изображениям заднего плана (biohazard.png, Man-and-woman.png, Yin_and_Yang.png), они должны становиться ближе к пользователю чем изображение переднего плана (No_smoking.png), т.е. всплывать на передний план.

До изображений — произвольный многострочный текст на всю ширину окна браузера; изображения находятся посередине ширины окна браузера и примыкают к тексту снизу, т.е. когда при сужении окна браузера высота текста автоматически увеличивается, изображения должны смещаться вниз, оставаясь отцентрованными. Сразу после блока изображений — ещё один абзац произвольного многострочного текста.

Изображения должны быть размещены с использованием абсолютного позиционирования (*position: absolute*), координаты изображений должны быть заданы в пикселях.

Я к вам пишу — чего же боле? Что я могу еще сказать? Теперь, я знаю, в вашей воле
Меня презреньем наказать. Но вы, к моей несчастной доле Хоть каплю жалости храня,
Вы не оставите меня. Сначала я молчать хотела; Поверьте: моего стыда Вы не узнали
б никогда, Когда б надежду я имела Хоть редко, хоть в неделю раз В деревне нашей
видеть вас.



Чтоб только слышать ваши речи, Вам слово молвить, и потом Все думать, думать об одном И день и ночь до новой встречи.

Домашнее задание №8с

В проекте **MAKET_DIV** доверстать висячий блочок, т.е. остающийся на одном и том же месте окна браузера при прокрутке веб-страницы. Блок должен быть всегда на одном и том же расстоянии от правого верхнего угла окна браузера.

ПРИВЕТ УЧАСТИКАМ СОРЕВНОВАНИЙ!

[ВСТУПИТЬ В КЛУБ](#)

Виды спорта:

- [конькобежный спорт](#)
- [биатлон](#)
- [футбол](#)
- [стрельба из лука](#)
- [прыжки в длину](#)
- [прыжки в высоту](#)
- [фигурное катание](#)
- [синхронное плавание](#)
- [шахматы](#)
- [шашки](#)

ФУТБОЛ



Футбол — командный вид спорта, в котором целью является забить мяч в ворота соперника ногами или другими частями тела (кроме рук) большее количество раз, чем команда соперника.

Есть 17 официальных правил игры, каждое из которых содержит список оговорок и руководящих принципов. Эти правила предназначены для применения на всех уровнях футбола, хотя есть некоторые изменения для таких групп, как юниоры, взрослые, женщины и люди с ограниченными физическими возможностями. Законы очень часто формулировались в общих чертах, которые позволяют упростить их применения в зависимости от характера игры.

ГЛАВНОЕ НЕ ПОБЕДА, А УЧАСТИЕ!

IT-Academy.by

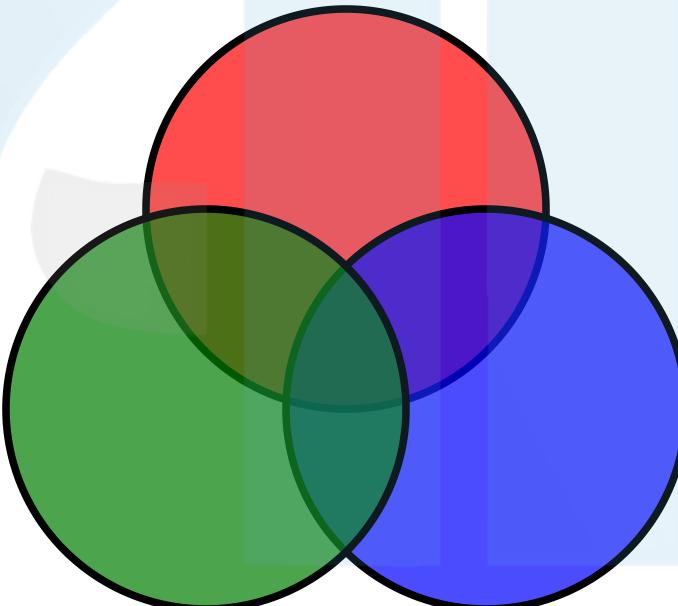
Тема 9. CSS3. Анимация. Flex. Grid layout

(9.15) SVG — пример

SVG (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики.

Предназначен для описания двумерной векторной или векторно/растровой графики в формате XML (а то есть на языках, родственных языкам HTML и CSS).

Поддерживает как неподвижную, так и анимированную интерактивную графику, т.е. элементами SVG-графики можно управлять средствами CSS или JavaScript.

HTML-код
<svg style="width: 500px; height: 350px"> <g style="fill-opacity:0.7;"> <circle cx="300" cy="120" r="100" style="fill:red; stroke:black; stroke-width:0.1cm"/> <circle cx="370" cy="220" r="100" style="fill:blue; stroke:black; stroke-width:0.1cm" /> <circle cx="230" cy="220" r="100" style="fill:green; stroke:black; stroke-width:0.1cm" /> </g> </svg>
Отображение в браузере


Поддержка: Chrome 27+, Firefox 29+, IE 9+, Safari 5.1+, iOS Safari 6.1+, Android Browser 4+

(9.25) SVG — краткое описание

HTML-код	Отображение в браузере
<pre><svg width="100" height="100" xmlns="http://www.w3.org/2000/svg"> <line x1="0" y1="0" x2="100" y2="100" stroke="red" stroke-width="2" /> <ellipse cx="50" cy="50" rx="40" ry="20" fill="yellow" stroke="blue" /> </svg></pre>	

специальные теги

тег, описание	основные атрибуты, стилевые свойства
<svg> парный тег, включает в себя все описания и построения рисунка	width, height — размеры рисунка на HTML-странице, в пикселях viewBox — четыре координаты (через пробел) в системе координат рисунка, задающие прямоугольник, который отображается на HTML-странице (он может охватывать не всю площадь построений); пропорции viewBox могут не совпадать с пропорциями атрибутов width и height; если не задан, предполагается что координатная система рисунка совпадает с сеткой пикселей HTML-страницы и рисунком является прямоугольник от точки 0,0 до точки width,height preserveAspectRatio — позволяет задать, как выравнивается и как масштабируется рисунок внутри прямоугольника width,height; по умолчанию рисунок растягивается, игнорируя пропорции

HTML-код	Отображение в браузере
<pre><svg width="90" height="90" xmlns="http://www.w3.org/2000/svg" viewBox="20 20 50 50"> <line x1="0" y1="0" x2="100" y2="100" stroke="red" stroke-width="2" /> <ellipse cx="50" cy="50" rx="40" ry="20" fill="yellow" stroke="blue" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<defs> парный тег, служит для различного рода описаний, которые затем могут быть использованы в рисунке; используется внутри тега <svg>	
<desc> парный тег, служит для текстового комментирования графики, которое может быть использовано альтернативными агентами (речевыми браузерами, поисковыми роботами и т.д.); содержимым тега должен быть текст; используется внутри любых SVG-тегов	

рисование геометрических фигур

тег, описание	основные атрибуты, стилевые свойства
<line> рисует отрезок прямой линии	x1,y1 — координаты начала линии x2,y2 — координаты конца линии stroke, stroke — цвет линии (как в CSS) stroke-width, stroke-width — толщина линии, по умолчанию 1; здесь и далее — в значении SVG-атрибутов размера единицы измерения не указываются, они всегда в пикселях ; в значениях стилевых свойств размера единицы измерения обязательны!
<polyline> рисует ломаную линию - незамкнутую последовательность отрезков	points — строка с парами координат узлов ломаной через пробел, каждая в виде x,y stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии
<polygon> рисует многоугольник - замкнутую (и возможно заливую) последовательность отрезков	points — строка с парами координат узлов ломаной stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии fill, fill — цвет заливки (как в CSS)

HTML-код	Отображение в браузере
<pre><svg width="120" height="157" xmlns="http://www.w3.org/2000/svg"> <line x1="0" y1="0" x2="100" y2="100" stroke="red" /> <line x1="20" y1="0" x2="120" y2="100" style="stroke:red; stroke-width:2" /> <polyline points="5,85 80,100 15,115" stroke="red" fill="none" /> <polygon points="5,125 80,140 15,155" stroke="red" fill="yellow" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<rect> рисует прямоугольник (возможно заливкой)	x,y — координаты левого верхнего угла прямоугольника width,height — ширина и высота прямоугольника rx,ry — радиус скругления углов прямоугольника stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии fill, fill — цвет заливки

HTML-код	Отображение в браузере
<pre><svg width="120" height="77" xmlns="http://www.w3.org/2000/svg"> <rect x="10" y="5" width="100" height="30" stroke="#FF0000" fill="yellow" /> <rect x="10" y="45" width="100" height="30" rx="10" ry="10" stroke="red" fill="yellow" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<circle> рисует окружность (возможно залитую)	cx, cy — координаты центра окружности r — радиус окружности stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии fill, fill — цвет заливки
<ellipse> рисует эллипс (возможно залитый)	cx, cy — координаты центра окружности rx, ry — радиусы эллипса stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии fill, fill — цвет заливки

HTML-код	Отображение в браузере
<pre><svg width="170" height="72" xmlns="http://www.w3.org/2000/svg"> <circle cx="35" cy="40" r="30" stroke="red" fill="yellow" /> <ellipse cx="120" cy="40" rx="45" ry="30" stroke="green" fill="#FFD0D0" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<path> рисует линию, состояющую из прямых и/или криволинейных участков (возможно замкнутую и залитую)	d — строка, содержащая команды построения участков линии, разделённые пробелами stroke, stroke — цвет линии stroke-width, stroke-width — толщина линии fill, fill — цвет заливки

В строке, описывающей участки линии, могут быть использованы следующие команды:

команда	аргументы	описание
M x y m dx dy	x, y — новые координаты пера относительно всего рисунка dx, dy — новые координаты пера относительно текущего положения пера	перемещает перо в указанные координаты
L x y l dx dy	x, y — координаты конца отрезка относительно всего рисунка dx, dy — координаты конца отрезка относительно текущего положения пера	рисует отрезок из предыдущего положения пера в указанные координаты (и перемещает перо в эти координаты)
H x h dx	x — координата x конца отрезка относительно всего рисунка dx — координата x конца отрезка относительно текущего положения пера	рисует горизонтальный отрезок из предыдущего положения пера в указанные координаты (и перемещает перо в эти координаты)
V y v dy	y — координата y конца отрезка относительно всего рисунка dy — координата y конца отрезка относительно текущего положения пера	рисует вертикальный отрезок из предыдущего положения пера в указанные координаты (и перемещает перо в эти координаты)
Z z		замыкает линию, рисуя отрезок в точку, откуда линия началась (и перемещает перо в эти координаты)

В строке, описывающей участки линии, могут быть использованы следующие команды (продолжение):

команда	аргументы	описание
A rx ry u s d x y a rx ry u s d dx dy	rx, ry — радиусы эллипса u — угол поворота эллипса s — выбор большой дуги (1) или малой (0) d — выбор дуги, идущей по часовой стрелке (1) или против (0) x, y — координаты конца дуги относительно всего рисунка dx, dy — координаты конца дуги относительно текущего положения пера	рисует дугу эллипса из предыдущего положения пера в указанные координаты (между двумя точками можно построить 2 эллипса с указанными радиусами и углом, и параметрами s и d выбирается, какая из четырёх образовавшихся дуг рисуется)

HTML-код	Отображение в браузере
<pre><svg width="140" height="60" xmlns="http://www.w3.org/2000/svg"> <path d="M 5 5 L 20 60 m 10 0 l 0 -60" stroke="red" fill="none" /> <path d="M 70 5 h 50 v 40 Z" stroke="black" fill="yellow" /> </svg></pre>	
<pre><svg width="160" height="130" xmlns="http://www.w3.org/2000/svg"> <path d="M 70 40 a 40 50 -20 0 0 20 50" stroke="blue" fill="none" /> <path d="M 70 40 a 40 50 -20 0 1 20 50" stroke="red" fill="none" /> <path d="M 70 40 a 40 50 -20 1 0 20 50" stroke="green" fill="none" /> <path d="M 70 40 a 40 50 -20 1 1 20 50" stroke="black" fill="none" /> </svg></pre>	

настройка заливок

тег, описание	основные атрибуты, стилевые свойства
для любых тегов, рисующих залитые фигуры	fill-opacity , fill-opacity — непрозрачность заливки, дробное число от 0 до 1 (по умолчанию 1)

HTML-код	Отображение в браузере
<pre><svg width="160" height="75" xmlns="http://www.w3.org/2000/svg"> <circle cx="35" cy="40" r="30" stroke="red" fill="yellow" /> <circle cx="105" cy="40" r="30" stroke="red" fill="yellow" /> <circle cx="70" cy="40" r="30" fill="blue" fill-opacity="0.5" /> </svg></pre>	

настройка линий

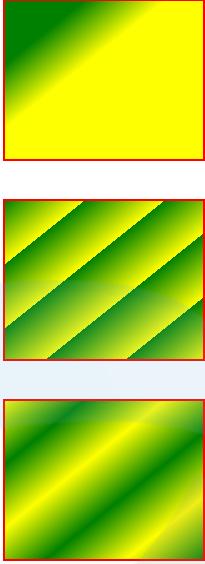
тег, описание	основные атрибуты, стилевые свойства
для любых тегов, рисующих линии	stroke-opacity , stroke-opacity — непрозрачность линии, дробное число от 0 до 1 (по умолчанию 1)
для любых тегов, рисующих линии	stroke-linecap , stroke-linecap — форма концов линии, одна из констант butt (торец, по умолчанию), round (круглое перо), square (квадратное перо)
для любых тегов, рисующих линии	stroke-linejoin , stroke-linejoin — форма углов линии, одна из констант miter (косые, по умолчанию), round (скруглённые), bevel (конические); также, атрибут и стилевое свойство stroke-miterlimit ограничивает выступ угла, образующегося при stroke-linejoin=miter
для любых тегов, рисующих линии	stroke-dasharray , stroke-dasharray — длины штрихов и интервалов между штрихами через запятую
для любых тегов, рисующих линии	stroke-dashoffset , stroke-dashoffset — отступ первого штриха (т.е. насколько короче должен быть первый штрих чем указано в stroke-dasharray)

HTML-код	Отображение в браузере
<pre><svg width="160" height="490" xmlns="http://www.w3.org/2000/svg"> <line x1="5" y1="20" x2="130" y2="70" stroke="red" stroke-width="5" /> <line x1="5" y1="30" x2="130" y2="80" stroke="red" stroke-width="5" stroke-opacity="0.5" /> <line x1="15" y1="60" x2="140" y2="110" stroke="green" stroke-width="20" stroke-linecap="butt" /> <line x1="15" y1="90" x2="140" y2="140" stroke="green" stroke-width="20" stroke-linecap="round" /> <line x1="15" y1="120" x2="140" y2="170" stroke="green" stroke-width="20" stroke-linecap="square" /> <line x1="15" y1="60" x2="140" y2="110" stroke="black" /> <line x1="15" y1="90" x2="140" y2="140" stroke="black" /> <line x1="15" y1="120" x2="140" y2="170" stroke="black" /> <polyline points="5,165 120,200 75,215" stroke="cyan" fill="none" stroke-width="20" stroke-linejoin="miter" /> <polyline points="5,225 120,260 75,275" stroke="cyan" fill="none" stroke-width="20" stroke-linejoin="round" /> <polyline points="5,285 120,320 75,335" stroke="cyan" fill="none" stroke-width="20" stroke-linejoin="bevel" /> <polyline points="5,165 120,200 75,215" stroke="black" fill="none" /> <polyline points="5,225 120,260 75,275" stroke="black" fill="none" /> <polyline points="5,285 120,320 75,335" stroke="black" fill="none" /> <polyline points="5,355 120,390 75,405" stroke="red" fill="none" stroke-dasharray="7,3,1,3" stroke-width="2" /> <polyline points="5,395 120,430 75,445" stroke="red" fill="none" stroke-dasharray="50,30" /> <polyline points="5,435 120,470 75,485" stroke="red" fill="none" stroke-dasharray="50,30" stroke-dashoffset="20" /> </svg></pre>	

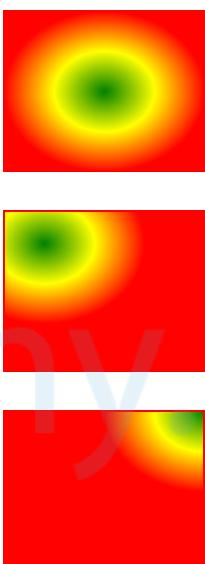
сложные заливки и обводки

тег, описание	основные атрибуты, стилевые свойства
<p><linearGradient> парный тег, описывает линейный градиент; содержимым тега должны быть теги <stop></p>	<p>id — идентификатор, по которому можно затем ссылаться на описанный градиент x1, y1, x2, y2 — координаты линии, вдоль которой строится градиент; за границами линии будет сплошной цвет; обычно задаются в процентах spreadMethod — указывает, как будет рисоваться градиент, если отрезок x1,y1-x2,y2 не закрывает всю площадь объекта:</p> <ul style="list-style-type: none"> • pad — первый и последний цвета градиента распространяются на всю оставшуюся площадь объекта • repeat — градиент повторяется многократно • reflect — градиент вместе с его зеркальным отражением повторяется многократно <p>gradientUnits — задаёт систему отсчёта для атрибутов x1, y1, x2, y2: objectBoundingBox (по умолчанию) — координаты задаются относительно объекта, к которому применён градиент, userSpaceOnUse — координаты задаются в пределах всего SVG-рисунка gradientTransform — трансформация, применяемая к градиенту (аналогично стилевому свойству transform в CSS3 и атрибуту transform в SVG)</p>
<p><stop> описывает точку изменения цвета (для линейного и радиального градиентов)</p>	<p>offset — позиция точки, от 0% до 100%</p> <p>stop-color, stop-color — цвет точки</p> <p>stop-opacity, stop-opacity — непрозрачность точки, число от 0 до 1</p>

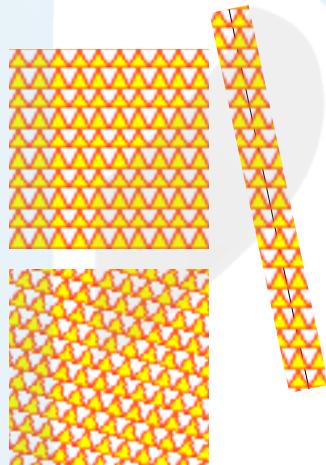
HTML-код	Отображение в браузере
<pre> <svg width="140" height="300" xmlns="http://www.w3.org/2000/svg"> <defs> <linearGradient id="LG1" x1="0%" y1="0%" x2="0%" y2="100%"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" stop-opacity="0.5" /> </linearGradient> <linearGradient id="LG2" x1="0%" y1="0%" x2="100%" y2="100%"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" stop-opacity="0.5" /> </linearGradient> <linearGradient id="LG3" gradientTransform="rotate(10)"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" stop-opacity="0.5" /> </linearGradient> </defs> <rect x="5" y="5" width="100" height="80" stroke="red" fill="url(#LG1)" /> <rect x="5" y="105" width="100" height="80" stroke="red" fill="url(#LG2)" /> <rect x="5" y="205" width="100" height="80" stroke="red" fill="url(#LG3)" /> </svg> </pre>	  

HTML-код	Отображение в браузере
<pre><svg width="140" height="300" xmlns="http://www.w3.org/2000/svg"> <defs> <linearGradient id="LG4" x1="20%" y1="20%" x2="40%" y2="40%"> <stop offset="0%" stop-color="green" /> <stop offset="100%" stop-color="yellow" /> </linearGradient> <linearGradient id="LG5" x1="20%" y1="20%" x2="40%" y2="40%" spreadMethod="repeat"> <stop offset="0%" stop-color="green" /> <stop offset="100%" stop-color="yellow" /> </linearGradient> <linearGradient id="LG6" x1="20%" y1="20%" x2="40%" y2="40%" spreadMethod="reflect"> <stop offset="0%" stop-color="green" /> <stop offset="100%" stop-color="yellow" /> </linearGradient> </defs> <rect x="5" y="5" width="100" height="80" stroke="red" fill="url(#LG4)" /> <rect x="5" y="105" width="100" height="80" stroke="red" fill="url(#LG5)" /> <rect x="5" y="205" width="100" height="80" stroke="red" fill="url(#LG6)" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<radialGradient> <i>парный тег, описывает радиальный градиент; содержимым тега должны быть теги <stop></i>	id — идентификатор, по которому можно затем ссылаться на описанный градиент fx, fy — координаты фокальной точки градиента cx, cy — координаты центра большого круга градиента (по умолчанию — центр объекта) r — радиус большого круга градиента (по умолчанию — размер объекта) gradientTransform — аналогично <linearGradient>

HTML-код	Отображение в браузере
<pre><svg width="140" height="300" xmlns="http://www.w3.org/2000/svg"> <defs> <radialGradient id="RG1"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" /> </radialGradient> <radialGradient id="RG2" cx="20%" cy="20%"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" /> </radialGradient> <radialGradient id="RG3" gradientTransform="translate(0.5 -0.5)"> <stop offset="0%" stop-color="green" /> <stop offset="50%" stop-color="yellow" /> <stop offset="100%" stop-color="red" /> </radialGradient> </defs> <rect x="5" y="5" width="100" height="80" stroke="red" fill="url(#RG1)" /> <rect x="5" y="105" width="100" height="80" stroke="red" fill="url(#RG2)" /> <rect x="5" y="205" width="100" height="80" stroke="red" fill="url(#RG3)" /> </svg></pre>	

тег, описание	основные атрибуты, стилевые свойства
<pattern> <i>парный тег, описывает элемент узора для тегов, рисующих линии или заливки; содержимым тега должны быть построения, рисующие один элемент узора</i>	id — идентификатор, по которому можно затем ссылаться на описанную форму конца линии width, height — размеры элемента узора patternTransform — трансформация, применяемая к градиенту (аналогично стилевому свойству transform в CSS3 и атрибуту transform в SVG) patternUnits — задаёт, в каких единицах указана ширина и высота узора, обычно указывается значение userSpaceOnUse — ширина и высота заданы в системе отсчёта того элемента, к которому применён узор

HTML-код	Отображение в браузере
<pre> <svg width="180" height="240" xmlns="http://www.w3.org/2000/svg"> <defs> <pattern id="TRP1" width="10" height="10" patternUnits="userSpaceOnUse"> <polygon points="5,0 10,10 10,0 10" stroke="red" fill="yellow" /> </pattern> <pattern id="TRP2" width="10" height="10" patternUnits="userSpaceOnUse" patternTransform="rotate(15)"> <polygon points="5,0 10,10 10,0 10" stroke="red" fill="yellow" /> </pattern> </defs> <rect x="10" y="30" width="100" height="100" fill="url(#TRP1)" /> <rect x="10" y="140" width="100" height="100" fill="url(#TRP2)" /> <line x1="120" y1="10" x2="160" y2="200" stroke-width="20" stroke="url(#TRP1)" /> </svg> </pre>	

IT-Academy.by

работа с текстом

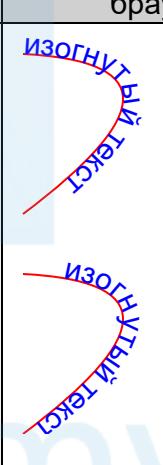
тег, описание	основные атрибуты, стилевые свойства
<p><text> парный тег, выводит строку текста; должен содержать текст либо тег <textPath>, может содержать теги <tspan></p>	<p>x, y — координаты текста font, font-family, font-size, font-style, font-weight, letter-spacing, word-spacing, text-decoration и другие стилевые свойства и одноимённые атрибуты задают различные свойства шрифта (аналогично CSS) textLength — ширина текста (если не задана, будет определена автоматически; если задана будут увеличены интервалы между буквами и, возможно, ширина букв, чтобы довести ширину текста до указанной) lengthAdjust — задаёт, что следует увеличивать, чтобы получить установленную через textLength ширину текста:</p> <ul style="list-style-type: none"> • spacing (по умолчанию) — только интервалы между буквами • spacingAndGlyphs — интервалы между буквами и ширину букв <p>rotate — задаёт углы поворота для каждого символа строки через запятую; если символов больше чем задано углов, последний угол действует на все символы до конца строки stroke, stroke — цвет линии (обводки букв) fill, fill — цвет заливки букв text-anchor, text-anchor — выравнивание текста относительно координат, заданных атрибутами x,y:</p> <ul style="list-style-type: none"> • start (по умолчанию) — x,y задают координаты левого конца базовой линии текста • middle (по умолчанию) — x,y задают координаты середины базовой линии текста • end (по умолчанию) — x,y задают координаты правого края базовой линии текста

HTML-код	Отображение в браузере
<pre><svg width="193" height="250" xmlns="http://www.w3.org/2000/svg"> <text x="0" y="10">строка текста</text> <text x="0" y="30" fill="red">цветная строка текста</text> <text x="0" y="80" fill="none" stroke="green" style="font-size: 30px"> обводка</text> <text x="0" y="115" textLength="150">строка текста</text> <text x="0" y="130" textLength="150" lengthAdjust="spacingAndGlyphs">строка текста</text> <text x="0" y="190" fill="green" font-size="40" rotate="20,0,-10"> строчка</text> <text x="100" y="210" text-anchor="start">левым краем</text> <text x="100" y="225" text-anchor="middle">центром</text> <text x="100" y="240" text-anchor="end">правым краем</text> </svg></pre>	<p>строка текста цветная строка текста</p> <p>обводка</p> <p>строка текста строка текста</p> <p>строчка</p> <p>левым краем центром правым краем</p>

тег, описание	основные атрибуты, стилевые свойства
<tspan> парный тег, задаёт подстроку текста, которую можно позиционировать и/или стилизовать отдельно от остального текста; должен содержать текст	dx , dy — относительные координаты подстроки текста font , font-family , font-size , font-style , font-weight , letter-spacing , word-spacing , text-decoration и другие стилевые свойства и одноимённые атрибуты задают различные свойства шрифта (аналогично CSS) stroke , stroke — цвет линии (обводки букв) fill , fill — цвет заливки букв

HTML-код	Отображение в браузере
<pre> <svg width="200" height="100" xmlns="http://www.w3.org/2000/svg"> <text x="5" y="30" fill="red"> часть1 <tspan dy="5" fill="green">часть2</tspan> <tspan dy="-20" dx="20" fill="blue" font-size="7">часть3</tspan> </text> </svg> </pre>	 <p>часть1 часть2 часть3</p>

тег, описание	основные атрибуты, стилевые свойства
<textPath> парный тег, задаёт кривую, вдоль которой нужно расположить текст; должен содержать текст	xlink:href — URI описанной ранее кривой; обычно кривая описана в этом же документе, тогда URI можно записать в виде "#XXX", где XXX — идентификатор кривой startOffset — начальное смещение текста вдоль кривой

HTML-код	Отображение в браузере
<pre> <svg width="150" height="200" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"> <path id="PPP1" d="M 5 10 q 100 5, 0 80" stroke="red" fill="none" /> <text fill="blue"> <textPath xlink:href="#PPP1">изогнутый текст</textPath> </text> <path id="PPP2" d="M 5 120 q 100 5, 0 80" stroke="red" fill="none" /> <text fill="blue"> <textPath xlink:href="#PPP2" startOffset="20"> изогнутый текст</textPath> </text> </text> </svg> </pre>	 <p>изогнутый текст изогнутый текст</p>

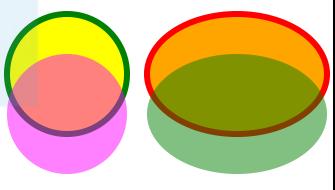
растровые изображения

тег, описание	основные атрибуты, стилевые свойства
<image> вставляет растровое изображение (аналогично тегу в HTML)	x, y — координаты изображения width, height — размеры прямоугольника, в который должно быть вписано изображение (пропорции изображения не меняются); эти атрибуты указываются в единицах SVG (как и все координаты и размеры в SVG), а не в пикселях xlink:href — URI изображения (т.к. атрибут принадлежит пространству имён xlink, следует в теге <SVG> указывать URI этого пространства имён атрибутом xmlns:xlink="http://www.w3.org/1999/xlink")

HTML-код	Отображение в браузере
<pre><svg width="200" height="80" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"> <rect x="0" y="5" width="150" height="50" fill="none" stroke="red" /> <image x="0" y="5" width="150" height="50" xlink:href="images/Hilu200.jpg" /> </svg></pre>	

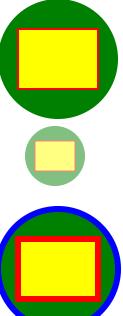
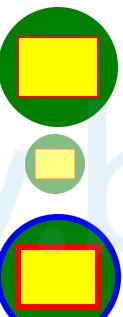
группировка

тег, описание	основные атрибуты, стилевые свойства
<g> парный тег; позволяет сгруппировать некоторые построения, для применения к ним общих атрибутов, стилевых свойств, трансформаций и т.д.	

HTML-код	Отображение в браузере
<pre><svg width="170" height="100" xmlns="http://www.w3.org/2000/svg"> <g stroke="green" stroke-width="3"> <circle cx="35" cy="40" r="30" fill="yellow" /> <ellipse cx="120" cy="40" rx="45" ry="30" stroke="red" fill="orange" /> </g> <g transform="translate(0,20)" opacity="0.5" fill="green"> <circle cx="35" cy="40" r="30" fill="magenta" /> <ellipse cx="120" cy="40" rx="45" ry="30" /> </g> </svg></pre>	

работа с библиотекой символов

тег, описание	основные атрибуты, стилевые свойства
<symbol> парный тег; описывает (неотображаемые) построения, которые затем можно отобразить многократно в рисунке (тегом <use>) с возможностью стилизации «по месту»; внутри тега <symbol> — своя система координат	id — идентификатор, по которому можно затем ссылаться на символ
<use> отображает описанные ранее построения; построения могут быть описаны тегом <symbol> или любым тегом, обычно внутри тега <defs>	x, y — координаты, в которых отображаются построение xlink:href — URI построений, обычно построения описаны в этом же документе, тогда URI можно записать в виде "#XXX", где XXX — идентификатор построений (например тега <symbol>); если построения описаны во внешнем SVG-файле, URI может быть в виде "путь/имя.svg#XXX", причём этот внешний SVG-файл в целях безопасности должен иметь одно происхождение с документом (т.е. один домен, протокол и порт)

HTML-код	Отображение в браузере
<pre> <svg width="120" height="180" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"> <symbol id="BRICK1"> <circle cx="33" cy="33" r="30" fill="green" /> <rect x="13" y="18" width="40" height="30" stroke="red" fill="yellow" /> </symbol> <use xlink:href="#BRICK1" x="20" y="10" /> <use xlink:href="#BRICK1" opacity="0.5" transform="scale(0.5)" x="70" y="150" /> <use xlink:href="#BRICK1" stroke="blue" stroke-width="3" x="20" y="115" /> </svg> </pre>	 <p>The first instance is a yellow rectangle with a red border, centered at (20, 10). The second instance is a smaller yellow rectangle with a red border, scaled by 0.5 and positioned at (70, 150). The third instance is a yellow rectangle with a red border, scaled by 0.5 and positioned at (20, 115), with a blue stroke and a stroke width of 3.</p>
<pre> <svg width="120" height="190" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"> <defs> <g id="BRICK2"> <circle cx="33" cy="33" r="30" fill="green" /> <rect x="13" y="18" width="40" height="30" stroke="red" fill="yellow" /> </g> </defs> <use xlink:href="#BRICK2" x="20" y="10" /> <use xlink:href="#BRICK2" opacity="0.5" transform="scale(0.5)" x="70" y="150" /> <use xlink:href="#BRICK2" stroke="blue" stroke-width="3" x="20" y="115" /> </svg> </pre>	 <p>The first instance is a yellow rectangle with a red border, centered at (20, 10). The second instance is a smaller yellow rectangle with a red border, scaled by 0.5 and positioned at (70, 150). The third instance is a yellow rectangle with a red border, scaled by 0.5 and positioned at (20, 115), with a blue stroke and a stroke width of 3.</p>

преобразования и эффекты

тег, описание	основные атрибуты, стилевые свойства
для любых тегов, рисующих что-либо	<p>transform, transform — преобразование (аналогично стилевому свойству transform в CSS3), значение представляет собой список операций, разделённых запятыми или пробелами, в списке могут быть указаны следующие операции:</p> <p>translate(x y) — перенос построений на x вправо и на y вниз</p> <p>scale(s) или scale(sx sy) — увеличение построений в s раз (во втором варианте указываются отдельные коэффициенты для x и для y)</p> <p>rotate(a) или rotate(a x y) — поворот построений на угол a (в градусах, по часовой стрелке) относительно начала координат (во втором варианте координаты x и y указывают, относительно какой точки должен быть осуществлён поворот)</p> <p>skewX(a) и skewY(a) — наклон построений на угол a по оси x и по оси y соответственно</p> <p>здесь и далее — в значении SVG-атрибутов угла единицы измерения не указываются, они всегда в градусах:</p> <p style="color: blue;">transform="rotate(45)"</p> <p>в значениях стилевых свойств единицы измерения обязательны:</p> <p style="color: green;">style="transform: rotate(45deg);"</p> <p>transform-origin работает, однако, кроссбраузерно может быть задано только в пикселях (указание без единиц размера, указание в процентах, указание ключевых слов вроде center некроссбраузерно); указание же в пикселях задаёт точку применения трансформации относительно всего тела <svg>, а не относительно трансформируемого элемента</p>

HTML-код	Отображение в браузере
<pre> <svg width="180" height="240" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"> <symbol id="STOP"> <circle cx="33" cy="33" r="30" /> <rect x="13" y="18" width="40" height="30" stroke="red" fill="yellow" /> </symbol> <use xlink:href="#STOP" x="0" y="0" fill="green" /> <use xlink:href="#STOP" x="0" y="0" fill="red" transform="translate(70 0)" /> <use xlink:href="#STOP" x="0" y="0" fill="blue" transform="translate(0 70) scale(0.8 0.5)" /> <use xlink:href="#STOP" x="0" y="0" fill="gray" transform="translate(70 70) rotate(10)" /> <use xlink:href="#STOP" x="0" y="0" fill="magenta" transform="translate(0 140) skewX(10)" /> <use xlink:href="#STOP" x="0" y="0" fill="cyan" transform="translate(70 140) skewY(10)" /> </svg></pre>	

(9.55) Модель flex-блоков

Модель flex-блоков — относительно новая концепция, позволяющая при блочной вёрстке управлять положением и размером блоков макета гибче и легче, чем при использовании стилевого свойства `float` и прочих ухищрений.

Стилевое свойство контейнера **display** со значением **flex** превращает его во flex-контейнер, т.е. контейнер, изменяющий положение и размеры своих элементов (flex-блоков) flex-способом.

Стилевое свойство flex-блока **flex-basis** задаёт стартовый размер flex-блока, т.е. размер flex-блока до того, как его размер будет скорректирован в зависимости от доступного размера flex-контейнера. По умолчанию **flex-basis** установлен в `auto`, а то есть, никакого стартового размера нет.

Стилевое свойство flex-блока **flex-grow** с числовым значением задаёт скорость, с какой размер блока **увеличивается** при распределении доступного размера flex-контейнера между всеми flex-блоками.

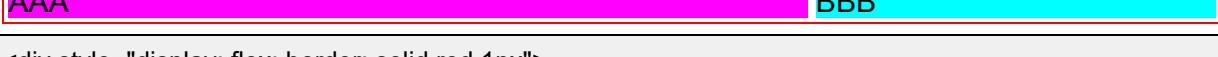
Стилевое свойство flex-блока **flex-shrink** с числовым значением задаёт скорость, с которой размер блока **уменьшается** при распределении доступного размера flex-контейнера между всеми flex-блоками.

Скорость увеличения или уменьшения размера flex-блока — относительная величина, т.е. неважно какое именно число задать, важно лишь во сколько раз это число

больше или меньше аналогичных чисел у других flex-элементов.

По умолчанию, и `flex-grow` и `flex-shrink` равны 0, а то есть, размер flex-блока не изменяется вообще, независимо от размера flex-контейнера.

HTML-код
<div style="display: flex; border: solid red 1px"> <div style="flex-basis: 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex-basis: 100px; margin: 2px; background-color: cyan">BBB</div> </div>
Отображение в браузере

<div style="display: flex; border: solid red 1px"> <div style="flex-basis: 100px; flex-grow: 3; margin: 2px; background-color: magenta">AAA</div> <div style="flex-basis: 100px; flex-grow: 1; margin: 2px; background-color: cyan">BBB</div> </div>
Отображение в браузере

<div style="display: flex; border: solid red 1px"> <div style="flex-grow: 3; margin: 2px; background-color: magenta">AAA</div> <div style="flex-grow: 1; margin: 2px; background-color: cyan">BBB</div> </div>
Отображение в браузере


Стилевое свойство flex-блока **flex** позволяет одновременно задать скорость увеличения, скорость уменьшения и стартовый размер flex-блока.

HTML-код

```
<div style="display: flex; border: solid red 1px">
  <div style="flex: 3 1 100px; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex: 1 1 100px; margin: 2px; background-color: cyan">BBB</div>
</div>
```

Отображение в браузере



AAA BBB

Стилевое свойство flex-блока **order** позволяет задать порядок, в котором flex-блоки «укладываются» внутри flex-контейнера.

HTML-код

```
<div style="display: flex; border: solid red 1px">
  <div style="flex-grow: 1; order: 7; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 1; order: 5; margin: 2px; background-color: cyan">BBB</div>
  <div style="flex-grow: 1; order: 1; margin: 2px; background-color: lightgreen">CCC</div>
  <div style="flex-grow: 1; order: 4; margin: 2px; background-color: yellow">DDD</div>
</div>
```

Отображение в браузере



CCC DDD BBB AAA

Стилевое свойство flex-контейнера **flex-direction** задаёт направление, в котором flex-блоки «укладываются» в него.

Возможные значения:

flex-direction: row — основная ось идёт слева направо, и соответственно между flex-блоками распределяется **ширина** flex-контейнера. Это значение по умолчанию.

flex-direction: row-reverse — основная ось идёт справа налево.

flex-direction: column — основная ось идёт сверху вниз, и соответственно между flex-блоками распределяется **высота** flex-контейнера.

flex-direction: column-reverse — основная ось идёт снизу вверх.

HTML-код

```
<div style="display: flex; flex-direction: column; border: solid red 1px; height: 100px">
  <div style="flex-grow: 2; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 1; margin: 2px; background-color: cyan">BBB</div>
</div>
```

Отображение в браузере



AAA BBB

Стилевое свойство **justify-content** задаёт, куда распределяется оставшееся во flex-контейнере свободное место вдоль основной оси. Свободное место во flex-контейнере может появиться, если не весь размер распределён на flex-блоки (т.е. если все flex-блоки нерастяжимые или если они достигли своего max-width/max-height).

Возможные значения:

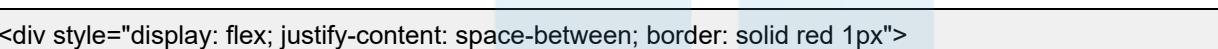
justify-content: flex-start — flex-блоки прижимаются к началу основной оси (например, к левому либо к верхнему краю, в зависимости от flex-direction). Это значение по умолчанию.

justify-content: flex-end — flex-блоки прижимаются к концу основной оси.

justify-content: center — flex-блоки центрируются на основной оси.

justify-content: space-between — в промежутки между flex-блоками вдоль основной оси поровну добавляется свободное место.

justify-content: space-around — в промежутки между flex-блоками, а также до них и после них вдоль основной оси добавляется свободное место вдоль основной оси.

HTML-код	Отображение в браузере
<pre><div style="display: flex; justify-content: flex-start; border: solid red 1px"> <div style="flex: 0 0 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex: 0 0 100px; margin: 2px; background-color: cyan">BBB</div> </div></pre>	
<pre><div style="display: flex; justify-content: flex-end; border: solid red 1px"> <div style="flex: 0 0 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex: 0 0 100px; margin: 2px; background-color: cyan">BBB</div> </div></pre>	
<pre><div style="display: flex; justify-content: center; border: solid red 1px"> <div style="flex: 0 0 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex: 0 0 100px; margin: 2px; background-color: cyan">BBB</div> </div></pre>	
<pre><div style="display: flex; justify-content: space-between; border: solid red 1px"> <div style="flex: 0 0 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex: 0 0 100px; margin: 2px; background-color: cyan">BBB</div> <div style="flex: 0 0 100px; margin: 2px; background-color: lightgreen">CCC</div> </div></pre>	
<pre><div style="display: flex; justify-content: space-around; border: solid red 1px"> <div style="flex: 0 0 100px; margin: 2px; background-color: magenta">AAA</div> <div style="flex: 0 0 100px; margin: 2px; background-color: cyan">BBB</div> <div style="flex: 0 0 100px; margin: 2px; background-color: lightgreen">CCC</div> </div></pre>	

Стилевое свойство **align-items** задаёт, как располагаются flex-блоки вдоль побочной оси (оси, повернутой на 90 градусов по часовой стрелке к основной оси).

Некоторые возможные значения:

align-items: flex-start — flex-блоки прижимаются к началу побочной оси.

align-items: flex-end — flex-блоки прижимаются к концу побочной оси.

align-items: center — flex-блоки центрируются на побочной оси.

align-items: stretch — flex-блоки растягиваются вдоль побочной оси на весь размер flex-контейнера (однако, их max-width/max-height учитываются). Это значение по умолчанию.

HTML-код

```
<div style="display: flex; align-items: flex-start; height: 45px; border: solid red 1px">
  <div style="flex-grow: 1; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 2; margin: 2px; background-color: cyan">BBB<br>BBB</div>
  <div style="flex-grow: 1; margin: 2px; background-color: lightgreen">CCC</div>
</div>
```

Отображение в браузере



```
<div style="display: flex; align-items: flex-end; height: 45px; border: solid red 1px">
  <div style="flex-grow: 1; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 2; margin: 2px; background-color: cyan">BBB<br>BBB</div>
  <div style="flex-grow: 1; margin: 2px; background-color: lightgreen">CCC</div>
</div>
```

Отображение в браузере



```
<div style="display: flex; align-items: center; height: 45px; border: solid red 1px">
  <div style="flex-grow: 1; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 2; margin: 2px; background-color: cyan">BBB<br>BBB</div>
  <div style="flex-grow: 1; margin: 2px; background-color: lightgreen">CCC</div>
</div>
```

Отображение в браузере



```
<div style="display: flex; height: 45px; border: solid red 1px">
  <div style="flex-grow: 1; margin: 2px; background-color: magenta">AAA</div>
  <div style="flex-grow: 2; margin: 2px; background-color: cyan">BBB<br>BBB</div>
  <div style="flex-grow: 1; margin: 2px; background-color: lightgreen">CCC</div>
</div>
```

Отображение в браузере



Синтаксис (названия и значения стилевых свойств) изменялись и ещё могут изменяться.

Вышеуказанный синтаксис поддерживают не все браузеры:

Chrome 31+, Firefox 29+, Opera 22+, IE 11+,

Safari 6.1+ (-webkit-), iOS Safari 7.1+ (-webkit-), Android Browser 4.4+

(9.57) Grid-раскладка

Grid-раскладка — новая концепция, ещё более гибкая чем flex, позволяющая при блочной вёрстке управлять положением и размером блоков макета ещё гибче и легче.

Стилевое свойство контейнера **display** со значением **grid** превращает его в **grid-контейнер**, т.е. контейнер, изменяющий положение и размеры своих элементов (grid-элементов) по своим правилам.

Grid-контейнер напоминает сетку (таблицу), т.е. состоит из **grid-строк**, **grid-столбцов** и **grid-ячеек**. Grid-строки и grid-столбцы называются **треками**. В grid-ячейках или группах grid-ячеек можно размещать произвольное содержимое — будем называть такое содержимое **grid-элементами**. Grid-элементами автоматически становятся все потомки (дочерние теги) grid-контейнера.

Стилевые свойства grid-контейнера **grid-template-columns** и **grid-template-rows** позволяют задать, сколько grid-строк и grid-столбцов в нём будет и каковы их размеры.

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 100px 150px 100px; grid-template-rows: 30px 20px"> <div style="background-color: magenta">AAA</div> <div style="background-color: cyan">BBB</div> <div style="background-color: silver">CCC</div> <div style="background-color: yellow">DDD</div> <div style="background-color: red">EEE</div> <div style="background-color: green">FFF</div> </div></pre>
Отображение в браузере


На этом примере видно, что grid-элементы (потомки grid-контейнера) автоматически распределились по grid-ячейкам слева направо и сверху вниз. Другие способы автоматического размещения можно задать стилем свойством **grid-auto-flow**, но мы это рассматривать не будем, т.к. основной вариант размещения grid-элементов — это как раз прямое указание grid-ячеек, в которых их нужно разместить (рассмотрим это ниже).

Также на этом примере видно, что grid-контейнер не занял всю доступную ему ширину, т.к. мы указали точные ширины колонок, в сумме меньшие, чем ширина страницы.

Положение grid-контейнера внутри доступного ему пространства можно задавать стилевыми свойствами **justify-content** и **align-content** grid-контейнера.

Для указания размеров треков (grid-колонок и grid-строк) можно использовать любые привычные единицы размеров (например, проценты — 50% это будет половина доступной ширины).

Но есть и ещё одна, новая и очень важная единица размера — **fr**:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 1fr 100px 2fr; grid-template-rows: 30px 20px"> <div style="background-color: magenta">AAA</div> <div style="background-color: cyan">BBB</div> <div style="background-color: silver">CCC</div> <div style="background-color: yellow">DDD</div> <div style="background-color: red">EEE</div> <div style="background-color: green">FFF</div> </div></pre>
Отображение в браузере


Смысл таков — вторая grid-колонка должна быть шириной 100px, а **всё оставшееся место** нужно поделить между первой и третьей grid-колонками, причём третьей grid-колонке должно достаться вдвое больше чем первой.

Аналогично управляются размеры grid-строк.

Если grid-колонок или grid-строк много, чтобы не повторять их размеры, можно воспользоваться синтаксисом `repeat`:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 100px repeat(2, 50px 1fr); grid-template-rows: 30px 20px"> <div style="background-color: magenta">AAA</div> <div style="background-color: cyan">BBB</div> <div style="background-color: silver">CCC</div> <div style="background-color: yellow">DDD</div> <div style="background-color: red">EEE</div> <div style="background-color: green">FFF</div> <div style="background-color: gray">GGG</div> </div></pre>
Отображение в браузере


Запись `repeat(2, 50px 1fr)` означает — два раза повторить комбинацию ширин grid-колонок 50px и 1fr, т.е. запись `grid-template-columns: 100px repeat(2, 50px 1fr)` аналогична записи `grid-template-columns: 100px 50px 1fr 50px 1fr`.

На этом примере также видно, что если grid-элементов меньше чем grid-ячеек, остаются незаполненные grid-ячейки.

Ширину grid-колонки или высоту grid-строки можно также задать в виде **minmax(100px, 200px)** — это значит минимум 100px, максимум 200px. Например, запись `minmax(100px, auto)` означает — размер должен быть не менее 100px и не более размера содержимого grid-ячеек.

Можно также попросить браузер автоматически заполнить всё имеющееся пространство треками (колонками или строками) с заданным размером:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px repeat(auto-fill, 50px 100px); grid-template-rows: 30px 20px"> <div style="background-color: magenta">AAA</div> <div style="background-color: cyan">BBB</div> <div style="background-color: silver">CCC</div> <div style="background-color: yellow">DDD</div> <div style="background-color: red">EEE</div> <div style="background-color: green">FFF</div> </div></pre>

Отображение в браузере

AAA	BBB	CCC	DDD	EEE	
FFF					

Первая grid-колонка здесь имеет ширину 200px, а затем браузеру удалось в оставшуюся ширину дважды разместить пары колонок шириной 50px и 100px.

Для любого из grid-элементов мы можем стилевыми свойствами **grid-column** и **grid-row** явно указать, в какую из grid-ячеек его надо разместить.

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px"> <div style="background-color: cyan; grid-column: 2; grid-row: 1">AAA</div> <div style="background-color: silver; grid-column: 1; grid-row: 2">BBB</div> <div style="background-color: magenta; grid-column: 3; grid-row: 2">CCC</div> </div></pre>

Отображение в браузере

AAA		
BBB		CCC

Grid-элемент может занимать и несколько grid-ячеек сразу.

Между grid-ячейками есть (воображаемые) **grid-линии**. Например, если в grid-контейнере 2 grid-колонки, то слева, справа и между ними есть 3 вертикальные grid-линии. Если в grid-контейнере 3 grid-строки, то выше, ниже и между ними есть 4 горизонтальные grid-линии.

Прямоугольные области из нескольких grid-ячеек называются **grid-области**.

Для любого из grid-элементов мы можем явно указать, в какую из grid-областей его надо разместить. Grid-область задаётся grid-линиями, которые её ограничивают.

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px"> <div style="background-color: magenta; grid-column: 2 / 4; grid-row: 1">AAA</div> <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4">BBB</div> <div style="background-color: yellow; grid-column: 2; grid-row: 2">CCC</div> </div></pre>
Отображение в браузере

The screenshot shows a grid container with a red border. Inside, there are three columns and three rows. Column 1 is filled with the text 'BBB' and has a silver background. Column 2 is filled with the text 'AAA' and has a magenta background. Column 3 is filled with the text 'CCC' and has a yellow background. The first row (magenta) covers both columns 2 and 3. The second row (yellow) covers both columns 1 and 2. The third row (white) covers only column 1.

Здесь элемент AAA занимает по горизонтали между 2-й и 4-й вертикальными grid-линиями, элемент В занимает по вертикали место между 2-й и 4-й горизонтальными grid-линиями.

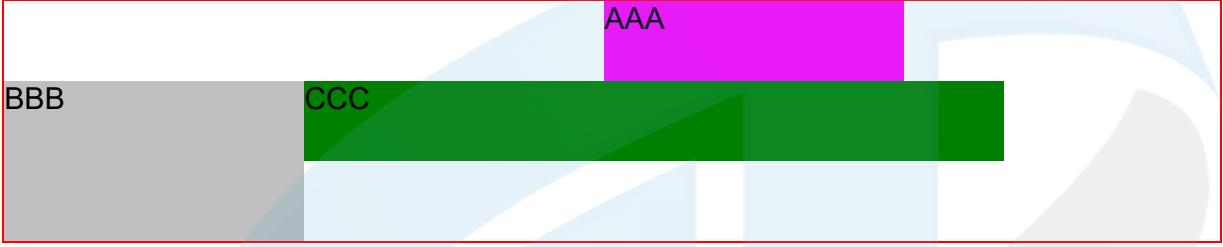
Добиться, чтобы grid-элемент занимал сразу несколько grid-ячеек, можно также ключевым словом `span`:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px"> <div style="background-color: magenta; grid-column: 2; grid-row: 1">AAA</div> <div style="background-color: silver; grid-column: 3; grid-row: 1 / span 2">BBB</div> <div style="background-color: yellow; grid-column: 1 / span 2; grid-row: 2 / span 2">CCC</div> </div></pre>
Отображение в браузере

The screenshot shows a grid container with a red border. Inside, there are three columns and three rows. Column 1 (yellow) contains element CCC. Column 2 (magenta) contains element AAA. Column 3 (grey) contains element BBB. The first row (magenta) spans both columns 2 and 3. The second row (yellow) spans both columns 1 and 2. The third row (grey) spans column 1.

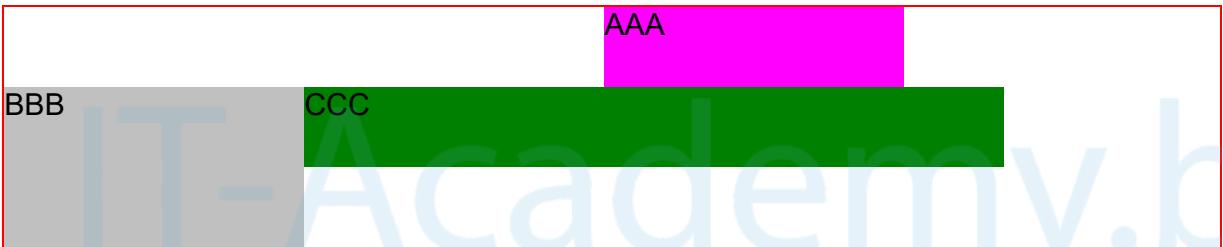
Т.е. элемент BBB начинается с 1-й горизонтальной grid-линии и занимает по высоте 2 grid-строки, элемент CCC расположен ниже 2-й горизонтальной grid-линии и правее 1-й вертикальной grid-линии и занимает по ширине две grid-колонки и по высоте две grid-строки.

Можно, указывая положение и/или размер grid-элемента, «вылезти» за пределы grid-контейнера. При этом автоматически создаются т.н. **неявные** grid-колонки и grid-строки, их размер можно задать стилевыми свойствами **grid-auto-columns** и **grid-auto-rows**.

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 150px 150px 150px; grid-template-rows: 40px 40px 40px; grid-auto-columns: 50px"> <div style="background-color: magenta; grid-column: 3; grid-row: 1">AAA</div> <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4">BBB</div> <div style="background-color: green; grid-column: 2 / span 3; grid-row: 2">CCC</div> </div></pre>
Отображение в браузере


Здесь в гриде задано 3 колонки с точно заданной шириной 150px, и элемент CCC начинается со 2-й вертикальной grid-линии и занимает ширину 3 колонок, хотя доступно только две. Браузер автоматически создаёт для размещения элемента CCC ещё одну колонку шириной 50px.

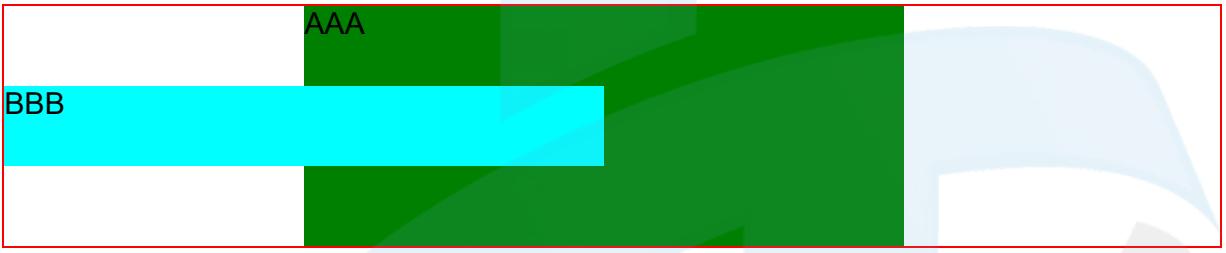
Не всегда удобно оперировать номерами grid-колонок. Им можно задавать символьические имена, прямо в стилевых свойствах **grid-template-columns** и **grid-template-rows**:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 150px [center] 150px 150px; grid-template-rows: 40px 40px 40px [bottom]; grid-auto-columns: 50px"> <div style="background-color: magenta; grid-column: 3; grid-row: 1">AAA</div> <div style="background-color: silver; grid-column: 1; grid-row: 2 / bottom">BBB</div> <div style="background-color: green; grid-column: center / span 3; grid-row: 2">CCC</div> </div></pre>
Отображение в браузере


Здесь 2-й вертикальной grid-линии (т.е. расположенной между 1-й и 2-й grid-колонками) дано имя **center**, а 4-й горизонтальной grid-линии (т.е. расположенной после 3-й grid-строки) дано имя **bottom**.

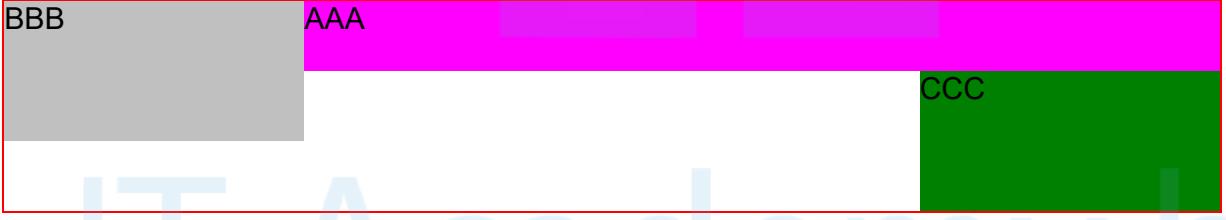
Очень интересная возможность — несколько grid-элементов могут занимать одну grid-ячейку или занимать частично пересекающиеся grid-области:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 150px 150px 150px; grid-template-rows: 40px 40px 40px"> <div style="background-color: green; grid-column: 2 / span 2; grid-row: 1 / span 3">AAA</div> <div style="background-color: cyan; grid-column: 1 / span 2; grid-row: 2">BBB</div> </div></pre>
Отображение в браузере



Который из grid-элементов окажется «выше» при таком расположении — определяется стилевым свойством **z-index**.

Можно при описании grid-контейнера в стилевом свойстве **grid-template-areas** задать именованные grid-области и потом размещать в них grid-элементы, указывая только имя grid-области в стилевом свойстве **grid-area**:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 150px 1fr 150px; grid-template-rows: 35px 35px 35px; grid-template-areas: 'menu header header' 'menu . refs' ' . refs'; "> <div style="background-color: magenta; grid-area: header">AAA</div> <div style="background-color: silver; grid-area: menu">BBB</div> <div style="background-color: green; grid-area: refs">CCC</div> </div></pre>
Отображение в браузере


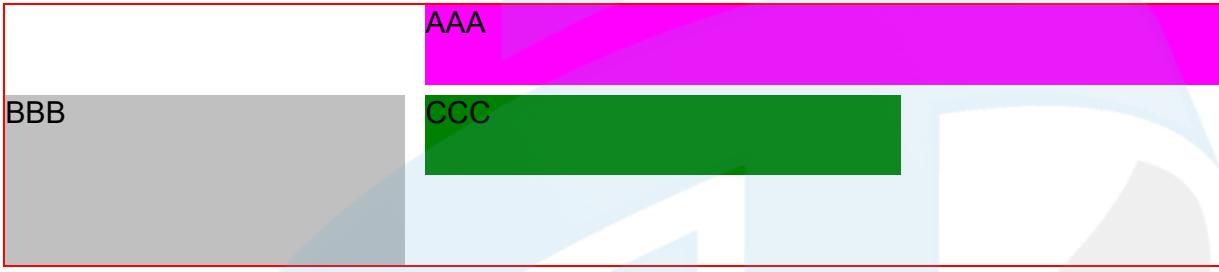
В стилевом свойстве **grid-template-areas** мы упоминаем столько строк в кавычках, сколько в grid-контейнере строк, и в каждой строке столько слов, сколько в grid-контейнере колонок, и каждым словом именуем соответствующую ячейку. Ячейки с одинаковым именем образуют именованную grid-область с этим именем. Символ «точка» означает, что эта grid-ячейка ни в какую именованную grid-область не входит (пробелы после точек расставлены для красоты, сколько их идёт подряд — не имеет значения).

Ширину зазора между grid-столбцами и grid-строками можно задать стилевыми свойствами **grid-column-gap** и **grid-row-gap**:

HTML-код

```
<div style="display: grid; border: solid red 1px;  
grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px;  
grid-column-gap: 10px; grid-row-gap: 5px;">  
  <div style="background-color: magenta; grid-column: 2 / 4; grid-row: 1">AAA</div>  
  <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4">BBB</div>  
  <div style="background-color: green; grid-column: 2; grid-row: 2">CCC</div>  
</div>
```

Отображение в браузере



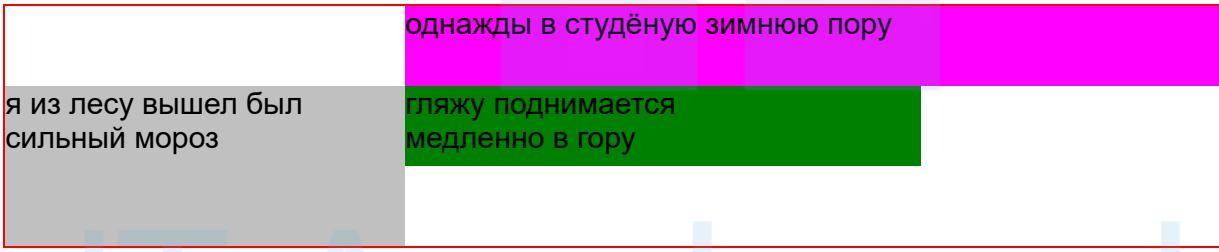
Эти зазоры не являются grid-линиями, не путайте их.

По умолчанию всё содержимое grid-ячеек (и grid-областей) растягивается на всю доступную площадь ячейки (области) и выравнивается к левому верхнему углу:

HTML-код

```
<div style="display: grid; border: solid red 1px;  
grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px;">  
  <div style="background-color: magenta; grid-column: 2 / 4; grid-row: 1">  
    однажды в студёную зимнюю пору</div>  
  <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4">  
    я из лесу вышел был сильный мороз</div>  
  <div style="background-color: green; grid-column: 2; grid-row: 2">  
    гляжу поднимается<br>медленно в гору</div>  
</div>
```

Отображение в браузере

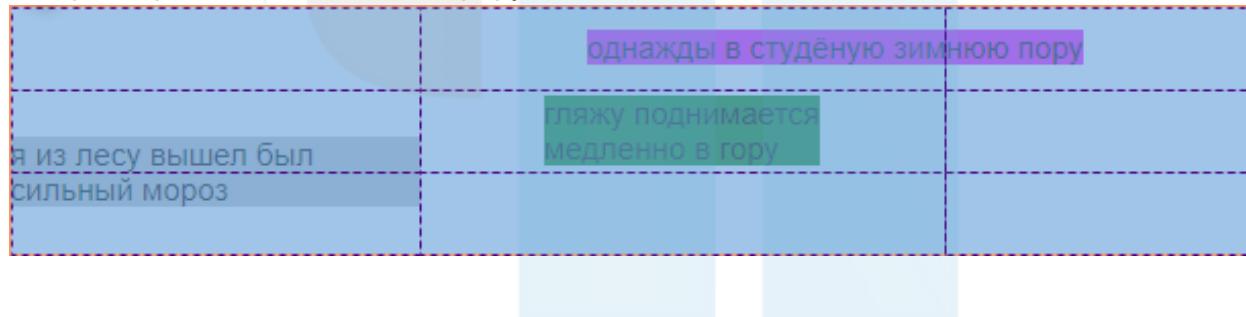


Стилевые свойства **justify-items** и **align-items**, применяемые к grid-контейнеру, позволяют содержимое grid-ячеек и grid-областей:

- **start** — выровнять влево и вверх соответственно;
- **end** — выровнять вправо и вниз соответственно;
- **center** — отцентрировать;
- **stretch** — растянуть на всю ширину и высоту соответственно (это значение по умолчанию).

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px; justify-items: center; align-items: center;"> <div style="background-color: magenta; grid-column: 2 / 4; grid-row: 1"> однажды в студёную зимнюю пору</div> <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4"> я из лесу вышел был сильный мороз</div> <div style="background-color: green; grid-column: 2; grid-row: 2"> гляджу поднимается
медленно в гору</div> </div></pre>
Отображение в браузере

На вкладке Elements в Chrome очень хорошо видно, что содержимое grid-областей теперь не растягивается, а центрируется:



Упомянутые выше стилевые свойства **justify-items** и **align-items** применяются к grid-контейнеру и действуют на все grid-элементы, а стилевые свойства **justify-self** и **align-self** (с такими же возможными значениями — **start**, **end**, **center**, **stretch**) применяются к конкретному grid-элементу и позволяют выровнять его как-либо иначе:

HTML-код
<pre><div style="display: grid; border: solid red 1px; grid-template-columns: 200px 1fr 150px; grid-template-rows: 40px 40px 40px; justify-items: center; align-items: center;"> <div style="background-color: magenta; grid-column: 2 / 4; grid-row: 1; align-self: stretch"> однажды в студёную зимнюю пору</div> <div style="background-color: silver; grid-column: 1; grid-row: 2 / 4"> я из лесу вышел был сильный мороз</div> <div style="background-color: green; grid-column: 2; grid-row: 2"> гляжу поднимается
медленно в гору</div> </div></pre>
Отображение в браузере
<p>однажды в студёную зимнюю пору</p> <p>гляжу поднимается медленно в гору</p> <p>я из лесу вышел был сильный мороз</p>

Здесь розовая ячейка по горизонтали по-прежнему не растягивается а центрируется, а по вертикали — растягивается на всю высоту grid-ячейки, т.к. ей задан стиль align-self: stretch.

Grid-раскладку поддерживают не все браузеры:

Chrome 57+, Firefox 52+, Opera 44+, IE 10+ и Edge 12+ (старый синтаксис), Edge 16+, Opera 44+, Safari 10.1+, iOS Safari 10.3+, Android Browser 76+

(9.65) Метаязыки описания стилей (Sass/SCSS)

Sass (от англ. *Syntactically Awesome Stylesheets* — синтаксически замечательные таблицы стилей) — мета-язык описания стилей.

Для работы требуется утилита, которая (на этапе разработки) код на языке Sass транслирует в CSS-файл; либо серверный модуль, который при отдаче Sass-кода клиенту «на лету» транслирует его в CSS-код.

1. Вместо фигурных скобок, для обозначения вложенности синтаксических конструкций используются отступы. Нет точек с запятой после стилевых описаний, поэтому, записывается только одно стилевое описание в строке.

Sass-код	соответствующий CSS-код
<pre>h1 font-size: 20px margin-top: 1em .remark color: red</pre>	<pre>h1 { font-size: 20px; margin-top: 1em; } .remark { color: red;</pre>

2. Вместо иерархических селекторов указываются селекторы с отступом.

Sass-код	соответствующий CSS-код
<pre>.remark color: red span.left margin-right: 1em a:hover text-decoration: underline</pre>	<pre>.remark { color: red; }.remark span.left { margin-right: 1em; }.remark span.left a:hover { text-decoration: underline;</pre>

3. Отступы также можно использовать для сокращения записи длинных CSS-свойств.

Sass-код	соответствующий CSS-код
<pre>.remark font: family: serif weight: bold size: 12px color: red</pre>	<pre>.remark { font-family: serif; font-weight: bold; font-size: 12px; color: red;</pre>

4. Чтобы не повторять значения стилевых свойств, можно использовать переменные, и вычислять значения выражений с переменными.

Sass-код	соответствующий CSS-код
\$maincolor = #804080 \$dist = 20px h1 color = \$maincolor margin-top = \$dist .remark color = darken(\$maincolor,25%) margin-bottom = \$dist / 2	h1 { color: #804080; margin-top: 20px; } .remark { color: #603060; margin-bottom: 10px; }

5. Чтобы не повторять целые куски стилевых описаний, можно использовать миксины.

Sass-код	соответствующий CSS-код
=pricetable th color: blue td color: red	.shortpricetable th { text-align: center; color: blue; } .shortpricetable td { color: red; }
.shortpricetable th text-align: center + pricetable	.fullpricetable th { font-size: 20px; color: blue; } .fullpricetable td { color: red; }
.fullpricetable th font-size: 20px + pricetable	

6. Миксин может получать аргумент.

Sass-код	соответствующий CSS-код
=infoblock(\$align) color: blue text-align = \$align .info1 + infoblock(right) .info2 + infoblock(left)	.info1 { color: blue; text-align: right; } .info2 { color: blue; text-align: left; }

**SCSS (Sassy CSS) — мета-язык описания стилей, наследник Sass.
Ориентирован на то, чтобы быть синтаксически ближе к CSS.**

Для работы требуется серверный модуль (препроцессор SCSS), который при отдаче SCSS-кода клиенту «на лету» транслирует его в CSS-код. Для LESS также существуют серверные модули, но более удобно использовать JavaScript-модуль, который на стороне клиента программно вносит нужные изменения в стилевые таблицы.

1. В отличие от Sass, фигурные скобки и точки с запятой используются так же как и в CSS. Поэтому, в простых случаях код SCSS вообще не отличается от кода CSS.

SCSS-код	соответствующий CSS-код
<pre>h1 { font-size: 20px; margin-top: 1em; } .remark { color: red; }</pre>	<pre>h1 { font-size: 20px; margin-top: 1em; } .remark { color: red; }</pre>

2. Иерархические селекторы тоже поддерживаются, но синтаксис немного другой.

SCSS-код	соответствующий CSS-код
<pre>.remark { color: red; span.left { margin-right: 1em; a:hover { text-decoration: underline; } } }</pre>	<pre>.remark { color: red; .remark span.left { margin-right: 1em; .remark span.left a:hover { text-decoration: underline; } } }</pre>

Также можно использовать символ «&» для ссылки на родительский селектор.

SCSS-код	соответствующий CSS-код
<pre>h1 { color: red; & span { color: blue; } &:hover { color: green; } }</pre>	<pre>h1 { color: red; } h1 span { color: blue; } h1:hover { color: green; }</pre>

3. Можно использовать переменные, и вычислять значения выражений с переменными, точно так же как в Sass, только синтаксис немного другой.

SCSS-код	соответствующий CSS-код
\$maincolor: #804080; \$dist: 20px; h1 { color: \$maincolor; margin-top: \$dist; } .remark { color: darken(\$maincolor,25%); margin-bottom: \$dist / 2; }	h1 { color: #804080; margin-top: 20px; } .remark { color: #603060; margin-bottom: 10px; }

4. Можно использовать миксины (в LESS другой синтаксис).

SCSS-код	соответствующий CSS-код
@ mixin rounded-corners(\$radius: 5px) { -webkit-border-radius: \$radius; -moz-border-radius: \$radius; border-radius: \$radius; } .header { @include rounded-corners; } .footer { @include rounded-corners(10px); }	.header { -webkit-border-radius: 5px; -moz-border-radius: 5px; border-radius: 5px; } .footer { -webkit-border-radius: 10px; -moz-border-radius: 10px; border-radius: 10px; }

5. В миксинах можно использовать @content.

SCSS-код	соответствующий CSS-код
@ mixin onlylarge(\$w) { @ media only screen and (min-width: \$w) { @ content; } }.txt { font-size: 20px; @ include onlylarge(600px) { color: blue } }	.txt { font-size: 20px; } @ media only screen and (min-width: 600px) { .txt { color: blue; } }

6. Можно склеивать строки.

SCSS-код
\$SmileDir: "images/smilies/"; .comment { background-image: url(\$SmileDir + "happy.gif"); }
соответствующий CSS-код
.comment { background-image: url("images/smilies/happy.gif"); }

7. Можно использовать циклы (LESS не поддерживает).

SCSS-код	соответствующий CSS-код
@for \$hhh from 1 through 4 { h#\${\$hhh} { text-align: center; font-size: \$hhh * 3 + 10px; } }	h1 { text-align: center; font-size: 13px; } h2 { text-align: center; font-size: 16px; } h3 { text-align: center; font-size: 18px; } h4 { text-align: center; font-size: 21px; }

8. Можно импортировать (вклеивать) другие SCSS-файлы. При этом препроцессор SCSS отдаст клиенту (браузеру) один, объединённый CSS-файл.

SCSS-код
@import "header.scss"; @import "footer.scss";

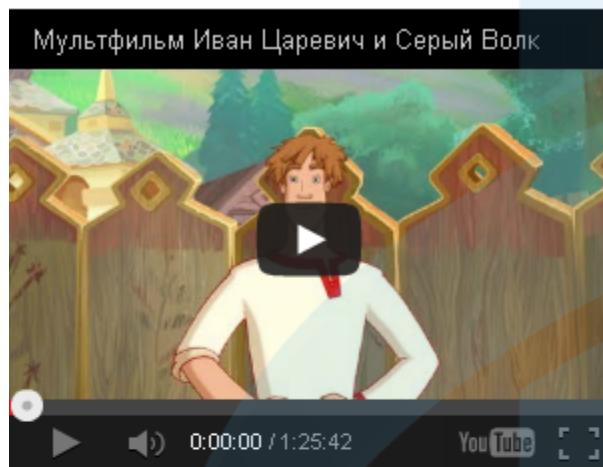
Домашнее задание №9а

Создать проект GALLERY. Сверстать согласно макету. Список видеороликов должен прокручиваться вертикальной полосой прокрутки.

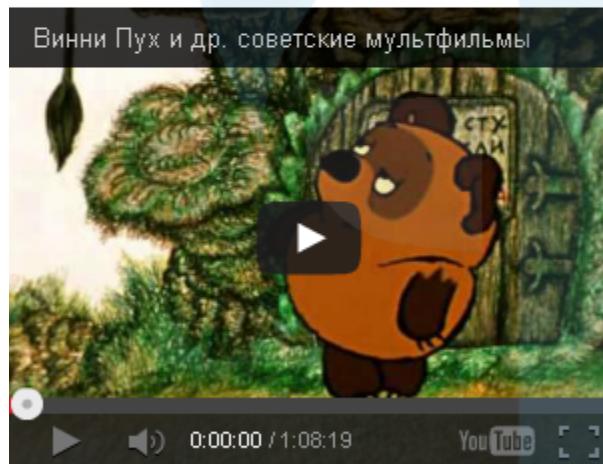
Тему видеороликов выбрать самостоятельно.

Каждый видеоролик должно быть возможно просмотреть как непосредственно в списке, так и открыть в отдельной вкладке браузера страницу youtube с данным видеороликом.

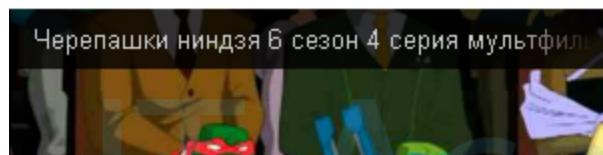
Мои любимые мультфильмы:



[смотреть в отдельном окне](#)



[смотреть в отдельном окне](#)



Домашнее задание №9b

Создать проект TEST_SVG.

С помощью онлайн-редактора нарисовать произвольный рисунок в формате SVG (например, домик) и вставить его на веб-страницу.

Доработать страницу так, чтобы при поднесении мыши к элементам рисунка элементы изменялись (размер, цвет, обводка...)

Домашнее задание №9c

Найти как можно больше способов отобразить на веб-странице данный рисунок (создавать рисунок не требуется, только придумать способы).

Обосновать достоинства и недостатки каждого найденного способа.

Важно: требуется чтобы рисунок мог растягиваться на всю ширину окна браузера без искажений.



Домашнее задание №9d

Создать проект TEST_ANIM.

Веб-страница должна содержать контейнер (элемент div), внутри которого расположено несколько элементов различного цвета, формы и т.д. (тоже могут быть элементами div).

Один из элементов должен быть постоянно анимирован (анимировать можно любое стилевое свойство, которое поддерживает анимацию).

При поднесении мыши к любой точке контейнера (даже вне элементов) сразу несколько неподвижных элементов должны анимироваться различными способами (т.е. у одного элемента может меняться расположение, у другого — фоновый цвет, у третьего — толщина рамки и т.д.).

Домашнее задание №9e

Создать новый проект *MAKET_FLEX* (за основу можно взять *MAKET_DIV* либо *MAKET_ADAPT*). Сверстать по макету резиновой блочной адаптивной вёрсткой с использованием flex-блоков.

В широком варианте ширина колонки с меню должна быть фиксированной.

В узком варианте меню должно опускаться ниже контента и растягиваться на всю доступную ширину, изображение также должно растягиваться на всю доступную ширину.

Фон меню должен задаваться цветом, а не фоновым изображением.

Макет широкой вёрстки (слева) — в библиотечном файле *football_w.png*, макет узкой вёрстки (справа) — в библиотечном файле *football_n.png*.

Гиперссылки могут вести куда угодно. Тексты могут быть произвольными.

ПРИВЕТ УЧАСТИКНАМ СОРЕВНОВАНИЙ!

Виды спорта:

- [конькобежный спорт](#)
- [биатлон](#)
- [футбол](#)
- [стрельба из лука](#)
- [прыжки в длину](#)
- [прыжки в высоту](#)
- [фигурное катание](#)

ФУТБОЛ



Футбол — командный вид спорта, в котором целью является забить мяч в ворота соперника ногами или другими частями тела (кроме рук) большее количество раз, чем команда соперника.

Есть 17 официальных правил игры, каждое из которых содержит список оговорок и руководящих принципов.

ФУТБОЛ



Футбол — командный вид спорта, в котором целью является забить мяч в ворота соперника ногами или другими частями тела (кроме рук) большее количество раз, чем команда соперника.

Есть 17 официальных правил игры, каждое из которых содержит список оговорок и руководящих принципов.

Виды спорта:

[конькобежный спорт](#) [биатлон](#)
[футбол](#) [стрельба из лука](#)
[прыжки в длину](#)
[прыжки в высоту](#)
[фигурное катание](#)

Домашнее задание №9f

Создать новый проект *MAKET_GRID*.

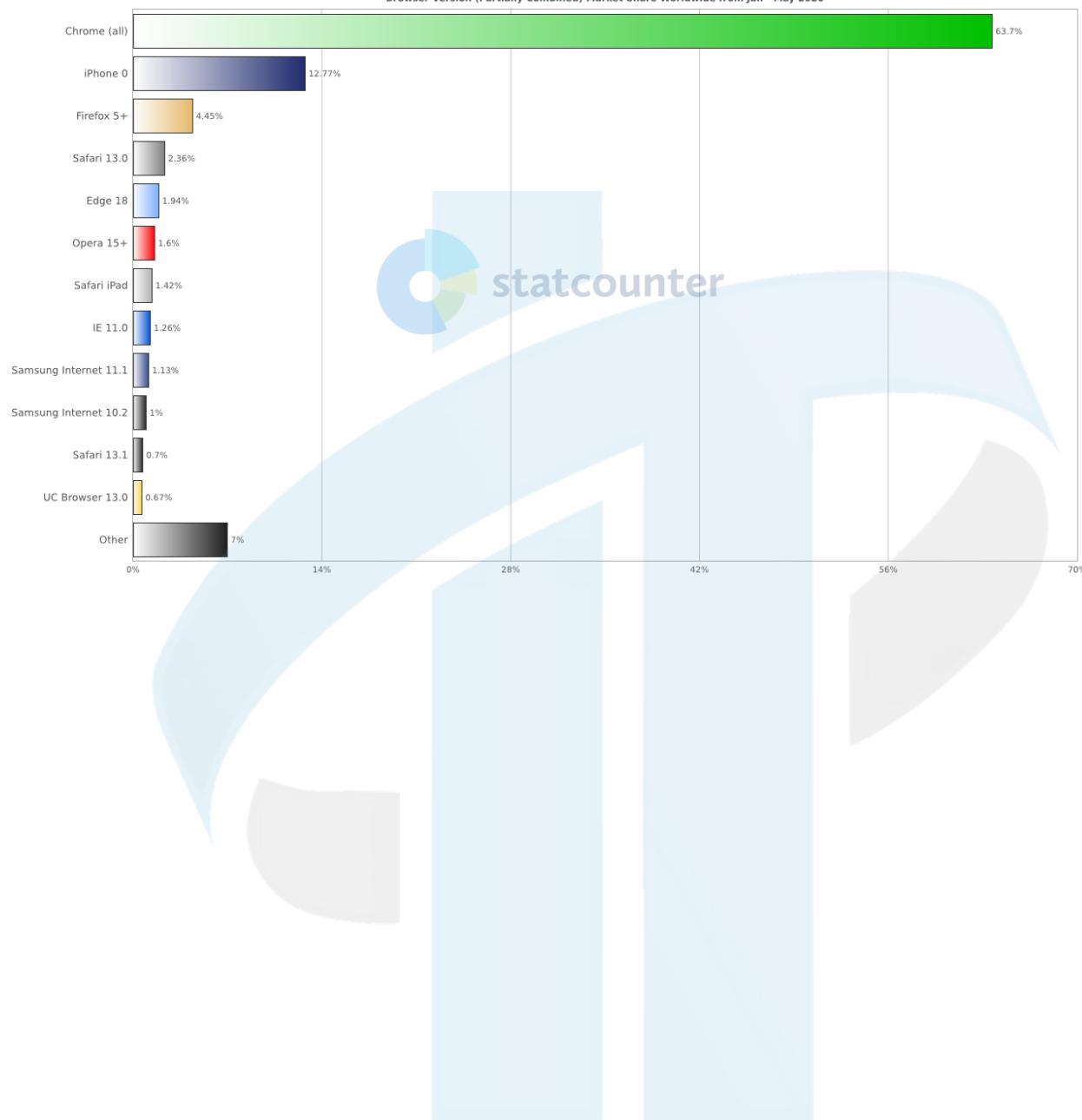
Сверстать grid-вёрсткой по тем же условиям, что и *MAKET_FLEX*.

Внимание! ВЕСЬ *МАКЕТ* нужно сделать grid-контейнером, а не только контейнер, содержащий меню и статью.

Приложения

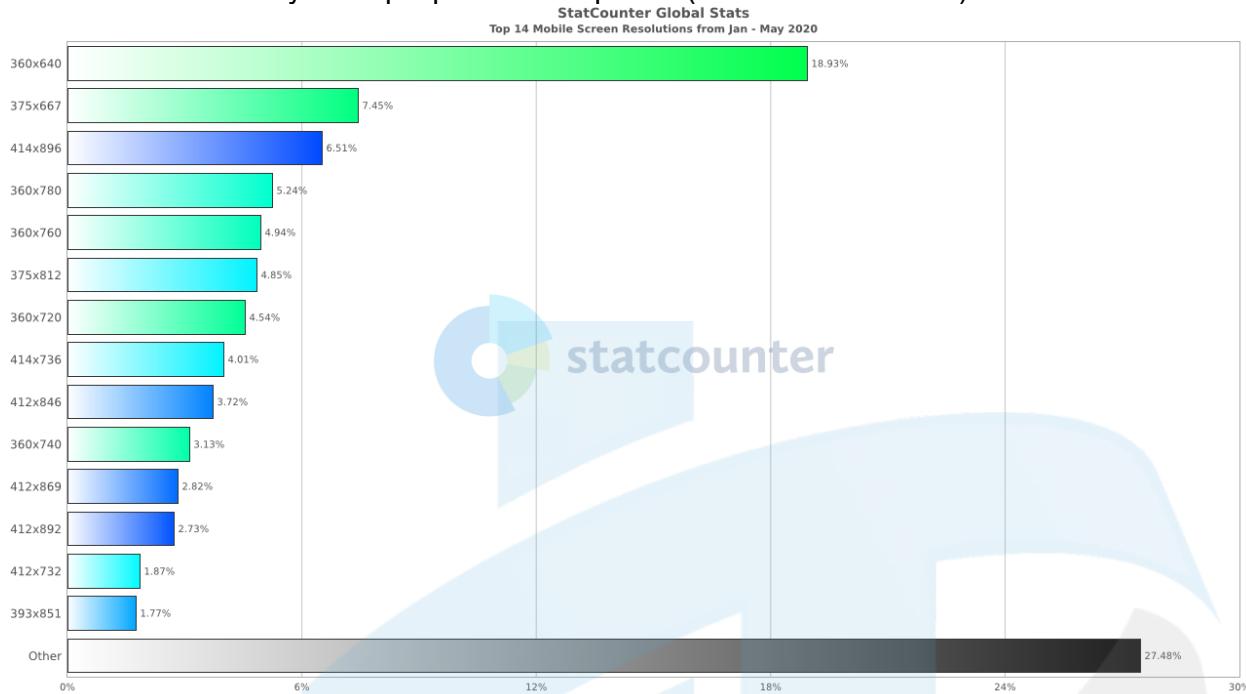
Статистика используемых версий браузеров (мобильные и десктопные)

StatCounter Global Stats
Browser Version (Partially Combined) Market Share Worldwide from Jan - May 2020

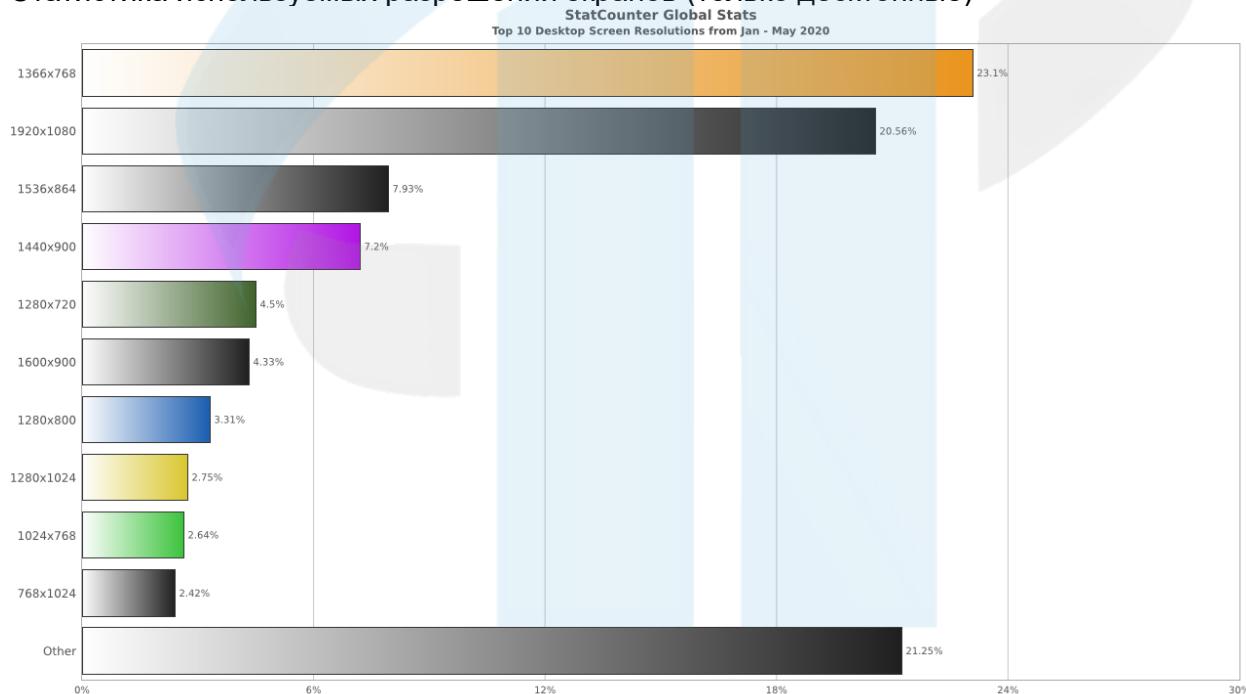


IT-Academy.by

Статистика используемых разрешений экранов (только мобильные)



Статистика используемых разрешений экранов (только десктопные)



Для заметок



IT-Academy.by

Для заметок



IT-Academy.by