**Понятие HTML**HTML расшифровывается как Hyper Text Markup Language (язык разметки гипертекста). Это XML-подобный язык, предназначенный для разметки веб-страниц. HTML-документ имеет определённую структуру, которая тесно связана с понятием DOCTYPE. Для того, чтобы просматривать HTML-документы, существуют приложения, называемыми браузерами. Браузеры занимаются отправкой различных запросов к серверу, получением различных файлов, в том числе и HTML, парсингом HTML-документов и их отрисовкой.

**Структура HTML-документа**HTML-документ имеет строгую структуру.

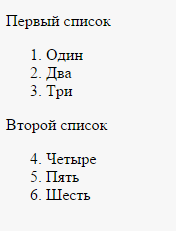
1. Сверху документа всегда находится элемент [<!DOCTYPE>](http://htmlbook.ru/html/!doctype). Он определяет тип текущего документа, который необходим для корректного отображения документа в браузере.
2. Затем следует тег html. Внутри обязательно находятся 2 секции: head и body.
3. Секция [head](http://htmlbook.ru/html/head). Нужна для хранения различной информации о документе.  
   Содержит элементы:
   1. [title](http://htmlbook.ru/html/title). Заголовок страницы
   2. [meta](http://htmlbook.ru/html/meta). Мета-теги содержат различную информацию о странице: кодировку, описание, ключевые слова, способ отображения в режиме совместимости и пр.
   3. [link](http://htmlbook.ru/html/link). Служит для подключения дополнительного файла, например css-файла
   4. [script](http://htmlbook.ru/html/script) (может располагаться не только в head, но и в body). Подключает JavaScript-файл.
4. Секция [body](http://htmlbook.ru/html/body). Предназначена для хранения содержимого страницы.

**HTML-теги и их виды**Тег - это структурная единица HTML-документа.   
Теги бывают [парными](http://htmlbook.ru/samhtml/tegi/parnye-tegi), например:  
<p>[Содержимое эелемента]</p>  
Существует некоторое количество одиночных самозакрывающихся тегов, например <input/>.  
  
Все HTML-элементы можно условно на разделить на 2 категории:

1. ***Строчные***. Предназначены для форматирования фрагментов текста.  
   К ним относятся теги span, a, i, b, strong, u, img, q, label, input, select.  
   <http://htmlbook.ru/samlayout/blochnaya-verstka/strochnye-elementy>  
   Основные свойства:
   1. Всегда занимают ширину, равную ширине содержимого
   2. Имеют высоту, равную высоте содержимого
   3. Переносятся на следующую строку, только если текущая переполнена
   4. Имеют только горизонтальные внешние отступы (margin-left и margin-right)
   5. Имеют только горизонтальные внутренние отступы (padding-left и padding-right)
   6. По умолчанию выравниваются с другими элементами вертикально по [базовой линии](https://ru.wikipedia.org/wiki/Базовая_линия_(дизайн_шрифта))
2. ***Блочные***. Предназначены для оформления блоков разметки.  
   К ним относятся: div, p, form, h1, h2, h3, h4, h5, h6.  
   <http://htmlbook.ru/samlayout/blochnaya-verstka/blochnaya-model>  
   <http://htmlbook.ru/samlayout/blochnaya-verstka/blochnye-elementy>Основные свойства:
   1. Всегда занимают 100% *доступной* ширины
   2. Всегда переносятся на новую строку
   3. Могут иметь любые внутренние отступы
   4. Могут иметь любые внешние отступы
   5. При отсутствии контента имеют высоту, равную 0
   6. При наличии контента имеют высоту, равную высоте содержимого

***Задания:***

1. Составить HTML-документ, подключить к нему пустой CSS-файл и пустой JS-файл, поместить в секцию head все необходимые meta-теги.
2. В теле документа оформить [стихотворение](http://www.100bestpoems.ru/item_info.php?id=4129), учитывая [правила типографики](https://www.artlebedev.ru/kovodstvo/sections/62/).
3. Взять [отсюда](https://referats.yandex.ru/referats/) любой текст и поместить его ниже на страницу в блок шириной 600 пикселей (ширину можно задать при помощи атрибута style). Оформить также в соответствии с правилами типографики.
4. Сделать при помощи HTML-тегов некоторые слова в этом тексте подчёркнутыми, жирными и курсивными.
5. Добавить в этот же документ оформленную с помощью HTML формулу с [картинки](http://htmlbook.ru/files/images/practical/13.png).
6. С помощью тегов <ol>, <ul>, <li> составить списки, как на картинках:
   1. [задание 1](http://htmlbook.ru/files/images/practical/06.png)
   2. [задание 2](http://htmlbook.ru/files/images/practical/07.png)

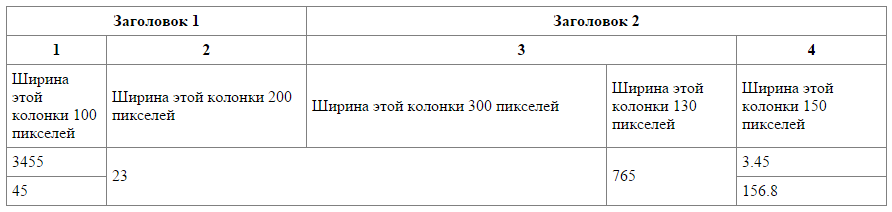
 **Таблицы**Таблицы иногда относят к блочным элементам, однако это не совсем так. В отличие от блочных, таблицы рассчитывают свою ширину исходя из содержимого ячеек, и внутренних отступов они не имеют. Таблица создаётся при помощи тега table, который служит контейнером для различных её элементов:

1. [thead](http://htmlbook.ru/html/thead). Шапка таблицы. Располагается только внутри <table>. Внутри должен содержать элементы <tr>.
2. [tbody](http://htmlbook.ru/html/tbody). Тело таблицы. Располагается только внутри <table>. Внутри должен содержать элементы <tr>.
3. [tfoot](http://htmlbook.ru/html/tfoot). Футер таблицы. Располагается только внутри <table>. Внутри должен содержать элементы <tr>.
4. [tr](http://htmlbook.ru/html/tr). Строка таблицы. Может находиться как внутри <table>, так и внутри <thead>, <tbody>, <tfoot>.
5. [td](http://htmlbook.ru/html/td). Ячейка таблицы. Должна находиться внутри <tr>
6. [th](http://htmlbook.ru/html/th). Ячейка шапки таблицы. Должна находиться внутри <tr>. Отличается от обычной ячейки жирным начертанием и выравниванием текста по середине.
7. [caption](http://htmlbook.ru/html/caption). Заголовок таблицы. Должен находиться внутри <table>, строго вначале.
8. [colgroup](http://htmlbook.ru/html/colgroup). Используется для задания ширины и нек. др. свойств колонкам таблицы. Помещается внутри <table>. Может быть как одиночным, так и парным, если содержит элементы <col>.
9. [col](http://htmlbook.ru/html/col). Аналогично colgroup. Может находиться как внутри <table>, так и внутри <colgroup>.

Для объединения ячеек используют атрибуты [colspan](http://htmlbook.ru/html/td/colspan) и [rowspan](http://htmlbook.ru/html/td/rowspan).

**Атрибуты**Атрибут - это пара вида [ключ=”значение”] (в некоторых случаях допустимо просто [ключ]), которая находится внутри открывающего тега. Различают атрибуты [универсальные](http://htmlbook.ru/samhtml5/globalnye-atributy)**,** [дата-атрибуты](http://htmlbook.ru/samouchitel-html5/atributy-data)**,** а также специфичные для конкретных тегов.

***Задания:***

1. Оформить таблицу, как на [картинке](http://htmlbook.ru/files/images/practical/08.png).
2. Оформить таблицу, как на картинке только при помощи HTML-тегов и атрибутов:

**Формы**

[Форма](http://htmlbook.ru/html/form) —  элемент, предназначенный для [сбора и отправки](http://htmlbook.ru/samhtml5/formy/otpravka-dannykh-formy) на сервер данных, вводимых пользователем. Создаётся при помощи тега form. Форма может содержать такие элементы, как:

1. [input](http://htmlbook.ru/html/input)
2. [select](http://htmlbook.ru/html/select)
3. [textarea](http://htmlbook.ru/html/textarea)
4. [button](http://htmlbook.ru/html/button)

**Понятие CSS**Любой элемент в HTML-документе можно стилизовать тремя способами:

1. При помощи универсального атрибута style:  
   <p style=”color: red”>text</p>
2. При помощи тега style (его принято располагать в секции head):  
   <style>  
    p {  
    color: red;  
    }  
   </style>
3. Поместить конструкцию как в примере выше в текстовый файл с расширением .css и расположить ссылку на него в секции head:  
   <link rel="stylesheet" type="text/css" href="style.css"/>

Во всех примерах выше был использован “язык”, который называется CSS - Cascading Style Sheets (каскадные таблицы стилей).

**Наследуемость и каскадирование**CSS базируется на следующих принципах:

1. [***Наследуемость***](http://htmlbook.ru/samcss/nasledovanie). Принцип, который позволяет *некоторым* CSS-свойствам, установленным у одного элемента, применяться к его потомкам. К таким свойствам относятся, например, цвет и шрифт. Важно понимать, что далеко не все свойства подчиняются этому принципу.
2. [***Каскадирование***](http://htmlbook.ru/samcss/kaskadirovanie). Принцип, которых применяется в том случае, когда какому-либо элементу HTML одновременно поставлено в соответствие более одного правила CSS, то есть, когда происходит конфликт значений этих правил. Чтобы разрешить такие конфликты, вводятся правила приоритета.
   1. Наиболее низким приоритетом обладает стиль браузера
   2. Следующим по значимости является стиль, заданный пользователем браузера в его настройках
   3. И наиболее высоким приоритетом обладает стиль, заданный непосредственно автором страницы (подробнее см. пункт “Вес селектора”)

**CSS-селекторы**Это строка, скомпонованная по определённым правилам, описывающая доступ к элементу (или их множеству) из HTML-документа для дальнейшей стилизации.  
Наболее популярны [селекторы](http://htmlbook.ru/metki/selektory) по тегу, по атрибутам (id, class), селекторы состояний (:hover, :focus, :active, :link), псевдоселекторы (::before, ::after), а также составные селекторы.  
Более подробно все виды селекторов подробно описаны [здесь](http://css.yoksel.ru/css-selectors/) и [здесь](http://css.yoksel.ru/css-selectors-part2/).

***Задания:***Написать следующие селекторы для выбора элементов в [документе](http://jsbin.com/worurenijo/edit?html,output)**:**

1. Элемент с идентификатором main
2. Все элементы с классом item
3. Все div с классом item
4. Первый элемент с классом item
5. Все элементы
6. Список ul
7. Все элементы div, которые находятся внутри списка ul
8. Все li внутри ul, не имеющие класса item
9. Все выделенные чекбоксы
10. Ссылки, по которым уже был сделан переход
11. Элементы, у которых есть атрибут data-id
12. Элементы, у которых значение атрибута data-id равно 1
13. Ссылки, которые откроются в новом окне
14. Элемент li без текста
15. Только те span, которые находятся внутри ссылки
16. Только те span, которые находятся сразу после не отмеченного чекбокса

**Вес селектора**  
Каждый селектор (в т.ч. и составной) имеет свой “вес”. Это означает, что если элементу соответствует несколько заданных CSS-правил, то применено будет только то, чей селектор имеет наибольший вес.  
[Подробнее](http://css.yoksel.ru/specifity/) о том, как рассчитывается вес.

**Отображение элемента и свойство display**Свойство [display](http://htmlbook.ru/css/display)определяет, как именно будет отображаться элемент. Оно может иметь следующие значения:

1. block. Делает элемент блочным
2. inline. Делает элемент строчным
3. inline-block. Делает элемент [строчно-блочным](http://htmlbook.ru/samlayout/blochnaya-verstka/strochno-blochnye-elementy)
4. table, table-row, table-cell и.т.д., отображает как таблицу или её элемент
5. flex. Делает все дочерние элементы резиновыми, подробнее см. главу “Flexbox”

**Плавающие элементы и свойства float и clear**Свойство [float](http://htmlbook.ru/css/float) используется для того, чтобы сделать элемент обтекаемым (текстом или любыми строчными элементами, которые идут следом за ним) по правому или левому краю.  
Свойство [clear](http://clear/) устанавливает, с какой стороны элементу запрещено обтекание другими элементами.  
Свойства float и clear активно используются в создании модульных сеток, т.к. flexbox-свойства, хотя и больше подходят для этих целей, поддерживаются только [современными браузерами](http://caniuse.com/" \l "feat=flexbox).  
Подробно о тонкостях в применении свойства float [здесь](http://htmlbook.ru/samlayout/blochnaya-verstka/plavayushchie-elementy) и [здесь](http://habrahabr.ru/post/142486/).

***Задания:***

1. Разместить на странице блок текста и картинку. Картинка должна быть прижата к правому краю, текст должен обтекать её слева.
2. Разработать 6-колоночную макетную сетку на основе свойств float и clear. В результате должны получиться классы: row, col-1, col-2, col-3, col-4, col-5, col-6.

**Flexbox**Технология, созданная специально для раскладки элементов в макете и вёрстки модульных сеток. Подробные руководства [здесь](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) и [здесь](http://css.yoksel.ru/flexbox/).

***Задания:***

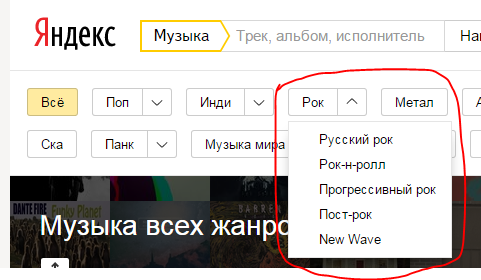
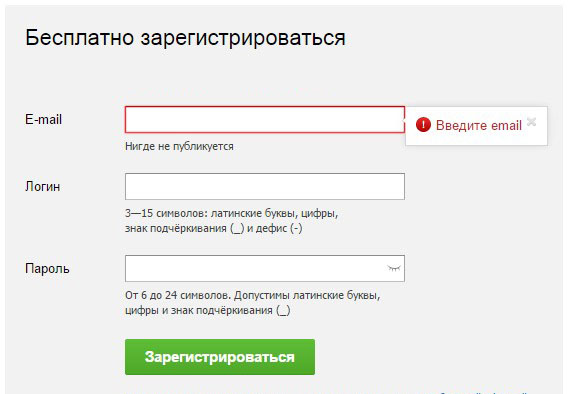
1. Разработать 6-колоночную макетную сетку на основе flexbox-свойств. В результате должны получиться классы: row, col-1, col-2, col-3, col-4, col-5, col-6.

**Позиционирование элементов и свойство position**Поток документа —  это порядок отображения элементов на странице. Нормальный поток документа предполагает следующее: блочные элементы следуют друг за другом сверху вниз, каждый на новой строке, строчные следуют слева направо и сверху вниз, переносясь на новую строку, когда в текущей не хватает места. Некоторые свойства нарушают такой порядок, например [position](http://htmlbook.ru/samlayout/blochnaya-verstka/pozitsionirovanie-elementov). У него может быть 4 значения:

1. static (по умолчанию у всех элементов). Статическое позиционирование.
2. absolute. Абсолютное позиционирование. Свойства:
   1. Выводит элемент из потока документа. Это означает, что освободившееся место займут следующие в этом потоке элементы.
   2. Свойства top, bottom, left и right задают отступ для элемента от краёв ближайшего по иерархии предка с заданным позиционированием (absolute, relative или fixed), или, если такого нет, от краёв документа.
3. relative. Относительное позиционирование. Свойства:
   1. Не выводит элемент из потока документа, т.е. место, которое он занимал, останется пустым.
   2. Свойства top, bottom, left и right задают сдвиг элемента относительно границ своего нормального положения.
4. fixed. Фиксированное позиционирование. Свойства:
   1. Выводит элемент из потока документа, как и absolute.
   2. Свойства top, bottom, left и right задают сдвиг относительно границ окна браузера

**Контекст наложения и свойство z-index**[Контекст наложения](http://css.yoksel.ru/kontekst-nalozheniya/) —  это совокупность элементов, имеющие общего родителя, перемещающиеся вместе на передний или задний план. Устанавливают порядок отображения контекстов с помощью свойства [z-index](http://htmlbook.ru/css/z-index). Некоторые CSS-свойства могут образовывать из элементов новые контексты наложения, наример, position или opacity. Подробнее [здесь](http://habrahabr.ru/post/166435/) и [здесь](http://habrahabr.ru/post/225721/).

***Задания:***

1. Создать страницу, на ней разместить элементы header и footer.
2. В header должно быть меню с произвольным числом пунктов, прижатое к правому краю. Каждый пункт —  ссылка. Количество текста в пункте может быть любое, поэтому по ширине они должны быть ограничены. Внутренние отступы у всех пунктов одинаковые, внешних нет. При наведении курсора на ссылку, цвет фона и цвет текста должен меняться.   
   Меню должно быть зафиксировано к верхнему краю. Т.е. при любом количестве контента далее на странице и при любом положении скроллбара оно должно находиться под верхним краем окна.
3. Сделать левое меню. Левое меню представляет собой блок, прижатый к левому краю документа, верхний его край находится под верхним меню, нижний край совпадает с нижним краем страницы. Меню не фиксировано, прокручивается вместе с содержимым страницы. Внутри находятся пункты, каждый пункт внутри содержит иконку и текст. Весь пункт является ссылкой.
4. Внутри на странице расположить несколько заголовков разного уровня.
5. Вслед за заголовками расположить на странице табы. Они должны выглядеть приблизительно так:   
     
     
     
     
     
   Сделать переключение вкладок и их содержимого без использования JavaScript.
6. На первой вкладке расположить блоки с произвольным количеством текста и картинкой вначале. Блоки располагаются в несколько строк и в 3 колонки. Ширина картинки равна ширине колонки. Между колонками должно быть одинаковое расстояние. Все колонки одинаковой ширины. По вертикали блоки выравниваются в строке по верхнему краю.
7. На второй вкладке расположить элемент с выпадающим меню:  
     
   Открывается меню по клику на стрелочке, как на картинке. Открываться и закрываться меню должно без использования JavaScript.
8. На третьей вкладке сверстать [прогрессбар](http://media02.hongkiat.com/beautiful-progress-bars/21-35-gorgeous-progress-bars.png) с использованием CSS-transitions (полоски должны двигаться, при изменении ширины прогрессбара он должен перемещаться плавно).
9. На четвёртой сверстать [круговой прогрессбар](http://i.stack.imgur.com/DzNnN.png).
10. Сверстать переключатель как на форме яндекс-денег. Картинки можно взять любые другие.
11. На отдельной вкладке сверстать форму. Когда пользователь нажимает кнопку «зарегистрироваться», все поля должны валидироваться, и те, которые валидацию не прошли, отмечаются обводкой и тултипом, как на картинке.