



— FRONT END —

# WEB ACADEMY

STUDY GUIDE

# Front end guideline

## Content

● <u>Чек - лист;</u>	3
● <u>Что такое ТЗ?</u>	10
● <u>Что такое проект?</u>	17
● <u>Как общаться с гуглом?</u>	18
● <u>Словарь Front end терминов.</u>	20

# Чек - лист

## Front End Web Developer Skills Checklist

- **HTML**
- **CSS**
- **JavaScript**
- **Responsive Web Design**
- **CSS Frameworks**
- **JavaScript Frameworks**
- **Version Control**
- **Web Performance**
- **Browser Development Tools**
- **Building and Automation Tools**
- **Testing**
- **Soft Skills**

## HTML

- **Semantic elements:** Семантический элемент четко описывает свое значение для браузера и девелопера. Например, такие элементы как `section` и `article` имеют четкое значение в структуре что и отличает их от элемента `div`.
- **Block-level elements:** Этот элемент занимает все пространство родительского элемента.
- **Inline elements:** Этот элемент занимает только пространство, ограниченное тегами, которые определяют встроенный элемент.
- **Forms:** Форма представляет собой раздел документа, который содержит интерактивные элементы управления для отправки информации на веб-сервер.
- **Input Types:** Элемент ввода используется для создания интерактивных элементов для веб форм, которые используются для приема данных от пользователя.

## CSS

- **Display Value Types:** Атрибут `display` позволяет управлять рендерингом графических или контейнерных элементов.
- **The Box Model:** Эта модель определяет размер прямоугольной коробки, которая представляет элемент в документе.
- **Basic Positioning:** Это свойство выбирает альтернативные правила для позиционирования элементов.
  - **Static:** Статичное позиционирование позволяет элементам использовать обычное поведение.
  - **Absolute:** Не оставляет места для элемента, вместо этого позиция находится на уровне конкретной позиции относительно ее ближайшего предка или содержащего блока.
  - **Fixed:** Не оставляет места для элемента, вместо этого указывается позиция относительно экрана просмотра.
  - **Flexbox:** Режим компоновки, обеспечивающий расположение элементов на странице так, чтобы элементы вели себя предсказуемо при различных размерах экрана и отображения на различных устройствах.
  - **Float:** Указывает, что элемент должен быть взят из нормального потока и расположится слева или справа.

- **Font Styling and Web Fonts:** Стиль шрифта позволяет изменять появление текста, веб-шрифты позволяют загружать веб-базированные шрифты, которые пригодны для использования всеми клиентами.
- **Backgrounds:** Фоны позволяют определить цвет или изображение, которое будет использоваться как фон контейнера.
- **Pseudo-selectors:** Псевдоэлементы позволяют выбирать гипотетические элементы, которые существуют вокруг элементов, которые вы определили в своем HTML.
- **Animations and transitions:** Анимации и переходы позволяют вам анимировать элементы или определять переход между двумя состояниями элемента.

## JavaScript

- **Syntax:** Общие правила определяющие, как набирается язык.
- **Data types:** Различные типы переменных, которые поддерживает язык (например, строки и целые числа).
- **Functions:** Блок кода, предназначенный для выполнения конкретной задачи.
- **Object Literals:** Все в JavaScript – это объект, но определение своих собственных литералов может упростить ваш код.
- **Object-Oriented Programming:** JavaScript предоставляет несколько способов реализовать объектно-ориентированное программирование, включая функциональное, прототипное и псевдоклассическое.
- **Design Patterns:** Шаблон проектирования является многоразовым решением обычной проблемы.
- **AJAX:** AJAX предоставляет возможность асинхронного запроса данных из Интернета, без необходимости перезагрузки страницы.
- **jQuery:** Чрезвычайно популярная библиотека, которая делает кросс-браузерный DOM манипулятивным, обработку событий и AJAX намного проще.

## Responsive Web Design

- **@media queries:** Медиа запросы позволяют создавать презентацию контента только для определенного диапазона устройств без необходимости изменять сам контент.
- **Relative sizing units:** CSS предоставляет множество других единиц измерения кроме пикселей. Например, em, rem, vw, vh, vmin.

### CSS Frameworks

- **Bootstrap:** Это CSS-структура, первоначально разработанная в Twitter, которая значительно упрощает создание адаптивных проектов.
- **Foundation:** Еще одна структура CSS, разработанная Zurb, которая также упрощает создание гибких конструкций.

## JavaScript Libraries and Frameworks

- **AngularJS:** Поддерживает двустороннюю привязку данных и позволяет расширять HTML словарь для создания интерфейсных веб-приложений.
- **EmberJS:** Устраняет необходимость в шаблонном коде, применяя использование строгого файла и соглашения об именах объектов.
- **KnockoutJS:** Упрощает создание приложений, управляемых данными с помощью системы декларативной привязки.

## Version Control

- **Git:** Это система управления распределенной версией.
  - **clone**
  - **add**
  - **commit**
  - **push**
  - **pull**
  - **branch**
  - **log**
- **GitHub:** Это веб-хостинг для репозитория Git, который предлагает другие функции для поддержки сотрудничества между разработчиками.
  - **forking**
  - **pull request**

## Web Performance

- **Critical rendering path:** Критический путь рендеринга – это процесс, в котором браузеры используются для преобразования HTML, CSS и JavaScript в фактические пиксели, которые отправляются на экран пользователя.
- **Image optimization:** Это процесс использования правильного типа изображения для контента изображения, а также удаление дополнительных метаданных в файле изображения.
- **JavaScript minification:** Минимизация JavaScript – это процесс удаления ненужных символов из JavaScript файлов, чтобы уменьшить их размер.

## Browser Developer Tools

- **Element inspecting:** Панель элементов, которая позволяет видеть все в одном DOM дереве и позволяет проверять и редактировать элементы DOM на лету.
- **Network:** Сетевая панель записывает информацию о каждой сетевой операции в вашем приложении.
- **Timeline:** Панель временной шкалы позволяет записывать и анализировать всю активность в вашем приложении во время его запуска.
- **Application profiling:** Панель профилирования позволяет вам наблюдать за использованием памяти вашего приложения, во время его работы.
- **Resources:** Панель ресурсов позволяет вам проверять загружаемые ресурсы в ваше приложение.

## Building and Automation Tools

- **npm:** npm – это менеджер пакетов, по умолчанию для Node.js. Он является фреймворком для большинства инструментов автоматизации.
- **Grunt:** Grunt — это инструмент для сборки javascript проектов из командной строки с использованием задач.
- **Gulp:** Gulp — это инструмент сборки веб-приложения, позволяющий автоматизировать повторяющиеся задачи, такие как сборка и минификация CSS- и JS-файлов, запуск тестов, перезагрузка браузера и т.д. Тем самым **Gulp** ускоряет и оптимизирует процесс веб-разработки.
- **Bower:** Bower – это самый популярный менеджер для клиентского javascript. Все, чем занимается Bower — устанавливает пакеты вместе с их зависимостями.
- **Yeoman:** Yeoman является инструментом скаффолдинга — с плагинами, известными как генераторы, Yeoman определяет как генерировать структуру проекта.



## Testing

- **Mocha:** Это среда тестирования JavaScript, которая работает в Node.js и браузер для асинхронного тестирования.
- **Jasmine:** Это фреймворк для тестирования JavaScript с открытой исходной средой.

## Soft Skills

- **Strong communication:** Передавать цели, прогресс и проблемы в более широкие команды, включая руководство, сверстников и клиентов.
- **Agile problem solving:** Понимать вопрос между строк. Разбивать проблему на более мелкие легко решаемые части.
- **Healthy Passion:** Будьте в курсе последних достижений в области.

**Self-Starting Motivation:** Экспериментируйте и исследуйте.

# Что такое ТЗ?

## Техническое задание

Техническое задание — исходный документ на проектирование технического объекта (изделия). ТЗ устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.

## ОСНОВНЫЕ РАЗДЕЛЫ ТЕХЗАДАНИЯ

На разработку сервиса, крупного сайта, спецификация на онлайн-каталог, тех. задание на интернет-магазин в целом при правильной проработке должны будут содержать один и тот же минимальный набор разделов.

### Общая информация о проекте

- Концепция продукта – потребность или проблема, которую продукт решает, общая логика решения, целевая аудитория;
- Цели и задачи проекта – кратко описать, какие есть конкретные цели проекта в цифрах (KPI): достичь трафика 10 000/мес, увеличить продажи на 20% и так далее;
- Словарь терминов, использованных в ТЗ – названия особых систем клиента, технические термины, у которых в контексте данного техзадания может быть свое значение;
- Перечень документов, на основании которых создается проект – ссылки на внешние документы, прототипы, исходники дизайна, для того чтобы не искать по телу документа;
- Карта разделов или страниц проекта – в формате дерева или хотя бы просто иерархического списка для того чтобы было сразу понятен объем разрабатываемого ресурса.

## Прототип и дизайн

- Прототип или мокапы – для любого современного интернет-проекта разработка прототипа или мокапов является необходимым и очень ускоряет процесс и точность оценки;
- Дизайн – если уже готов на данный момент, то прикрепить ссылку на исходные файлы макетов отдельным архивом;
- Описание динамики страниц – как должен реагировать интерфейс при нажатии на различные элементы управления, эффекты, появление всплывающих окон и подсказок и т.п. Обычно эту часть клиенты описывают легче всего;
- Требования к мобильной или адаптивной версии – на каких устройствах предполагается работа с этим проектом – платформы, браузеры.

## Функциональная часть проекта

- Информационная архитектура проекта должна быть описана структура сущностей проекта – сущностей базы данных, объекты системы, основные функции ядра, роли пользователей проекта;
- Функциональная спецификация под прототип – помимо динамики страниц, описанной выше, должно быть описано какие алгоритмы или какие функции ядра системы вызываются;
- Описание бэк-офиса – функционал для администратора, контент-менеджера и других – как будет построено администрирование и наполнение проекта;
- Интеграции с внешними и внутренними системами – какие данные передаются, куда, в каком виде, например, если используется API сторонних систем, то как именно;
- Тестирование ресурса – на каких устройствах, платформах, браузерах и при каких условиях будет тестироваться проект;
- Требования по безопасности ресурса – общие или специальные требования к безопасности;
- Требования по нагрузке и серверам – какую нагрузку должен выдерживать проект и на каких серверах размещаться.

Это как раз самая техническая часть проекта, которую в большинстве случаев клиент не может описать самостоятельно. Её нужно писать вместе с профильными специалистами.

## Организационная часть проекта

- Роли в проекте со стороны Заказчика и Исполнителя, и распределение ответственности – кто, поименно, участвует в проекте с обеих сторон, кто за что отвечает.
- Идеология проекта – общее описание основных принципов, которых и Заказчик, и Исполнитель договорились придерживаться на проекте
- Порядок приемки работ – как будут приниматься работы, как трактуются неточные формулировки, порядок исправления ошибок и недочетов
- Порядок запуска проекта – как конкретно, в какое время и при соблюдении каких условий происходит запуск. Какой процесс запуска проекта и как это синхронизировано с рекламой и другими сферами деятельности проекта.

Структура ТЗ	
Общие разделы	Описание(Описание к пунктам разделов)
Содержание документа	содержание и иерархическая структура документа
Концепция продукта	2-3 абзаца текста: потребность/проблема, которую он решает, общая логика решения, целевая аудитория
Цели и задачи ресурса	коротко описать, какие есть конкретные цели ресурса в цифрах: достичь трафика 10 000/мес, увеличить продажи на 20%
Словарь терминов в ТЗ	названия особых систем клиента, технические термины, у которых в контексте данного ТЗ может быть свое значение
Перечень документов, на основании которых создается ресурс	ссылки на внешние документы, прототипы, исходники дизайна
Карта страниц и разделов ресурса	в виде mindmap или иерархического списка

<b>Организационная часть проекта</b>	
<b>Роли в проекте: со стороны Заказчика и Исполнителя</b>	Минимумальный набор это проектные менеджеры со стороны Заказчика и Исполнителя и лицо, принимающее работы по проекту
<b>Идеология проекта</b>	фиксируется объем работ и тип соглашения (fixed price, time&materials), принципы управления (график работ и их протекание), а также работа с правками и порядок их внесения
<b>Распределение ответственности</b>	информация о том, за что проектный менеджер Исполнителя несет ответственность и за что нет, а также ответственность Заказчика
<b>Порядок приемки работ</b>	описывается, кто передает работы Заказчику и кто со стороны Заказчика их принимает, а также правила приемки
<b>Порядок запуска проекта (оргчасть после него)</b>	описываем, в котором часу пускаем сайт в мир, кто участвует в процессе запуска и что требуется от каждого из участников для успешного старта
<b>SEO-требования</b>	
<b>Правила генерации URL</b>	описания правила создания URL для страниц подкаталогов и категорий
<b>Правила автогенерации meta tags: title, description, h1</b>	из каких тегов в разметке страницы нужно брать данные, чтобы автоматически генерировать данные мета-теги
<b>Автогенерация карты сайта (sitemap.xml)</b>	частота обновления карты сайта авто. методом, не описан ее максимальный объем
<b>Микро-разметка (опционально)</b>	
<b>Правила наполнения/переноса контента (опционально)</b>	таблицы с мета-тегами и URL всех страниц, которые нужно перенести со старого сайта

<b>Прототип и дизайн</b>	
<b>Прототип или мокапы</b>	ссылка на прототип, мы рекомендуем динамические прототипы в Axure
<b>Дизайн</b>	если уже готов на данный момент, то прикрепить ссылку на исходные файлы макетов отдельным архивом
<b>Описание динамики страниц</b>	как должен реагировать интерфейс при нажатии на различные элементы управления, эффекты, появление всплывающих окон и подсказок
<b>Требования к мобильной или адаптивной версии</b>	на каких устройствах предполагается работа с этим проектом – платформы, браузеры
<b>Функциональная часть проекта</b>	
<b>Информационная архитектура проекта</b>	должна быть описана структура сущностей проекта – сущностей базы данных, объекты системы, основные функции ядра, роли пользователей проекта
<b>Функциональная спецификация под прототип</b>	помимо динамики страниц, описанной выше, должно быть описано какие алгоритмы или какие функции ядра системы вызываются
<b>Описание бэк-офиса</b>	функционал для администратора, контент-менеджера и других – как будет построено администрирование и наполнение проекта
<b>Интеграции с внешними и внутренними системами</b>	какие данные передаются, куда, в каком виде, например, если используется API сторонних систем, то как именно
<b>Тестирование ресурса</b>	на каких устройствах, платформах, браузерах и при каких условиях будет тестироваться проект
<b>Требования по безопасности ресурса</b>	Общие или специальные требования к безопасности
<b>Платформа (движок) ресурса</b>	тип платформы (CMS/framework) и требования к версии платформы и компонентам (опционально для сайтов и магазинов)
<b>Требования к серверам и хостингу</b>	CPU, RAM, HDD, интерфейсы доступа и сертификаты (SSH, FTP/SFTP, SSL), ОС на сервере, окружение, системы, расширения, которые должны быть установлены (apache/nginx/iis, и тп)

# Пример оформления ТЗ

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ на разработку туристического сайта сайта «\_\_\_\_\_»

### 1. ЭТАПЫ И СРОКИ СОЗДАНИЯ САЙТА:

Общий срок работ по созданию представительского сайта «\_\_\_\_\_», составляет 45 дней (30 дней – создание оригинал-макетов, флэш-заставки, вертка шаблона и разработка полного статического прототипа сайта + 15 дней программирование динамики и подключение системы управления контентом).

Разработка концепции сайта, информационное проектирование, оформление Технического задания,

Разработка эскиза базового дизайна,

Создание работающего шаблона сайта, включая полную разработку «визуала», ссылки, интерактивные элементы, флэш-заставку, музыкальное сопровождение флэш-ролика для первой страницы сайта.

Программирование на базе системы управления контентом

Подготовка, редактирование, верстка и настройка контента сайта под поисковые системы,

Публикация сайта на сервере, тестирование, прописка в поисковых системах и каталогах Yandex, Rambler, Aport!

### 2. ТЕХНОЛОГИЧЕСКИЕ ТРЕБОВАНИЯ:

2.1. Сайт разрабатывается под базовое разрешение экрана 800x600 пкс,

2.2. Корректное отображение броузерами Internet Explorer, Netscape Navigator версии 4.0 и выше,

2.3. Использование фирменных цветов и логотипа компании,

2.4. Обязательная визуальная поддержка действий пользователя – т.н.

«интерактив» (визуальное отображение активных, пассивных и посещенных ссылок; четкое обозначение местонахождения пользователя). По ссылке с каждой страницы загружается почтовая программа (бланк письма для обратной связи).

2.5. Мета-теги и контент сайта на этапе изготовления сайта д.б. настроены для поисковых систем, что обеспечить продвижение сайта по ключевым словам в поисковых системах Yandex, Aport!, Rambler.

### 3. СТРУКТУРА И ОПИСАНИЕ САЙТА:

#### 3.1. Главная страница (обложка сайта):

Содержит графическую часть, флэш-презентацию, основную навигацию, внутреннюю навигацию раздела «Каталог», а также контентную область для того, чтобы посетитель сайта с первой страницы мог получить вводную информацию «О компании» и «Каталог», а также ознакомиться с последними новостями компании..

Главная страница оформляется коллажем с использованием фирменных цветов. Графическая часть и флэш-презентация отображается в виде широкой горизонтальной шапки, внизу содержит основную навигацию сайта.

В дизайне отдельно предусматриваются ссылки на **список туров и спецпредложения**.

Внутренняя навигация раздела «Каталог» отображается в виде списка в левой части экрана под шапкой в виде колонки в текстовом виде для последующей легкой модификации данной области.

Между шапкой и контентной областью предполагается **«баннерная крутилка»**. Атрибуты:

Название баннера

Картинка

Ссылка на ресурс

Количество показов

Отображать на странице

Если «количество показов» больше установленной цифры в базе, галочка «отображать на странице» автоматически снимается

Контентная область первой страницы делится на разделы:

«**О компании**» - содержит краткую информацию раздела со ссылкой », ведущую на раздел «О компании».

«**Новости**» - содержит 3 последние новости (анонсы) компании в формате: дата, заголовок, краткое содержание. Ссылка с заголовка новости переходит на страницу с полным содержанием данной новости (анонса).

«**Каталог**» - верхние разделы каталога отображаются в виде списка в формате: заголовок, маленькая картинка, краткое содержание. Ссылки с заголовка и картинки ведут на полное описание верхних разделов каталога.

Внизу страницы отображается облегченная навигационная панель в текстовом виде, Copyright, контактный e-mail.

#### 3.2. Графическая оболочка внутренних страниц (общая для всех подразделов):

Графическая шапка с горизонтальной навигацией может содержать ненавязчивые флэш-элементы (например, анимированный логотип, являющийся одновременно ссылкой на основную страницу сайта), но не должна отвлекать от информационного наполнения страницы. Желательна разработка нескольких вариантов графической шапки в привязке к основным разделам каталога. Для разделов основной навигации разрабатывается общая шапка.

Внутренняя навигация раздела «Каталог» отображается в виде списка в левой части экрана под шапкой в виде колонки в текстовом виде для последующей легкой модификации данной области.

В дизайне отдельно предусматриваются ссылки на **список туров и спецпредложения**.

Между шапкой и контентной областью предполагается **«баннерная крутилка»**.

Внизу страницы отображается облегченная навигационная панель в текстовом виде, Copyright, контактный e-mail, ссылка на «Основную страницу» сайта.

### 3.3. Описание контента разделов сайта:

#### Основная навигация:

**О компании** – текст + фото – 1 страница

**Новости.** На странице отображается список новостей в формате: дата новости, заголовок, содержание. Последние новости всегда добавляются в верх списка.

**Услуги** – текст + фото – 1 страница

**Отзывы** – список в формате: дата, имя, e-mail, текст отзыва

**Отправить отзыв** – форма с полями: имя, e-mail, url, телефон, город, сообщение

**Вопросы и ответы** – список в формате: текст вопроса, текст ответа.

**Задать вопрос** – форма с полями: имя, e-mail, url, телефон, город, сообщение

**Контакты** – контактная информация + mail-форма с полями: имя, e-mail, url, телефон, город, сообщение

**Поиск** – стандартная форма с поиском информации по сайту

#### Туры:

**Спецпредложения** – список туров из базы данных с пометкой «спецпредложения», вывод информации в табличной форме по аналогии с разделом «туры».

**Туры** – список туров с возможностью поиска в соответствии с заданными параметрами.

Первая страница раздела «Туры» оформляется в виде формы с выпадающими полями:

Вид тура  
Подвид тура  
регион  
месяц

Спецпредложения (отмечается галочкой)

В зависимости от данных, введенных в поле «Вид тура», предусматривается модификация данных в выпадающих списках «подвид тура», «регион» и «месяц» в зависимости от данных, имеющихся по этому виду в базе туров. После выбора условий поиска нажимаем кнопку «искать»

В результатах поиска отображается таблица с полями:

Вид тура  
Подвид тура  
Месяц  
Регион  
Название тура  
Краткий комментарий

Если это спецпредложение, сделать пометку в таблице напротив данной строки.

По ссылке на заголовок тура пользователь переходит на страницу с полным описанием конкретного тура. С шаблоне полного описания тура вверху и внизу страницы предусматриваются ссылки «Заказать тур». По этим ссылкам пользователь переходит на страницу с mail-формой, где он может оформить заявку на выбранный им тур. Данные по выбранному туру автоматически передаются в поля формы. Пользователю остается только заполнить поля, относящиеся к его личным данным и комментарий.

Итого - база данных туров будет иметь следующие атрибуты:

Вид тура  
Подвид тура  
Месяц  
Регион  
Название тура  
Краткий комментарий  
Полное описание (html-формат)  
Спецпредложение  
Отображать на странице

**Заказать тур** – отдельная форма в формате: имя, телефон, e-mail, адрес, вид тура, желательное время, дополнительные комментарии. Со страницы идет дополнительная ссылка на раздел «Туры» с пометкой приблизительно такого содержания: «Вы также можете выбрать тур из нашей базы данных и оформить заказ на выбранный Вами тур»

#### Каталог (требуется структура полного дерева – иерархии каталога):

**Первый уровень каталога** отображается в левом меню в виде вертикальной навигации. По ссылке на каждый раздел отображается страница с полным описанием.

**Элементы второго уровня каталога (виды)** отображаются в виде списка под полным описанием в формате: название, маленькая картинка, краткое описание. С названия и маленькой картинки идут ссылки на полное описание элемента второго уровня.

**Элементы третьего уровня (регионы)** отображаются в виде списка под полным описанием в формате: название, маленькая картинка, краткое описание. С названия и маленькой картинки идут ссылки на полное описание элемента третьего уровня.

С каждого уровня страниц каталога идет дополнительная ссылка «Посмотреть туры», генерирующая запрос к Базе Данных туров в соответствии с разделом, в котором в данный момент находится посетитель.



# Что такое проект?

Проект - это документ или набор документов, являющийся основным результатом (выходом) процесса проектирования. Процесс проектирования в свою очередь присутствует в том или ином виде абсолютно во всех серьезных методологиях создания ПО.

Результатом данного курса есть создание и защита собственного проекта.

Примеры проектов, которые можно создавать, доступны по ссылке:

[https://drive.google.com/drive/folders/0B8L\\_Kt5x\\_qkDT21tU1NKU1NSc2s?usp=sharing](https://drive.google.com/drive/folders/0B8L_Kt5x_qkDT21tU1NKU1NSc2s?usp=sharing)

<https://all-psd.ru/site-templates/>

<http://psd-html-css.ru/shablony/psd-shablony-saytov>

<http://freepsd.ws/>

<http://mainview.ru/download/templates/besplatnye-shablony-sajtov-v-psd>

<https://camouf.ru/designes/ready-psd-layouts-for-saito/>

<https://freelance.today/poleznoe/15-besplatnyh-psd-maketov-saytov.html>

# Как общаться с гуглом?

В данном курсе, помимо материала который дает тренер, Вам нужно будет искать необходимую информацию самому.

К сожалению, поисковая система не умеет читать мысли пользователей. Поэтому важно правильно сформулировать запрос, чтобы результаты поиска вполне удовлетворили ваши информационные потребности.

Советы о том, как правильно «гуглить»:

1. Чтобы найти точную фразу или форму слова — возьмите ее в кавычки.

«дивлюсь я на небо»

2. Чтобы найти слово из цитаты, которое вы забыли — поставьте всю цитату в кавычки, а пропущенное слово отметьте звездочкой. Готово!

«Мама \* давай»

3. Как найти любое слово из нескольких перечисленных. Надо прописать все варианты через вертикальный слэш |.

гостиница | отель | хостел

4. Чтобы найти слова в пределах одного предложения, используйте ampersand — знакомый всем значок &.

Лучшие музеи мира & Париж

5. Если нужно найти документ с определенным словом, поставьте перед ним +, только без пробела.

Николаевский зоопарк +тигр

6. Иногда необходимо исключить слово из поиска. Для этого перед нужными словами поставьте минус.

Наполеон -торт

7. Ища информацию на определенном сайте, вставьте слово «site» и поставьте после него двоеточие.

Конституція України site:rada.gov.ua (без пробела)

8. Чтобы найти определенный тип документа, впишите оператор «mime», поставьте двоеточие и тип нужного документа.

заявление на загранпаспорт mime:pdf

9. С помощью оператора «lang» можно искать информацию на определенном языке. После него — поставить двоеточие и написать язык. Например, если это русский — то поставить ru, английский — en и т. д.

void glClearColor lang:ru

10. Для поиска синонимов используйте символ ~ перед словом. Таким образом откроются все ссылки на страницы со словами, похожими на ваше, при этом не содержащие это же слово.

~необычные шарфы -необычные

11. Как прогуглить значение слова? Ввести в поиск define: и узнать.

define: коучинг

12. Если вам надо узнать величину или курс валют, просто введите запрос в строке.

1 кг в фунтах

13. Точное время по городу найдется с помощью слов «время» и «город» в строке поиска.

time Киев

14. Калькулятор. Чтобы подсчитать уравнение, вбейте пример в поиск и получите результат.

695+583

15. Прогноз погоды по городу на несколько дней найдется при вводе слов «weather» и «город».

weather Киев

Также для поиска необходимой информации можно использовать такие ресурсы, как :

<https://stackoverflow.com>

<https://community.web-academy.com.ua>

где Вам обязательно дадут правильный ответ на поставленный вопрос!

# Словарь Front end терминов

0-9

## 9-slice scaling

**9-дольное масштабирование** (неискажающее масштабирование), техника, позволяющая изменять размер объекта без геометрических искажений. Объект делится на 9 долей (3 строки и 3 колонки), каждая из которых масштабируется по собственным правилам: углы не масштабируются совсем, центральная доля масштабируется по высоте и по ширине, оставшиеся доли масштабируются либо только по ширине, либо только по высоте.

**A**

## accessibility, a11y

**доступность**, возможность использования интерфейса всеми, независимо от физических или технических ограничений.

## adaptive design

**адаптивный дизайн**, подход к дизайну сайтов, учитывающий их адаптацию под различные устройства и условия, противопоставляется распространённому «фиксированному» дизайну только для компьютеров. Один из способов создания адаптивного дизайна — методики отзывчивого дизайна.

## almost standards mode

**почти стандартный режим**, режим интерпретации и отображения страницы браузером, в котором браузер целенаправленно отходит от стандартов (как при режиме совместимости) лишь для отдельных особых случаев, используется для страниц с переходными доктайпами HTML 4 и XHTML 1.

## anchor

1. **ссылка**, гиперссылка между двумя документами.
2. **якорь**, ссылка на элемент в пределах документа, может быть частью гиперссылки.

## animation

**анимация**, плавное изменение визуальных параметров объекта.

## asset

**ресурс**, составная часть сайта: текстовые файлы, графика, видео, базы данных и др. В узком смысле «статический ресурс»: стили, скрипты, оформительская графика, в противоположность динамическому содержанию.

## assert

**проверка**, условие, которое проверяется, употр. в тестировании, напр. *field assert has failed* — проверка поля не прошла.

## at-keyword

**ключ директивы.**

## at-rule

**директива**, тип объявления, начинающегося со знака @, напр. *@import*.

## attribute

**атрибут**, напр. *rel="stylesheet"*.

- attribute name — **название атрибута**
- attribute value — **значение атрибута**

## B

## back-end

## бэкэнд

1. область веб-технологий, работающих на сервере.
2. внутренняя часть серверной системы, занимается обработкой данных.

## background

**фон**, напр. *background image* — *фоновое изображение*.

- background-color — **цвет фона**
- background-position — **позиция фона**
- background-size — **размер фона**
- background-repeat — **повторение фона**
- background-origin — **границы фона**
- background-clip — **обрезка фона**
- background-attachment — **закрепление фона**
- background-image — **фоновое изображение**

## baseline

**базовая линия**, воображаемая линия в основании букв в строке.

## blending mode

**режим наложения**, алгоритм смешивания слоёв, применяемый в графических редакторах или прямо в CSS, напр. *overlay blending mode* — *режим наложения перекрытием*.

## block

1. **блок деклараций**, структурная часть объявления. Содержится в фигурные скобки и содержит декларации свойств (или, в случае медиавыражения, блоки стилей).
2. **блочный вид**.

## blockquote

**блочная цитата**, HTML-элемент для выделения цитируемой *нестрочной* области документа. Может содержать атрибут cite со ссылкой на первоисточник.

## border

**рамка**, внешняя видимая часть блока, напр. *border-left* — *рамка слева*.

- border-collapse — **слияние рамки**
- border-color — **цвет рамки**
- border-image — **графическая рамка**
- border-image-outset — **выступ графической рамки**
- border-image-repeat — **повторение графической рамки**
- border-image-slice — **разбивка графической рамки**
- border-image-source — **адрес изображения для графической рамки**
- border-image-width — **толщина графической рамки**
- border-radius — **скругление рамки**
- border-style — **стиль рамки**
- border-width — **ширина рамки**

## border-radius

**скругление рамки**, напр., *border-radius:3px* — *скругление рамки на 3 пикселя*, *border-top-left-radius* — *скругление верхнего левого угла*.

## bounding box

**окаймляющий блок**

## box model

**блочная модель**, система размерностей блока, состоит из содержимого, отступа, рамки и внешнего отступа.

## box-shadow

**тень (блока)**, напр. *box-shadow is red* — *тень блока красная*.

## breadcrumbs

**хлебные крошки**, вспомогательная навигация на сайте, последовательно отражающая структуру или пройденные шаги.

## breakpoint

1. **точка останова**, строка в коде, на которой выполнение скрипта приостанавливается для вызова отладчика.
2. **контрольная точка**, условие, при котором раскладка сайта меняется с одной на другую. Характерно для медиавыражений.

## browser

## браузер

## button

## кнопка

1. элемент интерфейса, при нажатии отправляющий форму или производящий иное действие
2. элемент формы `<button>` или `<input type="button">` (также `submit` и `image`).

## C

## callback

**колбэк**, исполнение кода по завершению работы функции, напр. *pass value in a callback* — *передай значение в колбэке*.

## candidate recommendation (CR)

**кандидат в рекомендации**, статус спецификации W3C, один из возможных вариантов для предложенной рекомендации.

## cascade

**каскад**, описанная в спецификации CSS схема, определяющая порядок применения к элементу блоков стилей — в зависимости от их источника, важности, специфичности и положения в коде. Ошибочно «каскадом» иногда называют селектор потомка.

## cell

## ячейка

1. структурная единица таблицы.
2. элемент таблицы `<td>` или `<th>`.

## cellpadding

**отступ в ячейках** таблицы.

## cellspacing

**расстояние между ячейками** таблицы.

## character reference

**символьная подстановка**, напр. `&nbsp;`. Допустим также не совсем точный, но более устоявшийся перевод **спецсимвол**.

## checkbox

## чекбокс

1. элемент интерфейса для выбора одного или нескольких вариантов, напр. *checkbox is disabled* — *чекбокс неактивен*.
2. элемент формы `<input type="checkbox">`.

## child

**потомок**, (во мн. ч. `children` — **потомки**).

## classitis

**классянка**, выдуманное заболевание, для которого характерно чрезмерное использование множественных классов: как для оформления, так и для хранения информации.

**отмена обтекания**, CSS-свойство, запрещающее блоку обтекать предшествующие плавающие блоки.

- `clear:left` — **отмена обтекания слева**
- `clear:right` — **отмена обтекания справа**
- `clear:both` — **отмена обтекания с обеих сторон**

## clearfix

**клиарфикс**, приём для предотвращения схлопывания контейнеров с плавающими блоками. Заключается в создании специальной распорки, которая располагается после плавающих блоков и отменяет их обтекание. Частный случай решения проблем с флоатами.

## click

**клик**, нажатие на объект с помощью курсора мыши или подобного устройства, напр. *double click is firing event* — *двойной клик вызывает событие*.

## client-side

**клиентсайд**, область технологий, работающих в браузере, часто синоним фронтенда.

## clipping path

**контур обрезки**

## closure

**замыкание**, функция, содержащая в теле ссылки на переменные, объявленные в окружающем коде.

## code

1. сущ. **код**, неисчисл. (как сахар), напр. *error in code* — *ошибка в коде*, но не *ошибка в кодах* (неправильно).
2. гл. **писать код, верстать, программировать**, напр. *to code a site* — *верстать сайт*.

## colspan

**объединение столбцов**, HTML-атрибут для объединения нескольких ячеек таблицы в одной строке.

## column

**столбец, колонка**, вертикальная область в таблице, раскладке или тексте.

## color

**цвет**

### combinator

**комбинатор**, специальный символ, использующийся в сложном селекторе для точного указания комбинации DOM-элементов, напр. >, + и др. Пробел между селекторами также считается комбинатором.

- descendant combinator — **комбинатор потомка**, (пробел).
- child combinator — комбинатор непосредственного потомка, >. Также называется дочерним.
- adjacent sibling combinator — **комбинатор непосредственного соседства**, +.
- general sibling combinator — **комбинатор соседства**, ~.

### comment

**комментарий**, пояснение к коду

### compositing

### композитинг

### composite layer

### композиционный слой

### compositor

**композитор** (в знач. «движок композитинга»)

### content

### содержимое

- metadata content — **метасодержимое**
- flow content — **потокосодержимое**
- sectioning content — **структурное содержимое**
- heading content — **заголовочное содержимое**
- phrasing content — **текстовое содержимое**
- embedded content — **внедряемое содержимое**
- interactive content — **интерактивное содержимое**
- palpable content — **содержимое для вывода**
- script-supporting content — **скриптовое содержимое**
- grouping content — **группирующее содержимое**

### cross-browser

**кроссбраузерный**, работающий во всех браузерах, слитно.

### cross-platform

**кроссплатформенный**, работающий на всех платформах, слитно.

### CSS Working Group (CSSWG)

### Рабочая группа CSS

### custom

**кастомный** (возм. пользовательский)

- custom element — **кастомный элемент**.
- custom attribute — **кастомный атрибут**.

### D

### data binding

**связывание данных**, автоматическая синхронизация данных между двумя сущностями, обычно моделью и представлением.

### declaration

**декларация**, объявление свойства и значения.

### deep comparison

**глубокое сравнение**, сравнение двух объектов, при этом производится сравнение структуры объектов и проверяется равенство примитивов содержащихся внутри объектов.

### deploy, deployment

**деплой, развёртывание, выкладка, выкатка**, публикация изменений либо напрямую на публичный сервер (продакшен), либо через промежуточные системы и процессы автоматической сборки.

### design

1. сущ. **дизайн**, визуальный облик сайта, может включать принципы взаимодействия и архитектурные решения, напр. *new design for a site* — **новый дизайн для сайта**.
2. гл. **разрабатывать**, создавать или проектировать внешний вид и функциональность.

### dirty checking

**проверка на «грязные» свойства объекта**, то есть глубокое сравнение свойств модели, который приводит к обратному вызову, если что-то изменилось.

### disable

### деактивировать

### disabled

**неактивный**, состояние элемента формы, в котором его изменение запрещено, напр. *disabled button* — **неактивная кнопка**.

### display

1. **вид**, css-свойство, определяющее тип отображения блока.
  - display:none — **скрытый вид**
  - display:block — **блочный вид**
  - display:inline — **строчный вид**
  - display:inline-block — **вид строчного блока**
  - display:flex — вид флекс-контейнера
  - display:inline-flex — вид строчного флекс-контейнера
2. **экран**, напр. *экран смартфона, разрешение экрана*.

### divitis

**диватоз**, выдуманное заболевание, для которого характерно использование только элементов <div> вместо семантически подходящих.

### DOM

**объектная модель документа, дом**

### doctype

**докта́ип**, конструкция вида <DOCTYPE html> в начале документа, которая обозначает используемую версию HTML. Используется браузерами для выбора режима (стандартного, почти стандартного или совместимости).

### dropdown

**выпадающее меню**, элемент интерфейса, раскрывающий дополнительную информацию при активации.

## E

### easing

**плавность, функция плавности**, описывает изменение скорости анимации с помощью ключевых слов или кривых Безье, напр. *transition easing is linear* — *плавность перехода линейная*.

### editor's draft (ED)

**редакторский черновик**, ранний статус спецификации W3C, предложенный одним из редакторов. На основании редакторского черновика может быть составлен рабочий черновик.

### element

#### элемент

- void elements — **пустые элементы**
- raw text elements — **элементы со свободным текстом**
- escapable raw text elements — **элементы с экранируемым свободным текстом**
- foreign elements — **сторонние элементы**
- normal elements — **обычные элементы**

### em

**ем**, единица измерения в CSS, которая зависит от размера шрифта родителя, напр. *font-size: 3em* — *размер шрифта 3 ема*.

### enable

#### активировать

#### enabled

**активный**, состояние элемента формы, в котором его изменение разрешено, напр. *enabled button* — *активная кнопка*.

### engine

**движок**, напр. *Safari is based on WebKit engine* — *Сафари работает на движке Вебкит*.

## F

### fallback

**фолбэк**, запасной вариант, дополнительная реализация на случай отсутствия в браузере нужных возможностей, вроде включённых скриптов или нужных видеокодеков, напр. *fallback to a link* — *сфолбэчить на ссылку*.

### favicon

**фавиконка**, значок сайта, обычно 16×16 пикселей в формате ICO.

### fieldset

**группа полей**, элемент, объединяющий несколько полей формы, напр. *green border on a fieldset* — *зелёная рамка на группе полей*.

### figure

**фигура**, элемент из HTML5, содержащий изображение, видео, таблицу, график или фрагмент кода.

### figcaption

**заголовок фигуры**, заглавие или подпись к фигуре.

### filter

#### фильтр

- filter: brightness — **яркость**
- filter: contrast — **контрастность**
- filter: grayscale — **бесцветность**
- filter: sepia — **сепия**
- filter: invert — **инверсия**
- filter: saturate — **насыщенность**
- filter: hue-rotate — **поворот цвета**
- filter: blur — **размытость**
- filter: opacity — **непрозрачность**
- filter: drop-shadow — **тень**

## flexbox

### флексбокс, механизм раскладки.

- flex container — **флекс-контейнер**
- flex item — **флекс-элемент**
- flex-direction — **направление главной оси**
- flex-wrap — **перенос флекс-элементов**, определяет, является ли контейнер однострочным или многострочным
- flex-basis — **базовый размер** флекс-элемента по главной оси
- flex-grow — **коэффициент растягивания**
- flex-shrink — **коэффициент сжатия**
- main axis — **главная ось**, направление, в котором флекс-элементы следуют друг за другом в контейнере
- cross axis — **поперечная ось**, направление, перпендикулярное главной оси
- main size — **главный размер**, размер флекс-элемента по главной оси
- cross size — **поперечный размер**, размер флекс-элемента по поперечной оси
- justify-content — **распределение флекс-элементов** по главной оси
- align-items — **выравнивание флекс-элементов** по поперечной оси
- align-self — **выравнивание флекс-элемента** по поперечной оси
- align-content — **выравнивание строк** контейнера по поперечной оси
- order — **порядковый номер** флекс-элемента

### float

1. прил. **плавающий**, свойство блока, заставляющее последующие блоки обтекать его, напр. *float layout* — *раскладка на плавающих блоках*.
2. гл. **обтекать**, следовать вплотную к плавающему блоку, напр. *text is floating to the right* — *текст обтекает справа*.
3. гл. **прижимать**, упираться вправо или влево, обычно из-за плавающих свойств, напр. *float block to the left* — *прижмите блок влево*.



font  
шрифт

- font-weight — **насыщенность**
- font-weight:normal — **нормальная**
- font-weight:bold — **жирная** или **полужирная** (в типографской традиции)
- font-weight:bolder — **жирнее**
- font-weight:lighter — **светлее**
- font-style — **стиль шрифта**
- font-style:normal — **прямой**
- font-style:italic — **курсив**
- font-style:oblique — **наклонный**
- font-size — **размер шрифта, кегль**
- font-family — **семейство шрифта, гарнитура**
- font-family:serif — **с засечками**
- font-family:sans-serif — **без засечек**
- font-family:monospace — **моноширинный**

footer  
подвал

1. характерная визуальная область в конце страницы.
2. элемент <footer> — смысловая часть с метаинформацией.

formatting context

**контекст форматирования**, в CSS — область, в которой выстраиваются визуальные элементы страницы (фрагменты текста, блоки, ячейки и т.п.) по определенным правилам.

- block formatting context, BFC — **блочный контекст форматирования, БКФ** (напр. *element with overflow establishes the new BFC* — *элемент с overflow создает новый БКФ*)
- grid formatting context — **контекст форматирования гридов**
- flexbox formatting context — **контекст форматирования флексбоксов**
- inline formatting context — **строчный контекст форматирования**
- table formatting context — **табличный контекст форматирования**

fragment identifier

**идентификатор фрагмента**, текст после # в URL, позволяющий сослаться на конкретную часть документа. Используется в основном в спецификациях и при работе с SVG, в HTML устоялся более привычный термин **якорь**.

framework

**фреймвóрк**, набор библиотек и компонентов для упрощения разработки. Как правило, представляет собой базовый каркас продукта. *Hanp., Twitter Bootstrap.*

front-end

фронтéнд

1. область технологий для разработки веб-интерфейсов, включает вспомогательные инструменты и технологии работающие в браузере или клиентсайд.
2. внешняя часть серверной системы, отвечающая за приём данных, отправку ответов и генерацию кода для браузера, входит в серверсайд.

full screen

**полноэкранный режим**, поведение программы или её части, в котором она занимает весь экран целиком.

function

функция

1. в программировании — поименованный фрагмент программного кода (подпрограмма), к которому можно обратиться из другого места программы.
2. в CSS — может присутствовать в декларации на позиции значения. Напр., `url(img/gradient.png)`.

G

generated content

**сгенерированное содержимое**, часть документа, созданная с помощью CSS, псевдоэлемента и свойства content.

graceful degradation

**постепенная деградация** — подход, при котором интерфейс сайта деградирует для старых браузеров, где не поддерживаются какие-то из используемых технологий. При этом ущерб для пользователя по возможности смягчается. Например: цвет подложки вместо градиента, растровая графика вместо векторной, проигрывание видео с помощью плагина. Встречается также более буквальный перевод «*изящная деградация*».

gradient

**градиéнт**, плавный переход от одного цвета к другому.

- linear gradient — **линейный градиент**
- radial gradient — **радиальный градиент**
- conical gradient — **конический градиент**

grid

1. **сетка**, способ упорядочить расположение элементов в дизайне с помощью вертикального или горизонтального ритма, напр. *module grid* — *модульная сетка*.
2. **грид**, двумерная сетка из рядов и колонок в грид-раскладке.

grid layout

**грид-раскладка**, механизм раскладки, позволяющий располагать элементы по двумерной сетке — гриду (2).

- **grid container** — *грид-контейнер*
- **grid item** — *грид-элемент*
- **grid track** — *грид-полоса*, собирательный термин для рядов и колонок *грида*
- **grid cell** — *грид-ячейка*
- **grid line** — *грид-линия*, виртуальная граница между соседними *грид-полосами*, к которой можно привязывать *грид-элементы*
- **grid area** — *грид-область*, пространство для размещения *грид-элементов*, ограниченное четырьмя *грид-линиями*
- **grid gutter** — *грид-интервал*, промежуток между соседними *грид-полосами*



## Н

### header шапка

1. характерная визуальная область в начале страницы.
2. элемент `<header>` — смысловая озаглавливающая часть.

### height высота

## I

### icon font

**иконочный шрифт**, синтетический шрифт, содержащий иконки и другие плоские векторные изображения для оформления сайта, напр. *build an icon font* — *собирай иконочный шрифт*.

### immutable

**неизменяемый**, термин чаще всего применяется к данным (типам данных) в функциональном программировании.

### implement

**внедрять, реализовывать** напр. *Firefox has implemented support of everything* — *Файрфокс внедрил поддержку всего*.

### inline

1. **строчный**, имеющий строчные свойства.
  - inline block — **строчный блок**.
  - inline box — **элемент строки** в строчном форматировании: часть наполнения контейнера строки, к которой непосредственно применяются CSS-правила, напр. *anonymous inline box* — *анонимный элемент строки*.
  - inline element — **строчный элемент**.
2. **встроенный**, находящийся внутри строки кода.
  - inline styles — **встроенные стили**.
  - inline SVG — **встроенный SVG**.
  - inline script — **встроенный скрипт**.

### input

**элемент формы**, обеспечивающий взаимодействие пользователя с интерфейсом, например: текстовое поле, кнопка, переключатель.

### interoperability

### интероперабельность

## K

### kerning

**кернинг**, расстояние между парами букв, учитывающее их форму для более равномерного набора.

### keyframe

**ключевой кадр**, одна из заданных точек анимации, изменения между которыми происходят автоматически.

### keyframe animation

**покадровая анимация**, способ задания анимации в CSS с помощью ключевых кадров.

### keyword

### ключевое слово

## L

### label

**метка**, устанавливает связь между определённым элементом и элементом формы.

### last call (LC)

**последний черновик**, статус спецификации W3C для последних правок перед статусом кандидата в рекомендации (*отменен в 2014 г., встречается только у старых спецификаций*).

### layout

**раскладка**, расположение основных блоков на странице, напр. *3 columns layout* — *раскладка в 3 столбца*.

### legend

**легенда**, заголовок группы элементов формы, определяется с помощью группы полей

### length

### длина

### letter-spacing

**межбуквенное расстояние, трэкинг**, общее расстояние между буквами в тексте, отличается от кернинга.

### library

**библиотека**, набор готовых решений для упрощения разработки. Как правило, библиотека реализует какую-то конкретную функцию или набор связанных функций, напр., анимацию.

### line-box

**контейнер строки** в строчном форматировании: виртуальный контейнер, содержащий в себе все элементы одной строки.

### line-height

**высота строки, интерлиньяж**, расстояние между базовыми линиями текста.

### list

**список**, последовательное перечисление группы элементов.

- ordered list — **упорядоченный список**, отсортированный по какому-либо принципу.
- unordered list — **неупорядоченный список**, в котором порядок следования не имеет значения.

### list-style

### стиль списка

**M**

**margin**  
**внешний отступ**, напр. *margin-bottom:-10px* — *нижний отрицательный отступ 10 пикселей*.

**margin box**  
**внешние отступы чего-л.**

**matrix**  
**мáтрица (трансформации)**, описание трансформации объекта с помощью матрицы.

**media query**  
**медиавыражение**, объявление директивы @media с некими условиями, позволяет применять стили в зависимости от возможностей устройства. Вариант перевода «медиазапросы» менее точно отражает суть работы.

**media query list**  
**условия медиавыражения**, список условий, определяющих, в каких случаях будут применяться объявления внутри директивы @media. Напр., *only screen and (min-width: 35em)*.

**media type**  
**тип устройства**, может быть условием медиавыражения. Указывает, для какого способа вывода предназначены стили (напр. *screen* — на экран, *print* — на печать, *speech* — для синтезатора речи).

**media feature**  
**характеристика устройства**, может быть условием медиавыражения. Определяет характеристику устройства. Напр., *min-width*.

**mixin**  
**прímесь**, набор свойств, расширяющий поведение другой сущности, встречается в CSS-препроцессорах и паттернах JavaScript, напр. *mixin is clearing* — *примесь очищает поток*.

**mobile first**  
**Сначала мобильные**, подход разработки, при котором проектирование сервиса начинается с мобильной версии, а не с версии для больших экранов. Термин введён Люком Вроблевски в одноимённой книге.

**mock object**  
**фиктивный объект**

**mock-up**  
**макет**, внешний вид или дизайн сайта различной степени детализации, напр. *designer has sent a page mock-up* — *дизайнер прислал макет страницы*.

**modal**

1. прил. **модáльный**, блокирующий интерфейс для выполнения какого-либо действия, напр. *modal dialog* — *модальный диалог*.
2. сущ. **модальное окно (диалог)**, элемент интерфейса, см. *модальный*.

**multiple**  
**множественный**, повторённый два и более раз, напр. *multiple backgrounds* — *множественные фоны*.

**multiple columns**

1. **мультико́лонки**, механизм из CSS, позволяющий выстроить текст в несколько колонок, напр. *multiple columns support in IE* — *поддержка мультиколонок в IE*. Предпочтительнее, чем громоздкие «множественные колонки».
2. **многоко́лоночный**, состоящий из нескольких колонок, напр. *multiple columns layout* — *многоколоночная раскладка*.

**N**

**nested**  
**вложенный**, находящийся внутри чего-либо, напр. *double nesting* — *двойная вложенность*.

**node**  
**узел**, элемент структуры, напр. *child node* — *дочерний узел*.

**number**  
**число, числовой тип**

**O**

**opacity**  
**непрозрачность**, степень непрозрачности, напр. *opacity:0.1* — *непрозрачность 10%*. Часто ошибочно называется прозрачностью.

**opaque**  
**непрозрачный**, степень непрозрачности, напр. *opaque by 75%* — *непрозрачен на 75%*, значит прозрачность 25%.

**outline**  
**обводка**, внешний равномерный контур элемента, не участвующий в блочной модели.

**P**

**package manager**  
**менеджер пакетов**, реже **диспетчер пакетов**, вспомогательное ПО для управления установкой, настройкой, обновлением и удалением компонентов ПО.

**padding**  
**внутренний отступ**, напр. *padding-top:10px* — *верхний внутренний отступ 10 пикселей*.

**pagination**

1. **постраничная навигация**, обычно список ссылок с номерами страниц, на которые разбит документ.
2. **разделение на страницы**, разделение документа на отдельные страницы для удобства чтения, загрузки или других целей.

**pattern**  
**пáттерн**, формализованный подход к написанию кода, напр. *JavaScript patterns* — *паттерны JavaScript*.

**parent**  
**родитель**

**percentage**  
**процентное значение, проценты**

**performance**  
**быстродействие**, характеристика скорости загрузки, отрисовки и т.п., напр. *CSS performance* — *быстродействие CSS*.

**performance budget**  
**бюджет на быстродействие**, максимально допустимые показатели быстродействия веб-приложения. Выход за рамки такого бюджета означает спад быстродействия. Напр. *Performance Budget Metrics* — *показатели бюджета на быстродействие*.

**persistent data structures**  
**постоянные структуры данных**

**pinch**  
**сведение (разведение) пальцев**, жест для управления сенсорным интерфейсом: сведение (в зависимости от направления — разведение) пальцев на сенсорном экране.

## pixel

**пíксель**, единица измерения для экранного дизайна, в мн.ч. **пиксели**, сокр. **пк**.

- physical pixel — **физический пиксель**, ячейка на матрице, единица разрешения дисплея устройства.
- device independent pixel (dip) — **виртуальный пиксель**, независимый от устройства пиксель, может быть больше, меньше или равен физическому.
- bitmap pixel — **растровый пиксель**.

## pixel density

**плотность пикселей**, количество пикселей, которое помещается в единицу линейного размера экрана, напр. *pixel density of the screen is 326 ppi* — *плотность пикселей экрана 326 пикселей на дюйм*.

## pixel ratio

**пиксельное соотношение**, отношение физического пикселя к виртуальному, напр. *screen has pixel ratio of 2* — *пиксельное соотношение экрана равно 2*.

## placeholder

1. **заглушка**, временный заменитель картинки, текста и т.п. Напр.: *Вместо портрета президента вставь пока заглушку с котиками*.
2. **подсказка поля**, атрибут текстового поля, предназначен для подсказки о том, как это поле заполнять. Чаще всего представляет собой пример заполнения. Напр.: `placeholder="+7 (999) 123-12-13"`.

## ppi (pixel per inch)

**пикселей на дюйм**, единица измерения плотности пикселей.

## plugin

## TODO

## polyfill

**полифíл**, скрипт, воссоздающий недостающую функциональность, напр. *new polyfill for IE6* — *новый полифил для IE6*.

## polyfill

**пролифíл**, один из вариантов полифила, отличающийся тем, что воспроизводит *вероятную* функциональность, описанную в спецификации, но пока не реализованную в браузерах.

## porup

**всплывающее окно, попáп**, отдельное окно или элемент интерфейса, появляющийся поверх текущего, напр. *annoying porup* — *назойливый попáп*.

## position

### позиционирование

- position:static — **статичное**
- position:absolute — **абсолютное**
- position:relative — **относительное**
- position:fixed — **фиксированное**
- position:sticky — **закреплённое**

## preprocessor

**препроцéссор**, инструмент, преобразующий код из одного синтаксиса в другой, напр. *Sass CSS preprocessor* — *CSS-препроцессор Sass*.

## progress bar

**индикатор выполнения**, элемент интерфейса, отражающий прогресс выполнения операции.

## progressive enhancement

**прогрессивное улучшение** — подход, при котором все браузеры получают одинаковую базовую функциональность, но в современных браузерах эта функциональность расширяется в соответствии с поддержкой новых технологий. Например: градиенты, векторная графика, воспроизведение видео.

## Progressive Web App, PWA

**прогрессивное веб-приложение**, — подход к разработке веб-приложений, сочетающий преимущества обычных сайтов и нативных приложений: быстрая загрузка, установка в ОС, работа в офлайне, пуш-уведомления, доступ к системным API. Информация для установки описывается в веб-манифесте. Прогрессивность подхода в том, что в обычном браузере — это просто сайт, а в поддерживающих браузерах и ОС — приложение.

## promises

**прóмисы**, способ асинхронного выполнения скриптов, противопоставляется использованию обратных вызовов.

## property

**свойство**, напр. *CSS-свойство*.

## proposed recommendation (PR)

**предложенная рекомендация**, статус спецификации W3C, вероятный кандидат в рекомендацию.

## pseudo-element

**псевдоэлемент**, дополнительный внутренний элемент, созданный с помощью CSS и `::before` или `::after`.

## pseudo-class

**псевдокласс**, используется для привязки декларации к определенному состоянию элемента DOM. Отделяется от селектора знаком двоеточия `:`. Напр., декларация блока правил с селектором `.item:hover` применится, когда на `.item` наведут курсор.

## pull request, pr

**пулреквест**, запрос на принятие изменений в ветку репозитория из форка или другой ветки.

## Q

## quirks mode

**режим совместимости**, режим интерпретации и отображения страницы браузером, в котором браузер целенаправленно отходит от стандартов ради отображения документов, созданных до начала 2000-х гг. и полагавшихся на ошибки и особенности браузеров того времени. Используется для страниц без доктайпа и с устаревшими доктайпами (напр. HTML 3.2).

## R

### radio button

**радиокнопка**, элемент формы `<input type="radio">` для выбора одного варианта из нескольких, напр. *radio button is checked* — *радиокнопка выбрана*.

### recalculate

### перерасчет

### recommendation (REC)

**рекомендация**, финальный статус спецификации W3C, рекомендованной для внедрения в браузеры и использования разработчиками.

### reflow

### перекомпоновка

### rem

**рем**, единица измерения в CSS, которая зависит от размера шрифта корневого элемента `<html>`, напр. *font-size: 3rem* — *размер шрифта 3 рема*.

### render

### отрисовка

### repaint

### перерисовка

### reset

**сброс**, обычно сброс стилей по умолчанию в CSS, напр. *style reset file* — *файл сброса стилей*.

### resolution

**разрешение**, количество физических пикселей на экране устройства, напр. *screen resolution is 1024×768 pixels* — *разрешение экрана 1024×768 пикселей*.

### responsive design

**отзывчивый дизайн**, один из технологических методов создания адаптивного дизайна.

### Responsive Images Community Group (RICG)

**Общественная группа по адаптивным изображениям**

### root

### корень

### rounded corners

**скруглённые углы**, напр. *rounded corners are out of fashion* — *скруглённые углы вышли из моды*.

## rowspan

**объединение строк**, HTML-атрибут для объединения нескольких ячеек таблицы в одном столбце.

## rule-set

**блок стилей элемента**, состоит из селектора (или медиавыражения) и блока деклараций.

## S

### shallow comparison

**поверхностное сравнение**, сравнение, при котором проверяется ссылаются ли две переменные на один и тот же объект в памяти.

### scaffolding

**скафóлдинг**, автоматическая генерация кода по описанию, метод метапрограммирования.

## scale

1. суш. **масштаб**
2. гл. **масштабировать**

## scope

**область видимости**, ограниченная часть программной структуры, в которой доступна объявленная переменная, либо применяются `<style scoped>` стили.

## script

**скрипт**, инструкции, описывающие поведение страниц, напр. *scripts are not loaded yet* — *скрипты пока не подгрузились*.

## scroll

1. **прокрутка**, элемент интерфейса для перемещения скрытой части страницы или блока, является признаком того, что вложенный элемент больше родительского, напр. *horizontal scroll is disabled* — *горизонтальная прокрутка запрещена*.
2. **прокручивать**, перемещать скрытую часть страницы или блока.

## scrollbar

**полоса прокрутки**, формальное название прокрутки, напр. *system scrollbars* — *системные полосы прокрутки*.

## sectioning roots

## структурные основы

## semantics

**семантика**, смысловая нагрузка HTML-элементов, напр. *semantic coding* — *семантическая вёрстка*.

## selector

**селектор**, необходимая структурная часть блока стилей. Отвечает за привязку деклараций к элементам DOM.

- **simple selector** — **простой селектор**, может быть псевдоклассом или одним из следующих типов:
  - **type selector** — **селектор по типу**, обращается к тегу, напр. `ul`, `input` и др.
  - **universal selector** — **универсальный селектор**, «звездочка»: `*`.
  - **ID selector** — **селектор по идентификатору**, напр. `#content`.
  - **class selector** — **селектор по классу**, напр. `.item`.
  - **attribute selector** — **селектор по атрибуту**, напр. `[type=submit]`.
- **compound selector** (др. sequence of simple selectors) — **составной селектор**, цепочка простых селекторов, не разделённых комбинаторами, напр. `input[type=submit]:focus`.
- **complex selector** — **сложный селектор**, несколько простых и/или составных селекторов, разделённых комбинаторами:
  - **descendant selector** — **селектор потомка**, напр. `ul li`.
  - **child selector** — **селектор непосредственного потомка**, напр. `#buttons > *`. Также часто называется **дочерним**.
  - **adjacent sibling selector** — **селектор непосредственного соседства**, напр. `.item + .item`. Также часто называется **соседским**.
  - **general sibling selector** — **селектор соседства**, напр. `.item ~ .item`.



**server-side**  
**серверсайд**, область технологий, работающих на сервере, часто синоним бэкенда.

**shadow DOM**  
**теневой DOM**, теневая модель документа

**shapes**  
**фигуры**, CSS-фигуры

**shim**  
**шим**, код, помогающий унифицировать работу с браузерами, обычно реализует недостающую или нормализует существующую поддержку, напр. *HTML5 shim for IE8 — HTML5-шим для IE8*.

**sibling**  
**сосед** (в знач. «соседний элемент»).

**sidebar**  
**боковая панель**, боковая часть сайта со второстепенным содержимым, напр. *sidebar is on the right side — справа находится боковая панель*.

**skew**  
1. сущ. **наклон**  
2. гл. **наклонять**

**source maps**  
**карты кода**, специальная разметка, позволяющая привести в соответствие исходные файлы и скомпилированный код для отладки.

**specification, spec**  
**спецификация**, документация по технологии для разработчиков и производителей браузеров, обычно в одном из статусов: от предложенной до утверждённой W3C, напр. *specification draft — черновик спецификации*.

**stacking context**  
**контекст наложения**, элемент визуальной структуры страницы, ограничивающий действие свойства z-index его потомков и размещаемый по оси z как единое целое (либо целиком поверх, либо целиком под любым другим элементом). Напр. *elements with opacity create new stacking context — элементы с полупрозрачностью образуют новый контекст наложения*.

**staging server**  
**отладочный сервер**, сервер, на котором продукт проходит тестирование и отладку.

**standards mode, standards compliance mode**  
**стандартный режим**, режим интерпретации и отображения страницы браузером, в котором браузер обязан максимально полно соблюдать стандарты, насколько может. Используется для страниц с более-менее современными доктайпами, напр. доктайпом HTML5.

**statement**  
**объявление**, структурный элемент CSS. Это может быть блок стилей элемента, директива импорта, медиавыражение и т.д.

**string**  
1. **строковый тип**, в программировании — тип данных, значениями которого является произвольная последовательность (строка) символов алфавита.  
2. **строка**, в CSS — строковое значение директивы или свойства. Напр., в content: "201d" строкой является «\201d».

**stub object**  
**заглушка**

**style**  
1. сущ. **стили**, напр. *apply styles to an element — применить стили к элементу*, мн.ч. предпочтительнее: *стили элемента*, а не *стиль элемента* (неправильно).  
2. гл. **оформлять**, напр. *how to style selects — как оформить селекты*, предпочтительнее, чем двусмысленное «стилизовать».

**style guide**  
**руководство по стилю**, напр. *project code style guide — руководство по стилю кода проекта*.

**stylesheet**  
**таблица стилей**, напр. *таблица стилей документа*.

**swipe**  
**смахивание**, жест для управления сенсорным интерфейсом: смахивающее движение пальцев по сенсорному экрану.

**submit**  
**отправка**, напр. *отправка формы*.

**T**  
**tab**  
1. **вкладка**, один из слоёв интерфейса сайта или программы, который активируется выбором его заголовка, напр. *open in new tab — открыть в новой вкладке*.  
2. **таб**, клавиша табуляции, вставляет символ табуляции, переключает фокус на следующий элемент интерфейса, либо дополняет частично набранное слово, напр. *press Ctrl Tab — нажмите контрол-таб*.  
3. **таб**, символ табуляции, имеет настраиваемую ширину и используется для отступов в коде, напр. *tabs are better than spaces — табы лучше пробелов*.

**tag**  
1. **тег**, обозначение элемента в HTML, существуют открывающие, закрывающие и одиночные теги. Напр. *закрывающий тег отсутствует — closing tag is missing*.  
◦ tag name — **название тега**  
◦ start tag — **открывающий тег**  
◦ end tag — **закрывающий тег**  
2. **метка**, маркер содержимого, обычно короткое слово.

**tap**  
**нажатие**, жест для управления сенсорным интерфейсом: однократное касание сенсорного экрана.

**text**  
**текст**

**textarea**  
**текстовая область**, элемент формы для ввода многострочного текста.

**text-align**  
**выравнивание**, **выключка**, ориентация строк текста в определённом направлении, либо равномерно по ширине блока.  
• text-align:left — **влево**  
• text-align:right — **вправо**  
• text-align:justify — **по ширине**

**text-shadow**  
тень (текста), напр. *text-shadow is red* — *тень текста красная*.

**thumb**  
1. Сокращение от thumbnail.  
2. **ползунок**, элемент управления полосой прокрутки.

**thumbnail**  
миниатюра, уменьшенная копия изображения.

**title**  
1. **заголовок**, если речь идёт об элементе <title>, либо о заглавии элемента.  
2. **подсказка**, если речь идёт об атрибуте title="".

**toggle**  
переключать, переводить из одного состояния в другое, переключить классы — *toggle classes*.

**tooltip**  
всплывающая подсказка, вспомогательная информация, появляющаяся при наведении на элемент или при его активации, в частности title="".

**touch interface**  
сенсорный интерфейс

**transform**  
трансформация, напр. *transform:scale(2)* — *трансформация масштабированием в 2 раза*.

**transition**  
переход, напр. *transition:all 2s linear* — *линейный переход длительностью 2 секунды*.

**translate**  
перемещать, перемещение объекта с помощью CSS-трансформации, напр. *translate(x, y)* — *перемещение на X и Y*.

**transpiler**  
транспилер, компилятор, преобразующий исходный код на одном яз. программирования в исходный код на другом яз. программирования.

**typography**  
типографика, оформление текста с помощью выбора гарнитуры, параметров строки и других.

**U**  
**unit**  
единица изменения, напр. *rem unit* — *единица измерения «рем»*.

**unit testing**  
модульное тестирование

**URI**  
унифицированный идентификатор ресурса, последовательность символов, идентифицирующая абстрактный или физический ресурс.

**URL**  
1. *разг. адрес*. Значение рекомендовано к использованию при переводах статей.  
2. **унифицированный указатель ресурса**, URI, который предоставляет ещё и информацию о местонахождении этого ресурса, напр. *background: url("http://www.example.com/picture.png")*

**user experience (UX)**  
пользовательский опыт, совокупность ощущений от взаимодействия с чем-л. (интерфейсом, девайсом, продуктом). В контексте области знаний рекомендуется употреблять аббревиатуру UX (напр., *UX-специалист*).

**user interface (UI)**  
пользовательский интерфейс, в прозрачном контексте рекомендуется употреблять просто «интерфейс».

**utility**  
утилита, программа с узким назначением.

**V**  
**valid**  
валидный, соответствующий стандарту, спецификации или другому нормативу, напр. *valid markup* — *валидная разметка*.

**validator**  
валидатор, средство проверки соответствия стандарту, спецификации или другому нормативу, напр. *W3C markup validator* — *валидатор разметки W3C*.

**value**  
значение, напр. *значение CSS-свойства*.

**vendor prefix**  
браузерный префикс, приставка к CSS-свойству -webkit- или -moz-, напр. *add vendor prefixes* — *добавьте браузерные префиксы*.

**vendor files**  
сторонние файлы, стили, скрипты и другие ресурсы (зависимости) других авторов, использующиеся в проекте, напр. *tons of vendors scripts included in the code* — *куча сторонних скриптов подключены в коде*.

**vertical-align**  
вертикальное выравнивание, расположение элементов по вертикали в пределах строки текста или таблицы.

**viewport**  
вьюпорт  
1. видимая область документа в рамках экрана.  
2. элемент <meta name="viewport">, управляющий адаптацией страниц на мобильных устройствах.

**W**  
**website, web site**  
сайт, объединённые под одним адресом страницы, перевод «веб-сайт» избыточен.

**working draft (WD)**  
рабочий черновик, ранний статус спецификации W3C.

**World Wide Web Consortium (W3C)**  
Консорциум всемирной сети, организация, разрабатывающая веб-стандарты. Переводить «web» как «паутину» в XXI веке уже смешно.

**web standards**  
веб-стандарты

**width**  
ширина

**word-spacing**  
межсловный пробел

**workaround**  
обходное решение (приём), способ решения задачи в ограниченных условиях, напр. *to figure out workaround* — *придумать обходной приём*.

# WEB ACADEMY

---

## PROGRAMMING COURSES

---



(044) 233 - 15 - 22, (063) 233 - 15 - 22

