**Вырезки из Макфарланда «Большая книга CSS»**

**ТЕХНОЛОГИИ, КОТОРЫЕ НАДО ЗНАТЬ**[**http://theasder.github.io/tools/2014/12/13/frotend-technologies-review.html**](http://theasder.github.io/tools/2014/12/13/frotend-technologies-review.html)

Новое в CSS:

* <http://css-live.ru/articles/razvlecheniya-s-css-schyotchikami.html> - рус статья по счетчикам CSS **counter**

#### <http://habrahabr.ru/company/wargaming/blog/245997/> - position: sticky – передвижной блок. counter – счетчик CSS. object-fit – фото ужимается\ расширяется в контейнере. Transition для z-index

#### <http://habrahabr.ru/company/wargaming/blog/241903/> - иконка без фоновых изображений. Наложение слоев в CSS. @supports – поддерживает ли браузер технлогию.

#### <http://habrahabr.ru/company/wargaming/blog/235889/> - анимация input без картинок,

#### <http://habrahabr.ru/company/wargaming/blog/228145/> - vmin и vmax – увеличение\уменьшение содержимого в зависимости от размера окна, стилизация выделенного текста, скролл на мобильных

1. В HTML5 тег **<header>** - для шапки**, <nav>** - для навигации, **<footer>** - для низа**, <figure>** - общий тег для фото и подписей, **<figcaption>** - тег для подписей, идет внутри **<figure>**
2. **a, p, img, h1 { font-weight: bold; }** выбор множества тегов идет через запятую
3. **\* { font-weight: bold; }** – выбрать все
4. **.banner \***выбирает все теги внутри элемента (контейнера), похоже на наследование
5. **h1 strong{ color: red; }** – Выбор дочернего элемента. В данном случае *любой* текст тега **<strong>,** находящегося внутри тега **<h1>,** бу­дет выделен красным цветом, Вы могли добиться того же результата, создав класс стиля, например .StrongHeader. Но в таком случае понадобилось бы вносить изменения в HTML, добавляя новый класс к тегу **<strong>**внутри заголовка.
6. Если вставить атрибут classвоткрывающий**<div>-**тег — **<divclass="news">,** то вы можете создать селекторы потомков, по-разному форматирующие HTML-теги внутри раздела новостей. Например,

**.news h2 { color: red; }**

**.news p{ color: blue; }**

Теперь содержимое <h2>-тегов внутри раздела новостей будет красным, а абза­цы — синими.

1. **p.intro{ color: yellow; } (БЕЗПРОБЕЛА!)** Это несколько иной вариант, выбирающий любой тег, стилизованный под класс **.intro,** который, в свою очередь, является потомком тега <p>.
2. **ПСВЕВДОКЛАССЫ.**

**L**o**V**e\**HA**te

**a:link** обозначает любую ссылку, по которой посетитель веб-страницы еще не переходил, даже если на нее не наведен указатель мыши. Это обычный стиль не­посещенных гиперссылок.

**a:visited** является ссылкой, по которой посетитель веб-страницы уже перехо­дил. Она сохраняется в истории браузера.

**a:hover** позволяет изменять вид ссылки, на которую посетитель навел указатель мыши.

**a:active**стиль для кратковремен­ного щелчка кнопкой мыши.

**:focus**разница с:hover- посетитель перемещает указатель мыши над ссыл­кой, а :focus — когда щелкают кнопкой мыши на текстовом поле (то есть требуется акцентировать внимание посетителя на кон­кретном (текущем) элементе веб-страницы). Этот стиль задает светло-желтый цвет любому текстовому полю, на котором посе­титель щелкает кнопкой мыши или в которое переходит с помощью табуляции:

**input:focus { background-color: #FFFFCC; }**

1. **ПСЕВДОЭЛЕМЕНТЫ (ПИШУТСЯ С ДВОЕТОЧИЕМ)**

**::first-letter** позволяет создавать буквицу — начальный сим­вол абзаца, который выделяется из остального текста, как в начале книжной главы. (две точки введено в CSS3, что бы отличать псведоклассы от псевдоэлементов)

**::first-line** – стилизация первой строки

**::before.** Псевдоэлемент :before позволяет добавлять сообщение, предшествующее определенному элементу веб-страницы. (напр. «ПОДСКАЗКА»)

**::after –** тоже самое, только в конце

**::selection -** Этот селектор, появившийся в CSS3, ссылается на элементы, ко­торые посетитель выбрал на странице. Например, когда посетитель щелкает на тексте и проводит над ним указатель мыши, браузер выделяет этот текст и посе­титель может затем скопировать текст. Обычно браузеры добавляют за текстом синий фон. Единственными свойствами, которые можно установить с помощью данного селектора, являются **color и background-color**

1. **СЕЛЕКТОРЫ АТРИБУТА – ВЫБОР ТЕГОВ КОНКРЕТНОГО ЭЛЕМЕНТА.**

**.photo[title]**выбирает все эле­менты стилевого класса **.photo** с HTML-атрибутом **title.**

**a[href^="http://"]** Символы ^= означают «начинается с», так что вы можете использовать этот селектор для форматирования любой ссылки, начинающейся с http://.

**a[href$=".pdf"]** – **$ заканчивается на**

**img[src\*="headshot"] – \* содержит в себе**

1. **СЕЛЕКТОРЫ ДОЧЕРНЫХ ЭЛЕМЕНТОВ**

**body> h1**выбирает лю­бой тег <h1>, дочерний по отношению к <body>. В отличие от селектора потомков, который применяется ко всем потомкам (то есть вложенным элементам), селектор дочерних элементов позволяет определить конк­ретные дочерний и родительский элементы.

**div:first-child**позволяет выбирать и форматировать первый элемент из <div>

**:last-child** – тоже самое, но наоборот

**:nth-child – выбор элемента строки, который является ДИТЕМ (более хрупкий, не использовать)**

**tr:nth-child(odd) { background-color: #D9F0FF; } –**нечетный

**tr:nth-child(even) { background-color: #FFFFFF; }** – четный

**trtd:nth-child(3n+2) { background-color:#900;} –** выбор каждой третей строки, начиная со второго элемента

**:first-of-type –**применяется к дочернему элементу, имеющему определенный тег.

**.sidebar p:first-of-type -** у нас есть боковая панель с классом **sidebar.** Для придания стилевого оформления первому абзацу этой боковой панели используется **надо указывать тег, который собираемся форматировать! (<p>)**

**.sidebarp:last-of-type –** тоже самое, но последний элемент

**img:nth-of-type(odd) { float: left; }** Работает так же, как и:nth-child(), но применяется к чередующимся

**img:nth-of-type(even) { float: right; }** дочерним элементам, имеющим определенный тег. Этот селектор может пригодиться при наличии больших абзацев текста с большим количеством фотографий. **Зависит от структуры! элементы должны быть одного уровня!!**

1. **СЕЛЕКТОРЫ СМЕЖНЫХ ЭЛЕМЕНТОВ**

**h1 + p {color: red}** задает красный цвет элементам одного уровня <h1> и <p>

**h1 ~ p -** определяет выбор всех смежных <p>-тегов, являющихся смежными элементами (то есть находящихся на одном и том же уровне) по отно­шению к заголовку **h1.**

1. **:not – позволяет отменить форматирование для некоторых элементов**

**p:not(.classy) { color: blue; }** Этот стиль задает тексту синий цвет во всех абзацах, к которым не применялсякласс**.classy**

* Нельзя использовать селекторы-потомки (такие как div p a), псевдоэлементы (такие как ::first-line), групповые селекторы или комбинации (такие как смеж­ный селектор одного уровня h2 + p).
* Нельзя в одной строке применять несколько **:not(…)-**селекторов.

1. **ПРИОРИТЕТНОСТЬ:  
   тег <div> = 1 балл p .content = 11 баллов  
   класс .main = 10 баллов  
   ID #aside = 100 баллов  
   <!important> = 1000 баллов**
2. **ПРАВИЛЬНОЕ ПРИКРЕПЛЕНИЕ ШРИФТОВ**

**@font-face** {

font-family: 'PTSans';

src: url('PTSansBoldItalic.eot');

src: url('PTSansBoldItalic.eot?#iefix') format('embedded-opentype'),

url('PTSansBoldItalic.woof') format('woff'),

url('PTSansBoldItalic.ttf') format('truetype'),

url('PTSansBoldItalic.svg') format('svg');

**font-weight: bold;**

**font-style: italic; }**

**если это есть, в самом тексте можно писать <em>ili<strong> для отображения курсива или полужира**

<http://css-tricks.com/flat-icons-icon-fonts/><http://css-tricks.com/html-for-icon-font-usage/> и <http://css-tricks.com/examples/IconFont/> - ресурсы с шрифтами с иконками.

Ссылки на шрифт: <http://www.google.com/fonts/>, <http://hellohappy.org/beautiful-web-type/>

16. **ОФОРМЛЕНИЕ ТЕКСТА**

16px – базовый размер шрифта текста (основной) = 1em = 100%

**62.5% = 10px, при задавании 1em будет 10px, 1.5em = 15px**  
24px = 1.5em = 150%.   
Шрифт наследуется, и вычисляется исходя от размеров родителя  
что бы избежать уменьшения шрифта в некоторых тегах **<ul style=”font-size: 75%”><ul><li><ul><ul>,** прописать  
**ul{font-size: 100%} или font-size: .75rem  
rem** исходит из значения корневого каталога, напр. **html {10px}**

**{text-transform: uppercase};** - все быквы будут в верхнем регистре, **lowercase** - в нижнем.**Capitalize** – делает первые буквы в словах большими

**font-variant: small-caps** – создание малых прописных букв

**text-decoration: underline, overline, line-through**

**letter-spacing: -1px** сжатие пространства между буквами  
**word-spacing** – сжатие\расширение пространства между словами.

**text-shadow**: горизонтального сме­щения (насколько левее или правее текста должна появиться тень), вертикаль­ного смещения (насколько выше или ниже текста будет тень), степень размытости тени и цвет отбрасываемой тени.

**text-shadow: -4px 4px 3px #999999;**

[**http://webexpedition18.com/articles/css3-text-shadow-property/**](http://webexpedition18.com/articles/css3-text-shadow-property/) **- статья по теням**

**line-height: 120% или line-height: 1,2 – высота строчки. Расстояние строки от верхней и нижней. 120% - стандарт**

Чтобы определить размер межстрочного интервала, браузер вычитает высоту шрифта из высоты строки. Допустим, размер шрифта составляет 12 пикселов. Межстрочный интервал, установленный в размере 150%, в итоге равняется 18 пикселам. Таким образом, браузер добавля­ет пустой промежуток размером 6 пикселов между двумя строками текста. Что бы у разных шрифтов была соблюдена пропорция, лучше использовать множитель, без процента.

**font-style, font-variant, font-weight, font-size, line-height и font-family** можно писать одной строкой:

**body {font: italic bold small-caps 18px/150% Arial, Helvetica, sans-serif}  
18px/150% =** размер шрифта и межстрочный интервал

**text-indent: 25px;** - абзацный отступ

**list-style-type: none**; - скрывает маркеры в списках<ul><ol>

**list-style-position: inside –** делает маркеры внутри текста (обтекание) или снаружи

1. **ИЗМЕНЕНИЯ РАЗМЕРОВ**

**Box-sizing: border-box –** в ширину содержимого включается **padding и border**

**Создание скругленных углов кнопок**

**border-radius: 40px/20px**; 40px – горизонтальный радиус, 20 - вертикальный

**overflow: scroll –** добавляет полосы прокрутки окну

**overflow: auto –** добавляет полосы прокрутки при необходимости

**overflow: hidden** – скрывает все содержимое, выходящее за рамки  
  
**max-width: 800px**. Установка максимально возможного размера окна.

**min-width: 300px** –тоже, но наоборот. Если вы заметили, что при изменении разме­ров окна вашего браузера элементы становятся настолько узкими, что раз­метка разваливается.

**clear: righ** - стилизуемый элемент смещается вниз относительно плавающего с уста­новленным правым обтеканием, параметр левого обтекания остается в силе;

1. **ФОНОВЫЕ РИСУНКИ**

**Background-repeat: repeat-x –** повтор рисунка вдоль оси Х  
**Background-repeat: no-repeat –** без повторa

**Background-position: left, center, right**, управление горизонтальным позиционированием

**Background-position: top, center, bottom –** управление вертикальным позиционированием

[**http://www.csszengarden.com**](http://www.csszengarden.com) **-** сайт с примерами изменения фона, не трогая содержимого

**background-attachment: fixed \ scroll** изображение скроллится при листании страницы или фиксировано

1. **РАЗМЕЩЕНИЕ ФОНОВОГО РИСУНКА**

**background-origin:border-box;** - изображение помещается начиная с верхнего левого угола рамки(border) окна, находится ПОД рамкой

**padding-box**— изображение помещается начиная с верхнего левый угол отступов (padding), но не под границей

**content-box** – изображение начинается от содержимого окна

**background-clip: border-box\padding-box \ content-box -** ограничивает область появления фонового изображения. Обычно фоновые изображения запол­няют всю область элемента, включая пространство за рамкой и отступами

**background-size: 100pxauto;** - задает ширину, а высота корректируется автоматически, что бы сохранить пропорции

**Ресурсы!**

[**http://www.morguefile.com**](http://www.morguefile.com) **,** [**http://www.freeimages.com/**](http://www.freeimages.com/) **,** [**http://openphoto.net**](http://openphoto.net) **, бесплатные фото**

[**http://search.creativecommons.org/**](http://search.creativecommons.org/) **- поиск бесплатных фото**

[**http://www.somerandomdude.com/work/sanscons/**](http://www.somerandomdude.com/work/sanscons/) **- маркеры для списка <li>**

[**http://www.colourlovers.com/patterns**](http://www.colourlovers.com/patterns) **,** [**http://www.kollermedia.at/pattern4u/classic**](http://www.kollermedia.at/pattern4u/classic)[**http://squidfingers.com/patterns/20**](http://squidfingers.com/patterns/20) **- мозаичный фон**

[**http://bgpatterns.com**](http://bgpatterns.com) **,** [**http://www.stripegenerator.com**](http://www.stripegenerator.com) **– сделать свою мозаику**

**background: url(scrollTop.jpg) center top no-repeat,**

**url(scrollBottom.jpg) center bottom no-repeat,**

**url(scrollMiddle.jpg) center top repeat-y;**

Размещение каскадных изображений типа свитка: топ и бот остаются прежними, а мид растягивается

1. **ГРАДИЕНТЫ**

Сайты: <http://lea.verou.me/css3patterns/>

<http://www.colorzilla.com/gradient-editor/> - создание градиентов. Мануал на стр. 284 Макфарланда

**Background**отменяет**background-image**

**background-image: linear-gradient(topleft,black,white)- перетегание цвета слева направо: из черного в белое**

background-image: -webkit-linear-gradient(top left, tomato, #0000ff);

Можно указывать в градусах (deg). Градус проходит от начала через ЦЕНТР элемента

**background-image: linear-gradient(45deg, rgb(0,0,0), rgb(204,204,204));**

**background-image: linear-gradient(bottom left, rgb(0,0,0), rgb(204,204,204));**

**Градиент без плавного перехода цветов (резкие переходы)**

background-image: -webkit-linear-gradient(top, white 0, white 33%,blue 33%, blue 66%, red 66%)

**Повторияющиеся градиенты (для мозаичного фона)**

background-image: -webkit-repeating-linear-gradient(right bottom, white 0, white 10px, black 10px, black 20px);

**Радиальные градиенты**

background: radial-gradient(white, black);

принимает форму фигуры (эллипс, если фигура вытянутая)

**background-image: radial-gradient(circle, red, blue);** для градиента в форме круга

**background-image: radial-gradient(20% 40%, circle, red, blue); -** размещение по ширине и вертикали

**background-image: radial-gradient(20% 40%, circle closest-corner, red, blue);** - тулитсякближайшемууглу

**closest-side** – k blijaiweistorone, **farthest-corner \ farthest-side** – k dalnei

**background-image: radial-gradient(20% 40%, circle, red 20%, orange 80%, yellow) –**вцентрекрасный, от 80% оранжевый, потом – желтый

1. **РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ**

Что бы сделать галерею из фото, у которых подписи разной длинны (высоты):

**inline-block**рассматривает каждую пару «изобра­жение/подпись» в качестве блока (у которого есть высота и ширина), но также и в качестве линейного элемента (то есть блоки могут выстраиваться в ряд). Кроме того, свойство **vertical-align** со значением **top** гарантирует выравнивание каждого тега <figure>по верхнему краю других тегов <figure>, имеющихся в данном ряду.

**Display: inline –** линейныеэлементыигнорируют **padding** and **margin,** атакже **width** and **height**

**Display:inline-block –** свойства блока, но размещаются на одной линии. Работает **text-align**

**Display:block** – размещаются друг под другом

1. **НАВИГАЦИЯ**

**Сайты с кнопками:**<http://css3button.net>, <http://hellohappy.org/css3-buttons/>, <http://webdesignerwall.com/tutorials/css3-gradient-buttons>

**Создание выпадающего меню:**<http://red-team-design.com/css3-animated-dropdown-menu/> , <http://purecssmenu.com/>

**Создание необычного меню(туториал)**<http://css.maxdesign.com.au/listutorial/>

**Готовые варианты меню:**<http://css.maxdesign.com.au/listamatic/>

**ul{ padding: 0; margin: 0, text-transform: none, list-style: none}** – резет для списка навигации меню

**a:link, :visited, :hover, :active. –** ссылка, посещенная, навели курсор, активная – пока грузится

Изменяться должны именно в этом порядке. Аббр**L**o**V**e\**Ha**te



Код для такой кнопки:

1. background-color: #ee432e;

2. background-image: linear-gradient(top, #ee432e 0%, #c63929 50%, #b51700 50%, #891100 100%);

6. border: 1px solid #951100;

7. border-radius: 5px;

8. box-shadow: inset 0px 0px 0px 1px rgba(255, 115, 100, 0.4), 0 1px 3px #333333; /\*две тени – внутр и внеш.\*/

9. padding: 12px 20px 14px 20px;

10. text-decoration: none;

11. color: #fff;

12. font: bold 20px/1 "Helveticaneue", helvetica, arial, sans-serif;

13. text-align: center;

14. text-shadow: 0px -1px 1pxrgba(0, 0, 0, 0.8);

Создание спрайтов (несколько изображений в одном, расположенных по вертикали)

Сайт, помогащий склеить изображения в одно <http://spritepad.wearekiss.com>

**a { background: #E7E7E7 url(images/pixy.png) no-repeat left top; }** размещениеизображения

**a:hover { background-position: 0 -39px; }** – смещение изображения вверх для показа другого

**Создание меню с выделенными подстраницами**

Смысл в том, что бы у основного класса (body) были все описания под разделы навигации  
**<style>**

**.features .featuresLink{**

**font-weight: bold;**

**background-color: #B2F511;**

**background-position: 3px 50%;}**

**</style>**

**<body class=”features”>**

**<ul><li class=”featuresLink”>FEATURES<li></ul>**

**</body>**

1. **ПРЕОБРАЗОВАНИЕ И АНИМАЦИЯ**

**Сайты:** <http://westciv.com/tools/transforms/index.html> - поворот и перемещение элементов

<http://blogs.msdn.com/b/ie/archive/2012/02/02/css3-3d-transforms-in-ie10.aspx> - описание 3D элементов

** transform: rotate(10deg) –** поворот элемента по часовой стрелке на 10град, накладывается на другие эл.

**transform: scale(.5,2) –** элемент становится в 2 раза меньше по width, но в 2 раза выше

**transform: scale(1, -1)** поворот элемента вокруг горизонтальной оси **(зеркальное)**

**transform: translate (1px 2px) –** смещение элемента. Остаются пустые пространства на прежнем месте. Писать в самом конце строчки трансформ, из-за перемещения

**transform: skew(45deg, 0);** наклон по горизонтальной оси, трехмерный обьект

**transform-origin: lefttop;** задает точку вращения – как на **рисунке**

**Переходы с анимацией**

Основноесвойство –**transition. Плавно переходит туда и назад САМ**

**.navButton {**

**background-color: orange;**

**transition-property: background-color;**

**transition-duration: 1s; /\* сколько будет длиться переход – 0.1s, 100ms\*/**

**}**

**.navButton:hover { background-color: blue;}**

**{transition-property: transform, color, background-color, border-color;**

**transition-duration: 1s, .25s, .75s, 2s;} –** перечисление времени соответственно свойства сверху

**transition-property: all; -**указываются все свойства или через запятую

**transition-timing-function: linear\ ease\ ease-in \ ease-out \ease-in-out. –**скоростьанимации

Сайт по примерам скорости анимации: <http://www.the-art-of-web.com/css/timing-function/>

**transition-timing-function: cubic-bezier(.20, .96, .74, .07);** сколько длится анимация по кривой

Сайт по настройке кривой Безье: <http://matthewlein.com/ceaser/>

**transition-delay: .5s;**- задержка начала перехода на 0.5 сек. Можно перечислить, как в **duration**

*При задании раскрывающихся меню с помощью CSS они слишком быстро пропадают при случайном выходе указателя мыши за их пределы. Но с помощью свойства transition-delay можно заставить меню быстро появляться и медленно исчезать. Для этого добавьте в исходный стиль следующий код:*

***transition-delay: 5s;***

*Затем добавьте задержку к стилю :hover:*

***transition-delay: 0;***

*Каким бы странным этот код ни показался, он заставляет переход по :hover происходить немедлен­но, без задержки. А вот возвращение к обычному стилю (при котором меню исчезает) занимает 5 секунд. За это время посетитель успеет вернуть свою непослушную мышь обратно на меню, пока оно не исчезло.*

1. **АНИМАЦИЯ**

**@keyframes*имяАнимации*{**

**from{ /\* Здесь перечисляются CSS-свойства \*/ }**

**to {/\* Здесь перечисляются CSS-свойства \*/} }**

**@keyframesglow{**

**0{ background-color: yellow;}** /\*можно перечислять несколько значений, \*/

**20%, 60% {background-color: blue;} /\***и использовать проценты везде – как в градиентах\*/

**40%, 80% { background-color: orange; }**

**100%{ background-color: red; }}**

**@keyframesfadeOut {**

**from { opacity: 0; } /\*обьектпрозрачный – невидим\*/**

**to { opacity: 1; } } /\*обьектстановитсявидимым\*/**

**.banner{**

**animation-name: fadeOut;** /\*указание название анимации, которое должно быть применено\*/

**animation-duration: 2s;** /\*сколько будет анимироваться\*/

**animation-timing-function: ease-in-out;**/\*как будет анимироваться\*/

**animation-iteration-count: infinite;**/\*сколько раз будет повторяться анимация\*/

**animation-direction: alternate**; - /\*анимация идет в обратном порядке\*/

**animation-delay: 5s; /**\*сколько ждать перед запуском\*/

**animation-fill-mode: forwards;** - /\*заставляет элемент оставаться в состоянии завершения анимации\*/**}**

можнозаписатькак

**.banner{fadeOut 2s ease-in-out infinite alternate 5s forwards;}**

fadeOut 2s ease-in-out 2 alternate 5s forwards;

**.banner:hover {animation-play-state: paused;}**– паузаанимацииприпроходемышинадней

1. **ТАБЛИЦЫ**

Обычно браузеры рисуют пустое пространство между ячейками. Пишем  
**table{ border-spacing: 0;**}

**border-collapse: collapse;** - границы сливаются, но **border-radius** работать не будет

Хинт: теги **<colgroup>и <col>,** указывающие на группу столбцов и на один отдельный столбец соответственно. Можно добавить к ним класс или тег, и прописывать любые характеристики, **width, color**

**<fieldset>** предназначен для группировки элементов, связан­ных друг с другом. (в нем несколько групп)

**<legend>**идет за тегом **<fieldset>,** и в нем содержится название для группы.

**<input type="text">, <input type="password">, <textarea>, <input type="submit">**

**<select> -** раскрывающиесясписки

**<label for="yourName">Your Name</label>**можно использовать также**<span>**

**<input name="yourName" id="yourName">**

**.label{**

**display: inline-block;** /\*выравнивание надписей (Ваш город, Ваше имя), \*/

**width: 20em;** /\*что бы они были напротив форм заполнения\*/

**vertical-align: top;**

**text-align: right;**

**margin-right; 15px;**}

**:focus** изменяет внешний вид текстового поля при щелчке на нем кнопкой мыши

**:checked** работает с переключателя­ми и флажками

**:enabled \ :disabled –** отключение\включение свойств

**Пример формы<input>, <text> – на стр 407-412 книги**

**Примертаблицы 2х2**

**<table class=”inventory”>**

**<colgroup>/\*тег для количества групп\*/**

**<colid=”Name1”>**

**<colid=”Name2”> /\*задаем IDчто бы применять характеристики сразу к столбцам\*/**

**</colgroup>**

**<tr>**

**<th>Name1</th> /\*Заголовок 1. Идет в первой строчке\*/  
 <th>Name2</th> /\*Заголовок 2. Идет в горизонтальной строчке за первым\*/**

**</tr>**

**<tr>**

**<td>info…blabla</td> /\*Строка1. Идет под первым заголовком\*/  
 <td>info…blabla</td> /\*Строка2. Идет в горизонтальной строчке за первой\*/**

**</tr>**

1. **РАЗМЕТКА**

**Сайт:** StencilKit (<http://developer.yahoo.com/ypatterns/wireframes>)- разметка страницы в иллюстраторе, скачать

<http://blog.html.it/layoutgala/> - 40 примеров разметки

<http://gridinator.com/> создание многоколоночной верстки

Для поисковых машин и скринридеров важно, что бы основной раздел был сверху.

Grid – система разбивки по колонкам, обычно 12 колонок

Foundation (<http://foundation.zurb.com/docs/grid.php>) и

Twitter Bootstrap (<http://twitter.github.com/bootstrap/scaffolding.html#grid>)

Можно создать свой ГРИД – искать в нете

**Мультиколонки**

**.multicol {**

**column-count: 3; - /\*создает количество колонок\*/**

**column-gap: 1em; /\*расстояние между колонками\*/**

**column-rule: 1px dotted black; /\*разделительная линия между колонками\*/**

**}**

**Проблемы компоновки перемещаемыхкомпонентовстр 438-440**

1. Сделать тег <div>со знаком **float,** что бы он мог расширятся под теги в нем
2. Написать **overflow:hidden**
3. **Перед закрытием </div>добавить <brclass=”clear”>. В CSS написать br.clear{clear:both}**

**Что бы добавить цвет колонкам до конца страницы**

Используем **background-image** или **linear-gradient,** если ширина колонок указана в процентах

**Flexbox, Grid Layout**

**Сайты:habr, frontender**

**Если элемент выпадает из контейнера**

**box-sizing: border-box;** - учитывается**margin и border**

Для**IE: .clear{ zoom: 1; }** добавить класс к контейнеру, содержа­щему плавающие элементы

**Как сделать продляющийся background по ширине, а текст центрировать в одной области**

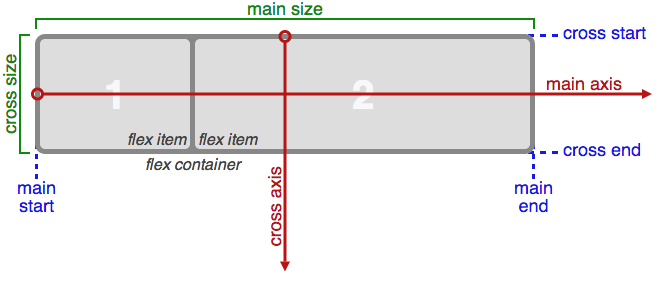
**.head nav, .main h1, .foot p, .wrapperCenter{**

**Max-width: 800px; /\*прописать макс-ширину для всех элементов, но не background\*/**

**Margin: 0 auto;**

**Разметка для HTML5 -** [**http://habrahabr.ru/post/214407**](http://habrahabr.ru/post/214407)

1. **FLEXBOX**

**main-axis** - главная ось, вдоль которой располагаются flex-элементы. Oна необязательно должна быть горизонтальной, всё зависит от свойства **justify-content**

**main-start** | **main-end** - flex-элементы размещаются в контейнере от позиции main-start до позиции main-end.

**main size** - ширина или высота flex-элемента в зависимости от выбранной основной величины. Основная величина может быть либо шириной, либо высотой элемента.

**cross axis** - поперечная ось, перпендикулярная к главной. Её направление зависит от направления главной оси.

**cross-start** | **cross-end** - flex-строки заполняются элементами и размещаются в контейнере от позиции cross-start и до позиции cross-end.

**cross size** - ширина или высота flex-элемента в зависимости от выбранной размерности равняется этой величине. Это свойство совпадает с width или height элемента в зависимости от выбранной размерности

**display: flex | inline-flex; /\*как блок или как инлайн-эелемент\*/**

Имейте в виду:

* CSS-столбцы **columns** не работают с flex-контейнером
* **float, clear и vertical-align** не работают с flex-элементами

**flex-direction: row | row-reverse | column | column-reverse /\*устанавливает ось main-axis\*/**

* **row** (по умолчанию): слева направо для **ltr,** справа налево для **rtl;**
* **row-reverse:** справа налево для **ltr,** слева направо для **rtl;**
* **column:** аналогично **row,** сверху вниз;
* **column-reverse:** аналогично **row-reverse,** снизу вверх.

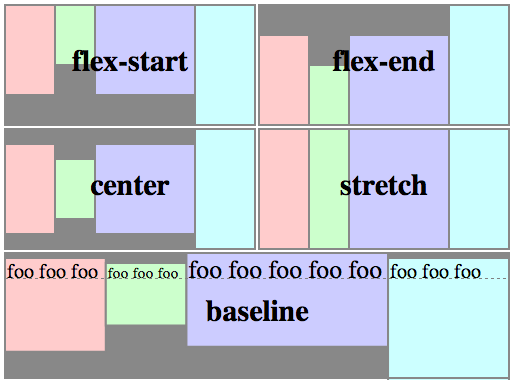
**flex-wrap: nowrap | wrap | wrap-reverse /\*контейнер будет однострочным или многострочным\*/**

* **nowrap**(по умолчанию): однострочный / слева направо для ltr, справа налево для rtl;
* **wrap:** многострочный / слева направо для ltr, справа налево для rtl;
* **wrap-reverse:** многострочный / справа налево для ltr, слева направо для rtl.

**flex-flow: <'flex-direction'> || <'flex-wrap'>** /\* сокращение для свойств flex-direction и flex-wrap, вместе определяющих главную и поперечную оси. По умолчанию принимает значение row nowrap.\*/

**justify-content: flex-start | flex-end | center /\*Определяет выравнивание относительно главной оси\*/**

* **flex-start** (по умолчанию): элементы сдвигаются к началу строки;
* **flex-end:** элементы сдвигаются к концу строки;
* **center:** элементы выравниваются по центру строки;
* **space-between**: элементы распределяются равномерно (первый элемент в начале строки, последний — в конце);
* **space-around:** элементы распределяются равномерно с равным расстоянием между собой и границами строки.

**align-items: flex-start | flex-end | center /\*выравнивание относительно поперечной строки\*/**

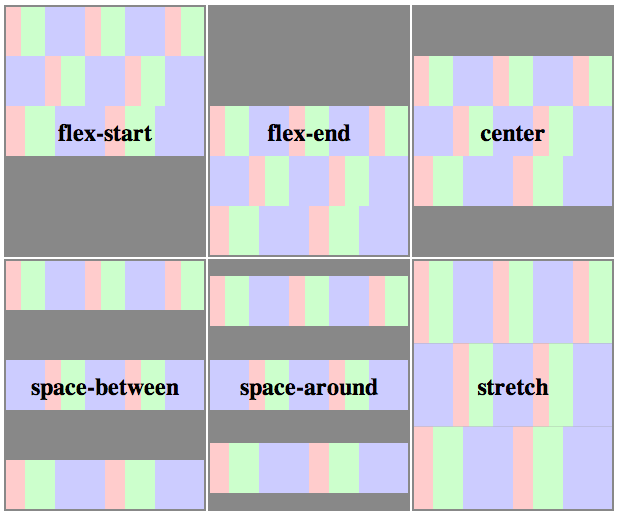
**flex-start:** граница cross-start для элементов располагается на позиции cross-start;

**flex-end:** граница cross-end для элементов располагается на позиции cross-end;

**center:** элементы выравниваются по центру поперечной оси;

**baseline:** элементы выравниваются по своей базовой линии;

**stretch** (по умолчанию): элементы растягиваютcя, заполняя контейнер (с учётом min-width/max-width).

**align-content: flex-start | flex-end | center | space-between | space-around | stretch /\*** Выравнивает строки flex-контейнера при наличии свободного места на поперечной оси аналогично тому, как это делает justify-content на главной оси.\*/

**Замечание**: это свойство не работает с однострочным flexbox.

**flex-start:** строки выравниваются относительно начала контейнера;

**flex-end:** строки выравниваются относительно конца контейнера;

**center:** строки выравниваются по центру контейнера;

**space-between:** строки распределяются равномерно (первая строка в начале строки, последняя — в конце);

**space-around:** строки распределяются равномерно с равным расстоянием между собой;

**stretch**(по умолчанию): строки растягиваются, заполняя свободное пространство.

**Применяется к дочерным элементам FLEX**

**order: <integer>;** /\*управляет порядком расположения элементов\*/

**flex-grow: <number> (по умолчанию 0) /\*** Определяет, какую долю свободного места внутри контейнера элемент может занять. Если у всех элементов свойство flex-grow задано как 1, то каждый потомок получит внутри контейнера одинаковый размер. Если вы задали одному из потомков значение 2, то он заберёт в два раза больше места, чем другие.\*/

**flex-shrink: <number> (default 1) – /\*элемент занимает меньше места, как со свойством grow\*/**

**flex-basis: <length> | auto (default auto) /\*** Определяет размер по умолчанию для элемента перед распределением пространства в контейнере.\*/

**flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ] /\*** сокращение для flex-grow, flex-shrink и flex-basis. Второй и третий параметры (flex-shrink, flex-basis) необязательны. Значение по умолчанию — 0 1 auto.\*/

**align-self: auto | flex-start | flex-end | center | baseline | stretch /\*** Позволяет переопределить выравнивание, заданное по умолчанию или в align-items, для отдельных flex-элементов.\*/

**Сайты:** [**http://frontender.info/a-guide-to-flexbox**](http://frontender.info/a-guide-to-flexbox)**,** [**http://habrahabr.ru/post/242545**](http://habrahabr.ru/post/242545) **,**

[**http://www.slideshare.net/radyno/flexbox-39134410**](http://www.slideshare.net/radyno/flexbox-39134410)[**http://html5.by/blog/flexbox/**](http://html5.by/blog/flexbox/)

1. **РАЗМЕТКА ПОД РАЗНЫЕ ЭКРАНЫ**

[**http://css-tricks.com/convert-menu-to-dropdown**](http://css-tricks.com/convert-menu-to-dropdown)- как сделать сворачивающееся меню с javascript

<http://bradfrost.com/blog/web/complex-navigation-patterns-for-responsive-design> - еще одна меню

<http://bradfrost.com/blog/web/responsive-nav-patterns> - еще одна меню

Смещение background при уменьшении экрана: <http://www.lukew.com/ff/entry.asp?1514>

**Подключаем стили в HTML**

**<link href="css/medium.css" rel="stylesheet" media="(min-width:481px) and (max-width:768px)">**

**Отключаем уменьшение масштаба на мобильниках**

**<meta name="viewport" content="width=device-width">**

**Если включаем в CSS**

Создаются в начале таблицы, располагаются так, что бы работала каскадность:

**@import url(css/base.css); /\* нет медиазапроса, применять ко всем \*/**

**@import url(css/medium.css) (min-width:481px) and (max-width:768);**

**@import url(css/small.css) (max-width: 480px);**

**Встраивание медиазапроса в CSS**

**/\* Сюда помещаются стили, перезапускающие исходные настройки браузера \*/**

**/\* Сюда помещаются стили для браузеров настольных устройств и основные стили**

**для всех устройств \*/**

**body {**

**/\* свойства для тела документа \*/**

**}**

**/\* только для дисплеев средней ширины \*/**

**@media (min-width: 481px) and (max-width:768px) {**

**body {**

**/\* свойства, применяемые только к браузерам планшетных устройств \*/**

**}**

**}**

**/\* только для дисплеев малой ширины \*/**

**@media (max-width:480px) {**

**body {**

**/\* свойства, применяемые только к браузерам телефонов \*/**

**}**

**}**

Что бы картинка нормально отображалась, следует указать ей размер в процентах. Процент будет рассчитываться исходя от размеров контейнера

К изображению можно дать два класса:  
**<img src=”tralala.jpg” class=”half right”> /\*** в CSS пишем **img.half {…} img.right{…} \*/**

**Тестирование адаптивного дизийна**

Изменение размеров окна: <http://responsivepx.com>, <http://www.responsinator.com>

Адаптивные изображения с SVG – муть какая-то <http://frontender.info/clown-car-technique-solving-for-adaptive-images-in-responsive-web-design>

**Работа с видео**

[**http://fitvidsjs.com**](http://fitvidsjs.com)– JS плагин для встраивания видео с **youtube**

1. **ПОЗИЦИОНИРОВАНИЕ**

**position: absolute –** размещение относительно окна браузера или элемента, к которому применено позиционирование! Другие элементы его не видят, возможен наплыв.

**position: relative** – применяется не к элементу, а к его родителю! Задача - в установке новой точки привязки для абсолютно позиционированных элементов, которые вложены в него. Сам родитель располагается относительно своего родителя тоже. Если хоть один корневой элемент не абсолютный, контент не плывет.

**position: fixed –** закрепляет на одном месте. При прокрутке страницы, элемент остается там, где был. Располагается относительно браузера или элемента с позиционированием.

**z-index: 1** – позиционирование по **Z оси** относительно родителя. Родитель всегда имеет степень **относительно детей 0**, даже ему присвоить z-index: 1000. **z-index: -1 –** смещает элемент за родителя

Создание невидимого элемента:

**Opacity: 0 \ 1 –** свойство можно анимировать  
**display: none**

**Visibility: visible / invisible**

**Figcaption{**

**Opacity 0;** /\*элемент невидим, пока не пройдет указатель мыши сверху\*/

**Transition: opacity 1s}** /\*элемент будет анимировать видимость на протяжении 1с\*/

**figure:hover figcaption { opacity: 1; }** /\*при проходе мыши над элементом, становится видим его ребенок\*/

1. **СТИЛИ ДЛЯ ПЕЧАТИ**

**<link rel="stylesheet" media="print" href="print.css"/>** так записывается добавка в стили

**@import url(print.css) print;** - тоже подходит, но IE8 не понимает

**Для добавление в CSS таблицу выглядит так:**

**@media print {**

**\* { color: #000 !important; background: #FFF !important; }**

**h1 { font-size: 24pt !important; } /\*если возникают конфликты стилей, поможет это\*/**

**p { font-size: 12pt !important; }**

**a[href^="http://"]:after { /\*печатает ссылки на ресурсы в скобочках, при этом разбивает слова\*/**

**content: " (" attr(href) ") ";**

**word-wrap: break-word; }**

**}**

**Что бы распечатать видимый баннер, нужно хранить его в img. Можно так же img спрятать, а в версии печати сделать видимым**

**Так же прячутся области, которые не нужно печатать. Через display: none**

**16px = 12pt** в большинстве принтеров

**.break\_before { page-break-before: always; } /\*появляется всегда сверху\*/**

**.break\_after { page-break-after: always; } /\*появляется снизу, че-то типо колонтитула\*/**

**Посмотреть, как будет выглядеть страница при печати, можно через нажатие «печать» в браузере**

1. **ДЕЛАЕМ КОД СОВЕРШЕНЕЕ**

* Оставлять комментарии.
* Давать классам понятные название, не основанные на их местоположении. **Правильно: .news, .events, .secondaryContent, .mainNav, .gallery, .figure, .banner, .wrapper**
* Использовать несколько стилей в одном элементе для облегчения изменения характеристик
* **Группировать стили:**

- Группируйте стили, которые применяются к связанным частям страницы. Груп­пируйте все правила, которые применяются к тексту, изображениям и ссылкам в баннере страницы, в одном месте, правила, которые относятся к главной на­вигации, — в другом, а стили для основного содержимого — в третьем.

- Группируйте стили со связанными задачами (имеющими друг к другу отноше­ние). Помещайте все стили для разметки в одну группу, для оформления — в дру­гую, для ссылок — в третью и т. д.

Использовать комментарии для разделения груп стилей!

* Разбивать стили на несколько групп и собирать их вместе через **@import** в одной таблице стилей
* Для того, что бы браузер юзера обновил **CSS,** пишем **<link rel="stylesheet" href="main.css?v=1">**

1. **MAP, AREA**

Статья: <http://www.onextrapixel.com/2009/04/30/how-to-create-multiple-links-on-a-single-image-with-image-map/>

* Берем нужное изображение, открываем в Photoshop
* Если нужно выделить прямоугольник - выделяем его прямоугольником, смотрим верхний левый угол - его координаты. Это можно сделать, навев курсор на пересечение границ и посмотрев в инфо координаты. Записываем координаты.
* Делаем тоже самое с нижним правым углом - записываем координаты. Переносим в html координаты, разделяя запятыми сначала первый угол, потом второй:

**<body>**

**<img src="img/info.jpg" usemap="#infographics\_1">**

**<map name="infographics\_1">**

**<area shape="rect" coords="31, 130, 226, 206" href="http://site.com">**

**</map>**

**</body>**