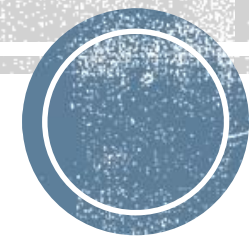


ДИЗАЙН САЙТА



ОСНОВНЫЕ ЗАДАЧИ И КЛЮЧЕВЫЕ ЦЕЛИ::

- аргументация основных принципов и теории в области веб-дизайна и программирования;
- акцентирование внимания на необходимости использования эффективного веб-дизайна, основанного на инновациях, технологиях и современных тенденциях, для достижения стратегических целей организации;
- рассмотрены основные концепции дизайна и разработки веб-сайтов, основные элементы, инструменты и современные тенденции в их построении и реализации;
- выявлена необходимость улучшения организации сайта и поисковых систем с целью повышения конверсии и увеличения доли целевой аудитории;
- представлены основные методы и этапы программирования для современных языков программирования HTML, JavaScript и PHP.

Полученные научные и методические аспекты могут быть практически применены в качестве:

- инструментов для изучения поведения потребителей в Интернете;
- понимания потребностей в веб-дизайне и разработке веб-сайтов;
- развития современного веб-дизайна и программирования.



СОДЕРЖАНИЕ

The diagram shows a vertical blue bar on the left containing the word 'СОДЕРЖАНИЕ' (Table of Contents) in white, rotated 90 degrees. A horizontal line extends from the right side of this bar to a vertical line. From this vertical line, ten horizontal lines branch out to the right, each pointing to a blue rectangular box containing a chapter title in Russian. The boxes are stacked vertically and connected to the central vertical line by a single continuous line.

РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ВЕБ-ДИЗАЙНА

РАЗДЕЛ 2. ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА

РАЗДЕЛ 3. ФУНКЦИОНАЛЬНОСТЬ ВЕБ-САЙТА

РАЗДЕЛ 4. ВЕРСТКА ВЕБ-САЙТА

РАЗДЕЛ 5. ЯЗЫКИ ПРОГРАММИРОВАНИЯ ДЛЯ ВЕБ-САЙТОВ

РАЗДЕЛ 6. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ HTML

РАЗДЕЛ 7. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVASCRIPT

РАЗДЕЛ 8. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP

РАЗДЕЛ 9. ПРОДВИЖЕНИЕ И ПОИСКОВАЯ ОПТИМИЗАЦИЯ ВЕБ-САЙТА

РАЗДЕЛ 10. ТРЕНДЫ ВЕБ-ДИЗАЙНА И РАЗРАБОТКИ ВЕБ-САЙТОВ

Историческое развитие и ключевые особенности ведущих школ графического дизайна

| Школа графического дизайна и годы развития | Ключевые особенности |
|---|--|
| Школа американской рекламной графики (1920-1950 гг.) | Реклама 1920-х в Америке была в значительной степени традиционна по внешнему виду, творческая сила, главным образом, черпалась от копирования лучших европейских образцов. Послевоенное процветание проходило под лозунгом «покупать, покупать, покупать». Стремительно меняется дизайн автомобилей, каждый год - новая модель, но дизайн рекламных объявлений в целом остается прежним. Приравнивая графический дизайн к искусству, все равно школе не удалось переломить вал непривлекательных объявлений, сопровождавших потребительский бум Америки. |
| Швейцарская школа графического оформления (1950-1960 гг.) | Это стиль графического дизайна, который был разработан дизайнерами в Швейцарии в 1950-х годах и оказал глубокое влияние на графический дизайн как часть модернистского движения, повлияв на многие области, связанные с дизайном, включая архитектуру и искусство. Данный стиль подчеркивает простоту, удобочитаемость и объективность. Одной из самых важных вещей, которые дает этот метод, является использование сеток для организации контента или то, что называется дизайном сетки. Идея этого подхода заключалась в том, чтобы помочь дизайнерам использовать последовательный, последовательный и структурированный подход к макету страницы. Это согласовалось с идеологией швейцарского дизайна, в которой упор делается на регулярность и удобочитаемость превыше всего, поскольку сеточный дизайн стал опорой на протяжении 20-го века и стал стандартом корпоративного дизайна. Строгое соблюдение этой концепции с тех пор уменьшилось, ее влияние, безусловно, все еще ощущается. В настоящее время дизайн сетки по-прежнему преподается во многих школах дизайна как важный инструмент для многих проектов. |
| Польская школа плаката (1960-1970 гг.) | В польском плакате универсальность соединилась с уникальностью, рациональность с чувственностью, графичность с живописностью, простота со сложностью. Все это внесло в графический дизайн особую, узнаваемую фактуру и неповторимые образы. Произведение польского плаката представляло собой четкий замысел в рамках дизайнерского искусства, поскольку выражало мысли, чувства и заботы людей, а не было просто изображением или средством привлечения. На первой Международной биеннале плаката в Варшаве в 1966 году, на сопровождавшем ее международном научном симпозиуме, посвященном образу и роли плаката в урбанистической среде, влияние польской школы на мировой графический дизайн получило окончательное признание. |
| Японская школа плаката (1960-1980 гг.) | Работы японских дизайнеров поражают своей странной, ни на что не похожей, удивительной эстетикой. Одни дизайнеры создают свои произведения, основываясь на многовековой истории и культуре Страны восходящего солнца, а другие, наоборот, полностью от нее абстрагируются и работают исключительно с современными образами. |

СУТЬ ВЕБ-ДИЗАЙНА И ЕГО КЛЮЧЕВОЕ НАЗНАЧЕНИЕ В СОВРЕМЕННЫХ УСЛОВИЯХ

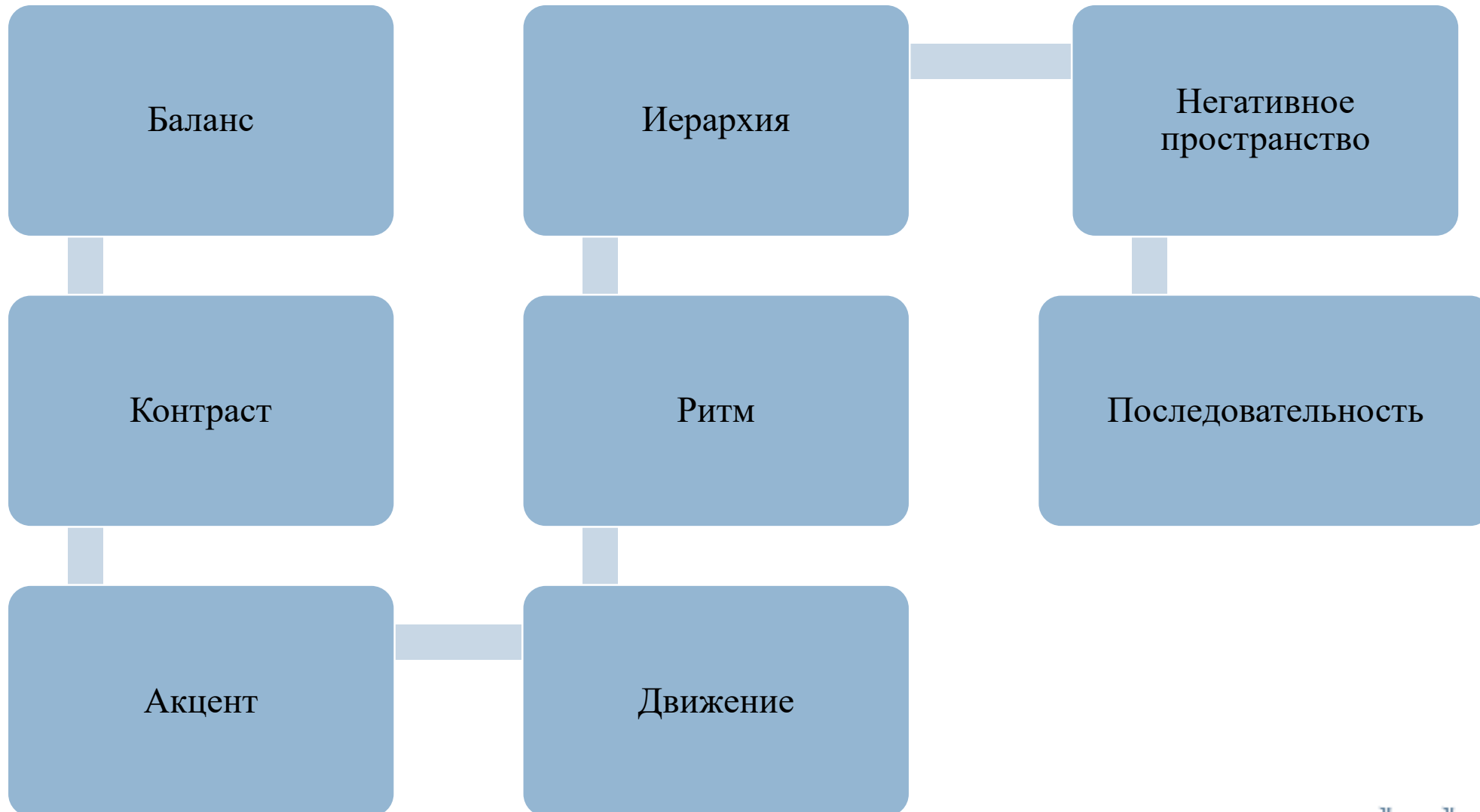
Веб дизайн — это планирование и создание контента таким образом, чтобы сайт был красивым, функциональным и удобным. Чтобы достичь каждой из этих целей, веб-дизайнеры используют широкий набор инструментов.

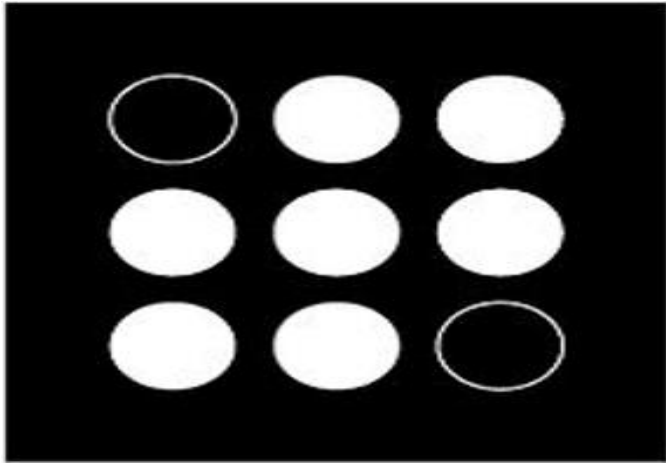
При использовании инструментов веб-дизайна веб-сайты приобретают правильную цветовую схему, подходящие шрифты, фотографии, изображения и элементы анимации, а также специальные эффекты.

Для обеспечения максимальной функциональности веб-сайта при помощи веб-дизайна продумывается структура страниц и их навигация, наполняются страницы и разделы информативным и актуальным контентом и добавляются дополнительные элементы в виде: опросов, форм подписки, виджетов с социальными сетями, интерактивные карты и многое другое.

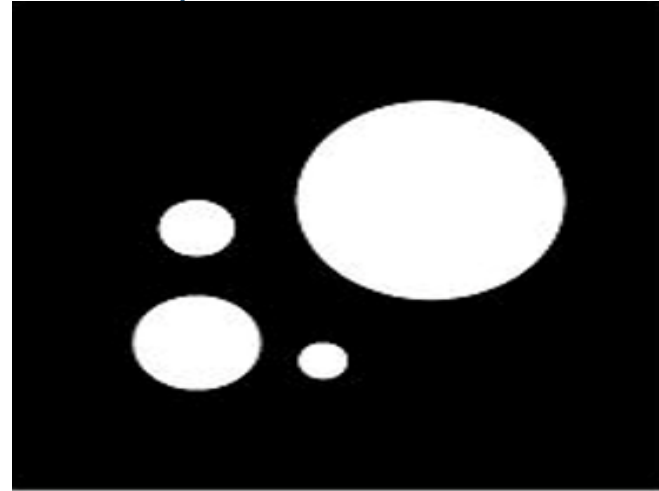
При помощи веб-дизайна осуществляется взаимодействие с веб-сайтом интуитивно понятным и для избежания путаницы. Расположение самой важной информации в верхней части экрана, чтобы пользователю не приходилось прокручивать страницу вниз.

ОСНОВНЫЕ ПРИНЦИПЫ ВЕБ-ДИЗАЙНА В СОВРЕМЕННЫХ УСЛОВИЯХ



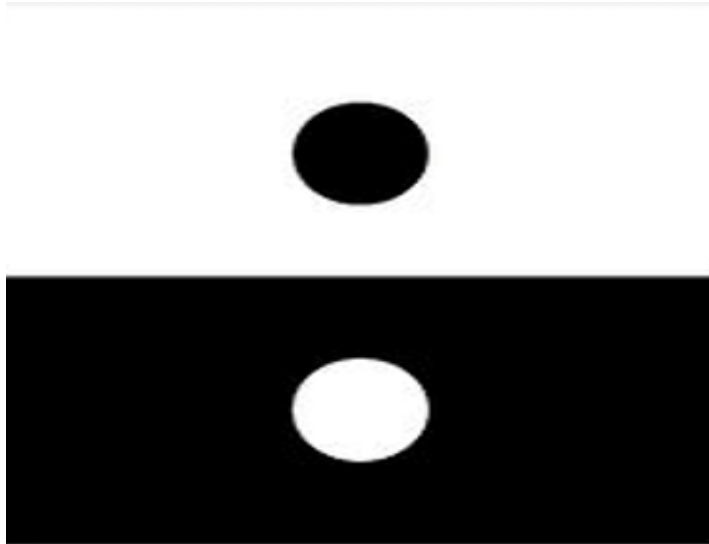


Symmetrical web design

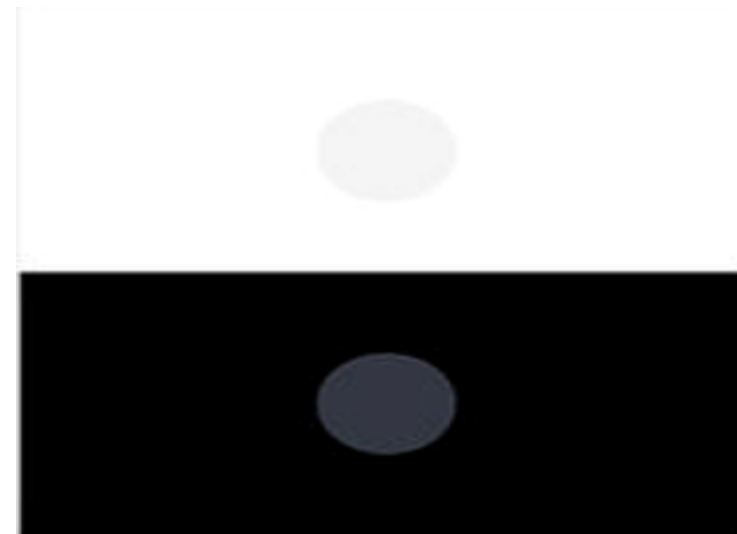


Asymmetrical web design

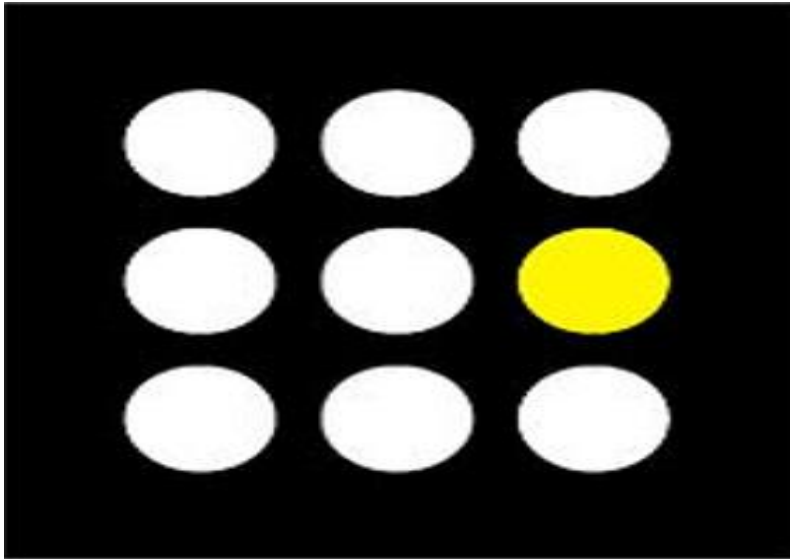
ОСНОВНЫЕ ВАРИАНТЫ КОНТРАСТА ВЕБ-ДИЗАЙНА В СОВРЕМЕННЫХ УСЛОВИЯХ



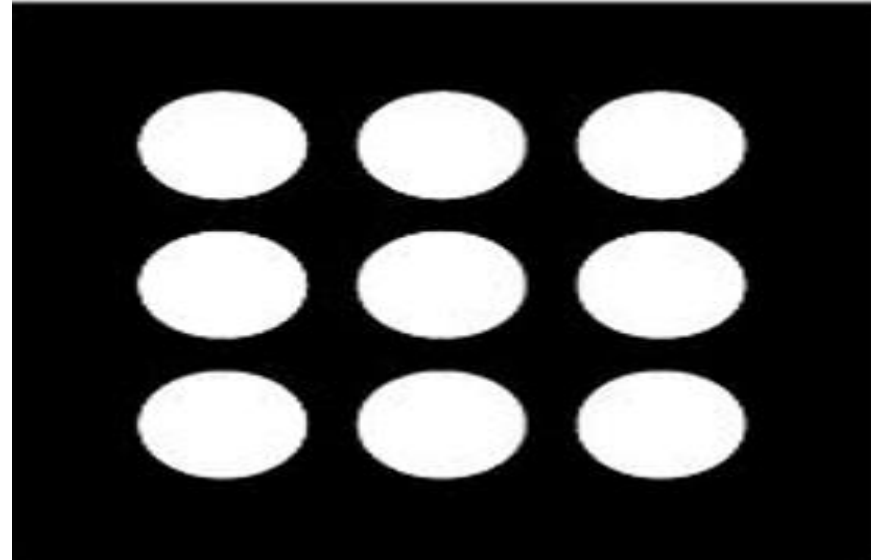
Rational Contrast



Not a rational contrast

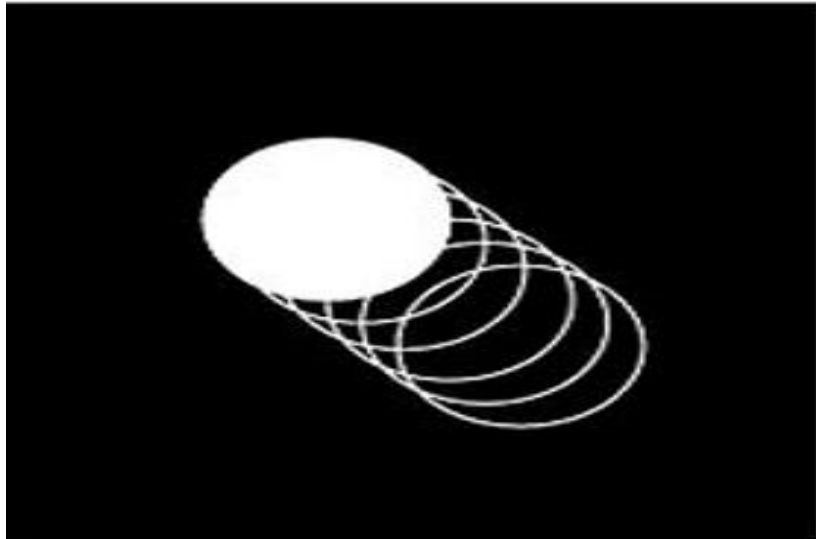


Accent web page design website

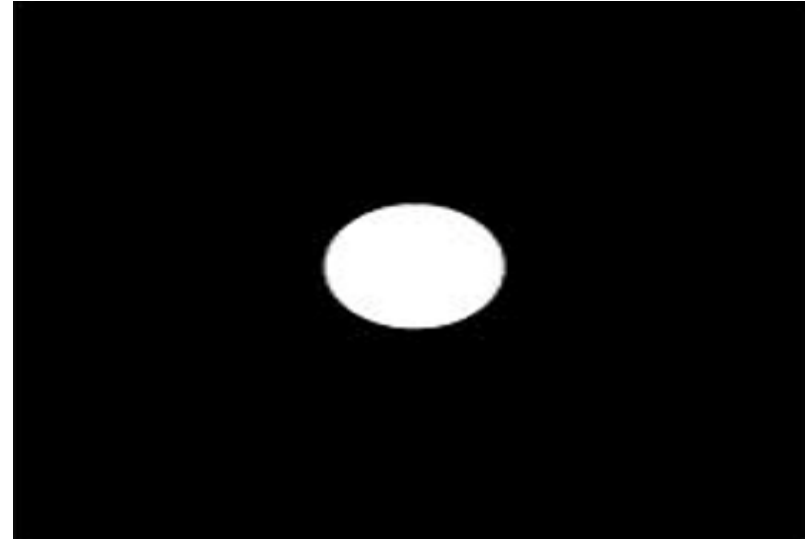


Not accentuated web design of a website page

ОСНОВНЫЕ ВАРИАНТЫ ДВИЖЕНИЯ ЭЛЕМЕНТОВ НА СТРАНИЦАХ ВЕБ-САЙТА ПРИ ПОМОЩИ ИНСТРУМЕНТОВ ВЕБ-ДИЗАЙНА В СОВРЕМЕННЫХ УСЛОВИЯХ

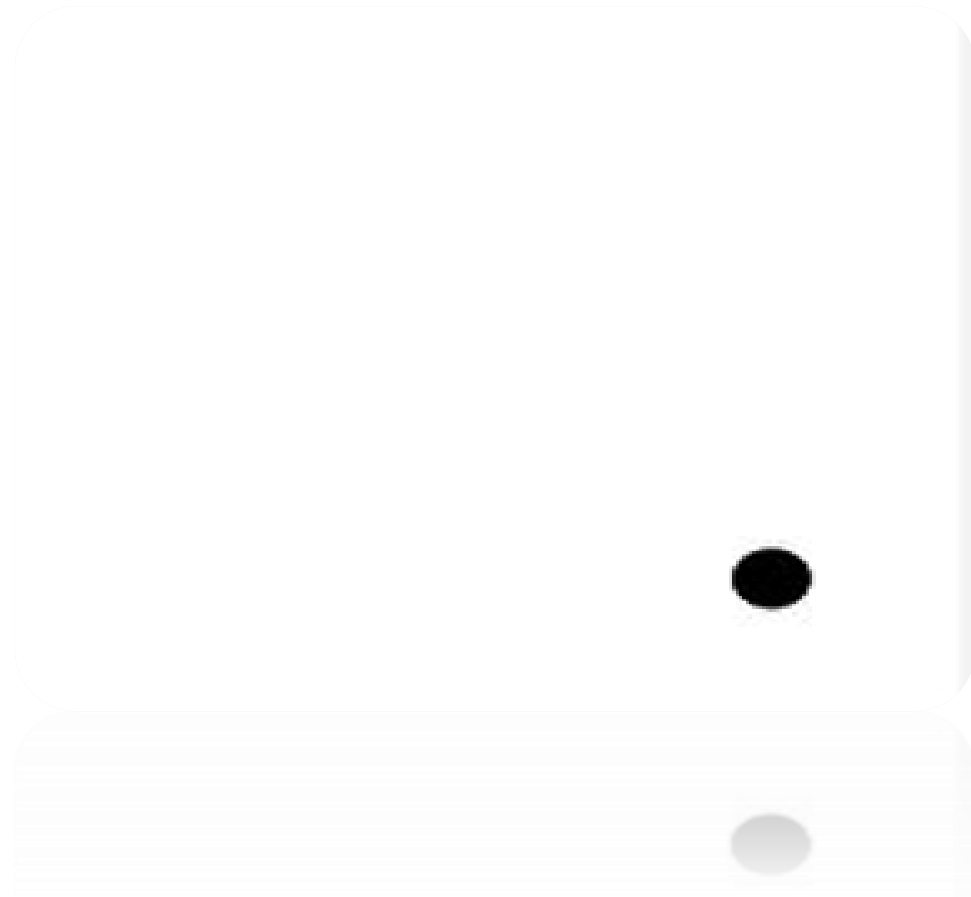


Driving element on a website page



Static element on a website page

ОСНОВНЫЕ ВАРИАНТЫ СОЗДАНИЯ ВЕБ-САЙТА ПРИ ПОМОЩИ ИНСТРУМЕНТОВ ВЕБ-ДИЗАЙНА, КАК НЕГАТИВНОЕ ПРОСТРАНСТВО



Веб-сайт из элементами негативного пространства веб-дизайна

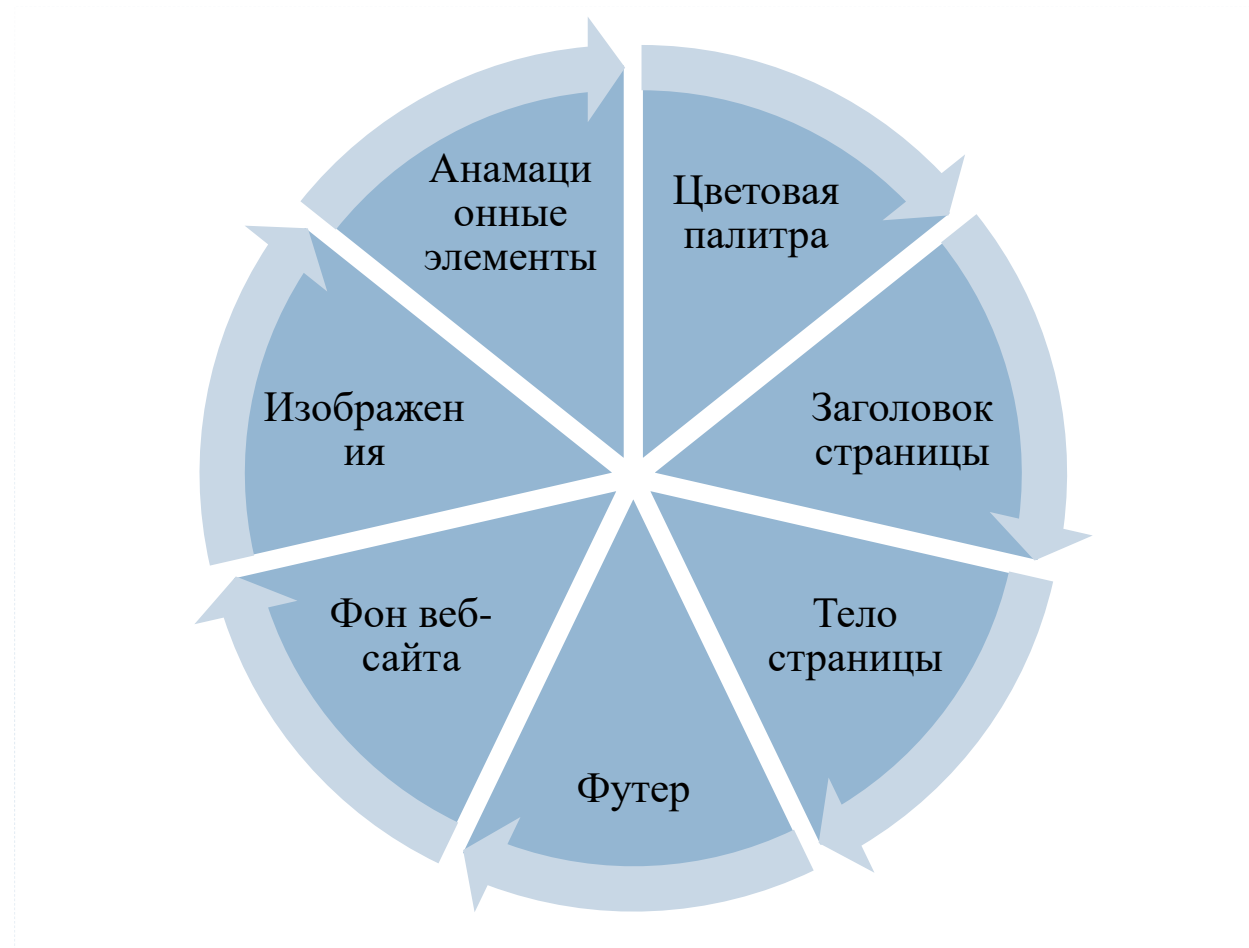


Веб-сайт без элементами негативного пространства веб-дизайна

КЛЮЧЕВЫЕ ХАРАКТЕРИСТИКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА В ВЕБ- ДИЗАЙНЕ



ОСНОВНЫЕ ВИЗУАЛЬНЫЕ ЭЛЕМЕНТЫ ВЕБ-ДИЗАЙНА СОВРЕМЕННЫХ ВЕБ- САЙТОВ



1. *Определение бизнес-цели.*

Важным элементом создания веб-ресурса является определение его бизнес-цели для понимания, зачем и для кого разрабатывается веб-сайт. Кроме того, веб-дизайнеры должны понимать, чего ждут клиенты от веб-сайта компании.

2. *Идентификация области внедрения веб-сайта*

Определение масштаба - один из самых сложных этапов всего процесса создания веб-сайта. Масштаб и объем работ по проекту может быть изменчивым и меняться в зависимости от различных факторов при создании веб-сайта.

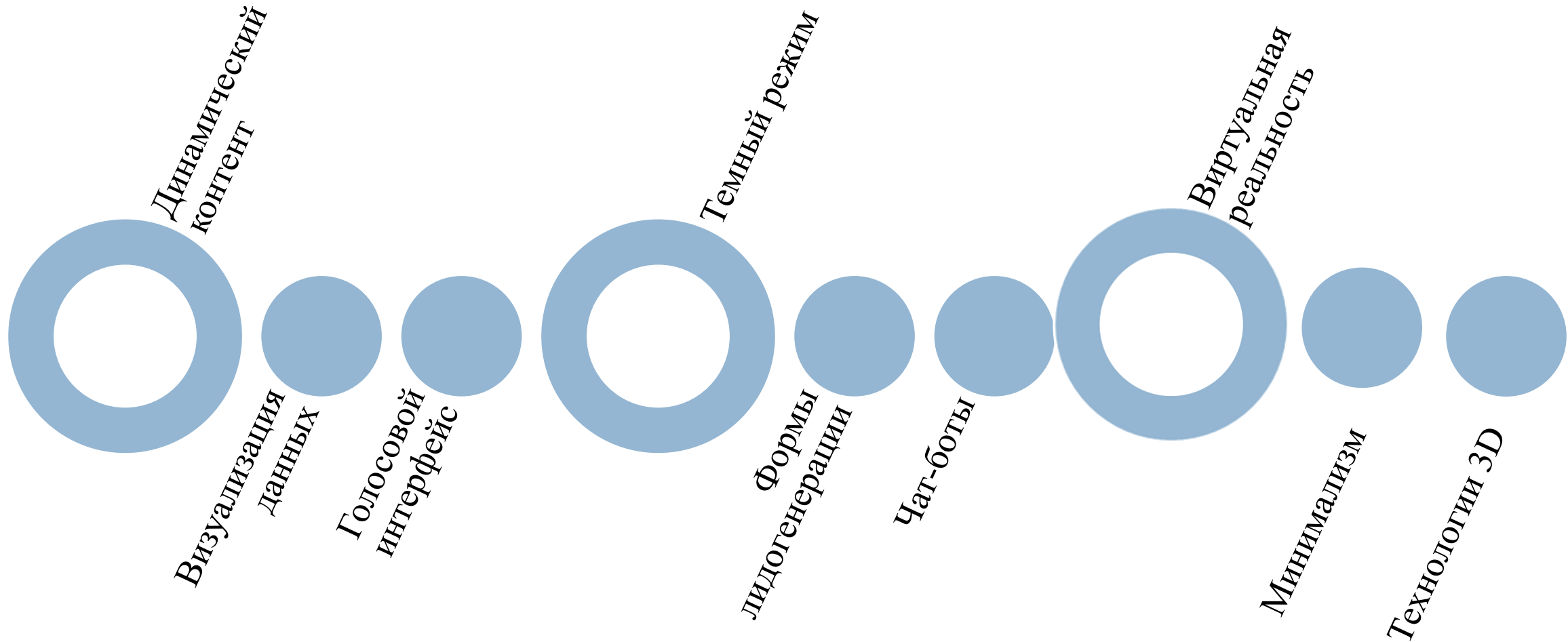
3. *Выбор платформы для внедрения*

Когда возникает идея создания качественного веб-сайта, следует задаться вопросом, какую платформу выбрать. Существует множество разновидностей CMS в разработке электронной коммерции, однако нужно выбрать лучшие варианты для создания веб-сайта.

4. *Создание прототипа и макета веб-сайта.*

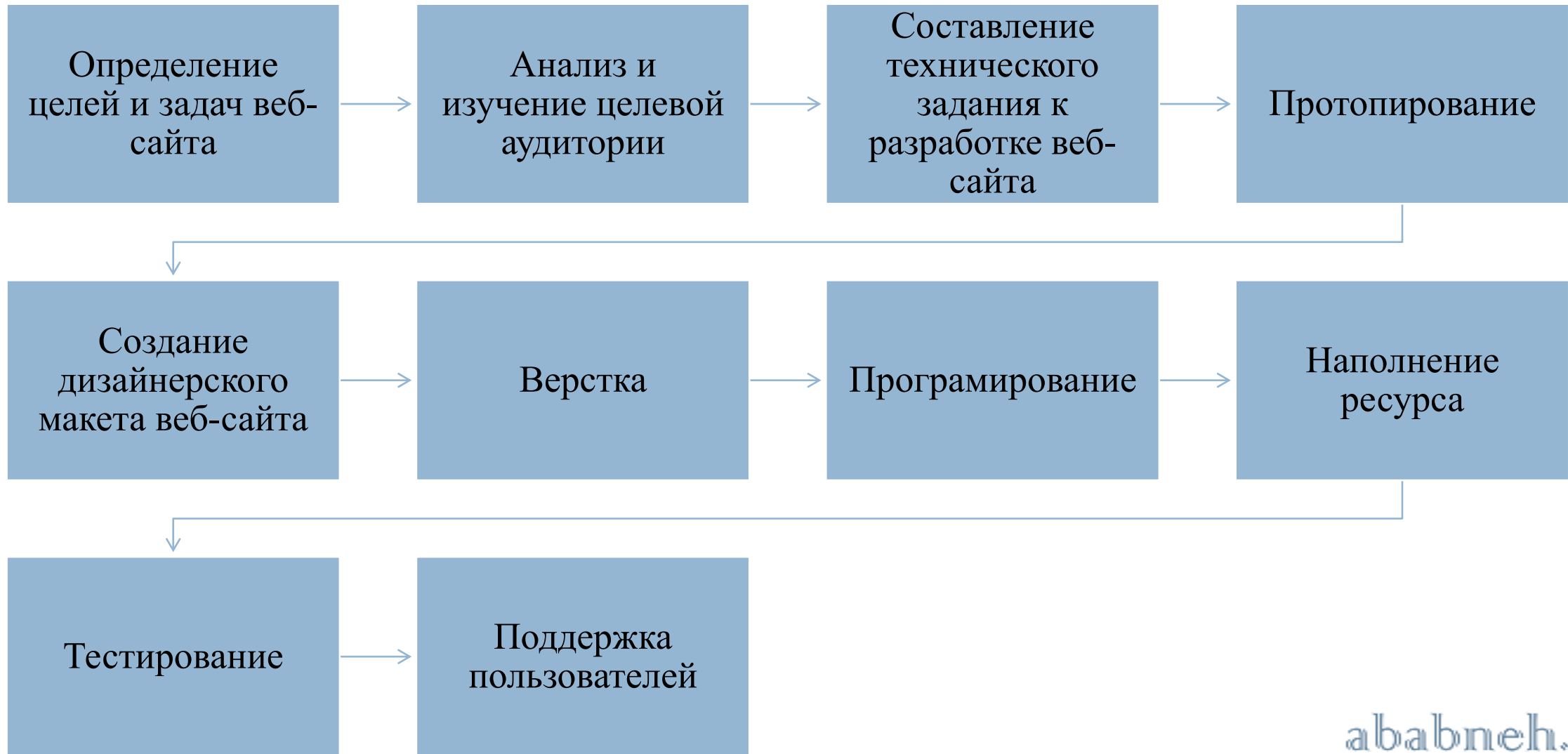
Этот этап позволяет определить информационную архитектуру и объясняет взаимосвязь между различными страницами и элементами контента. Создание веб-сайта без прототипа сайта похоже на строительство дома без чертежа, и вряд ли это удастся.

КЛАССИФИКАЦИЯ КЛЮЧЕВЫХ ТРЕНДОВ ВЕБ-ДИЗАЙНА СОВРЕМЕННЫХ ВЕБ-САЙТОВ



2. ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА

СТРУКТУРИЗАЦИЯ КЛЮЧЕВЫХ ЭТАПОВ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ВЕБ-САЙТОВ В СОВРЕМЕННЫХ УСЛОВИЯХ



ОСНОВНЫЕ КОМПОНЕНТЫ ДЛЯ ОПРЕДЕЛЕНИЯ ЦЕЛЕВОЙ АУДИТОРИИ, ТЕМАТИКИ И НАПРАВЛЕНИЯ СОВРЕМЕННОГО ВЕБ-САЙТА

Обратная связь
и организация
процесса
взаимодействи
я

Цифровые
каналы для
коммуникаци
и

Какие типы и
виды контента
будут
использоваться
для
продвижения
веб-сайта

Какая
информация и
контент будет
размещаться
для
привоечения
аудитории

Какие
предпочтения и
ожидания
покупателей и
потенциальных
клиентов

Целевая
аудитория и
направленность
веб-сайта
компании

ТРЕБОВАНИЯ НА КОТОРЫХ ДОЛЖНО БАЗИРОВАТЬСЯ ТЕХНИЧЕСКОЕ ЗАДАНИЕ К РАЗРАБОТКЕ СОВРЕМЕННОГО ВЕБ-САЙТА

*Детализация требований к
разработке веб-сайта.*

Все особенности и специфика функционирования веб-сайта должна быть определена и зафиксирована в техническом задании.

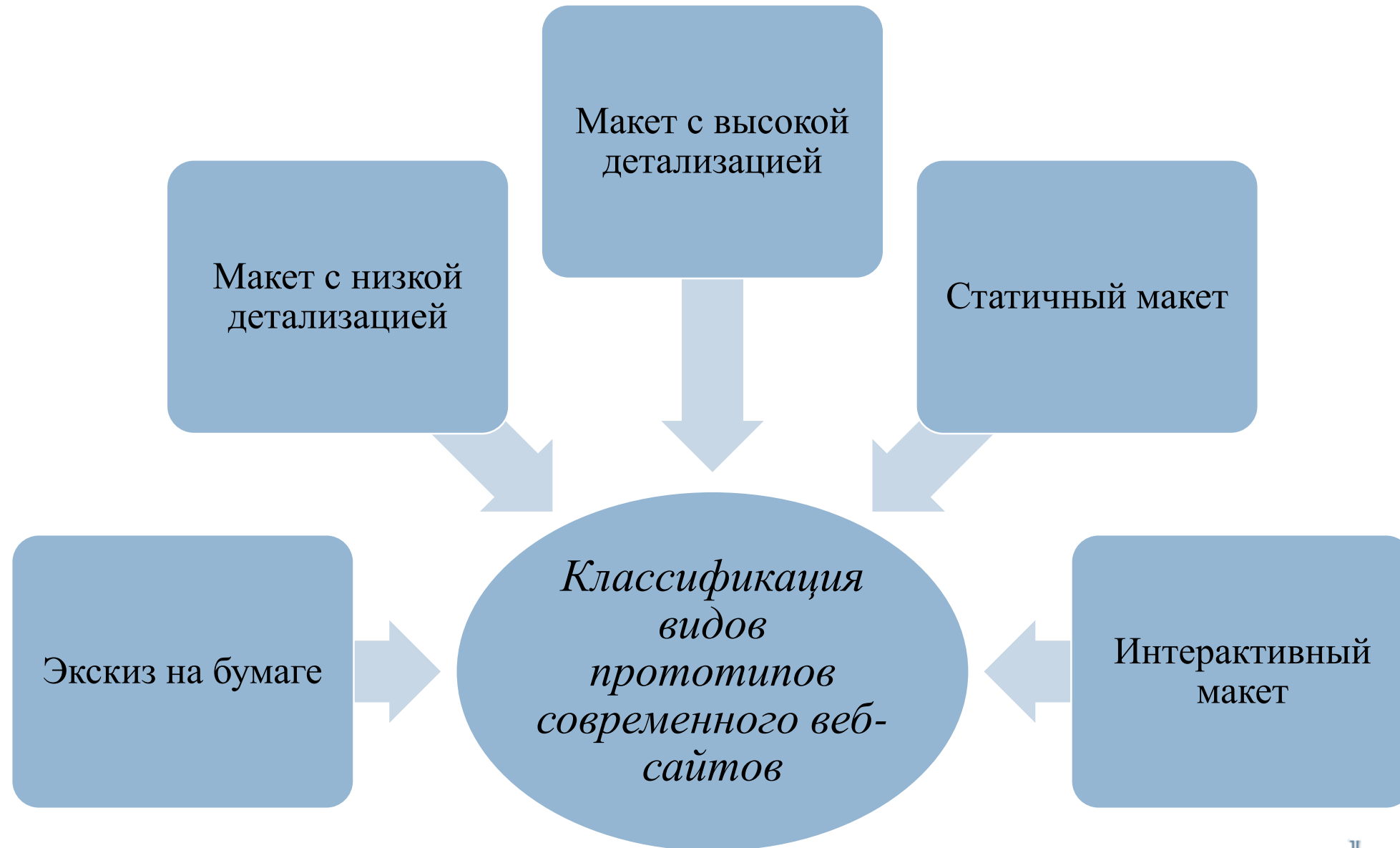
*Точность описания
требований к разработке веб-
сайта.*

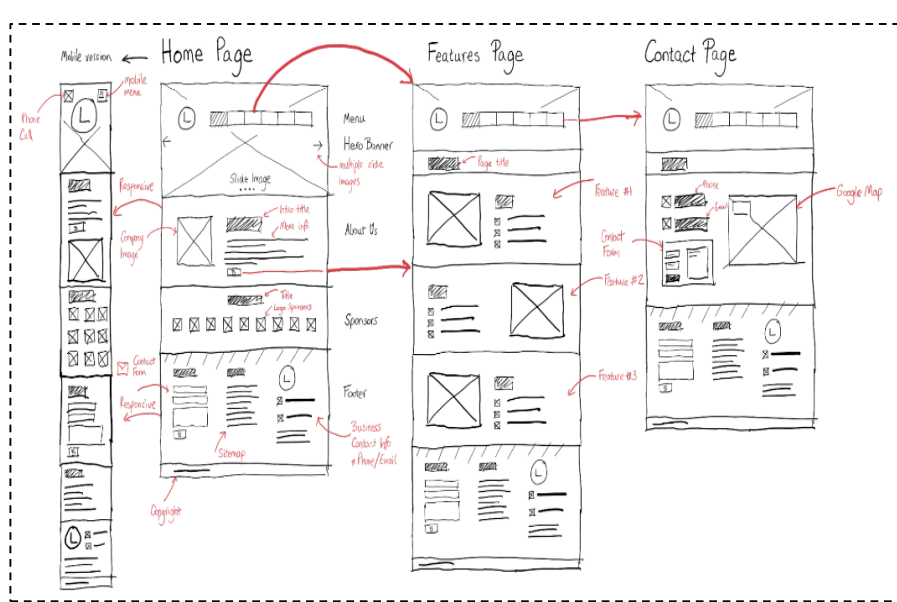
Техническое задание должно включать в себя точное описание всех функциональных, нефункциональных требований к веб-сайту, которые ожидаются.

*Доступность изложения
требований.*

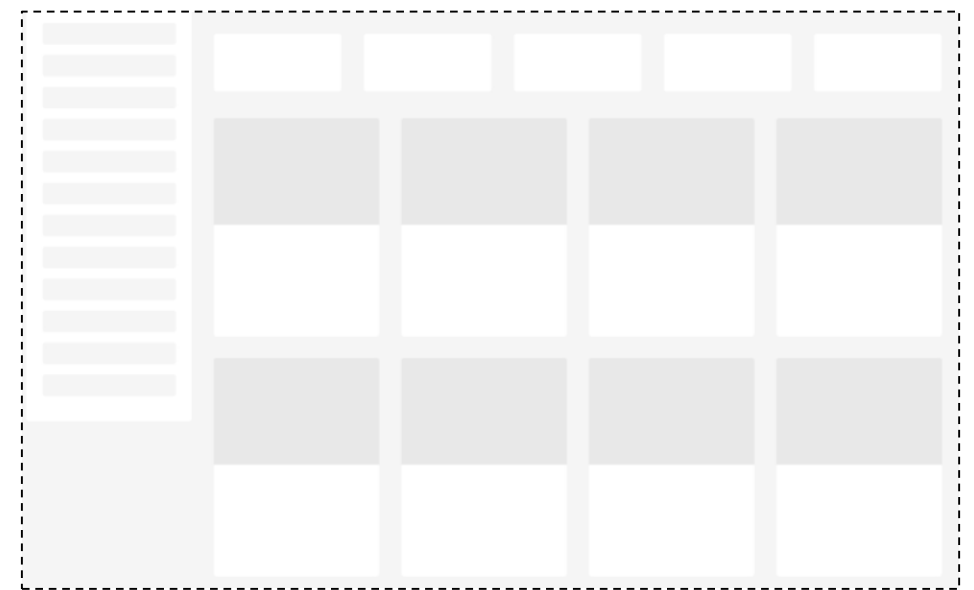
Техническое задания к разработке веб-сайта должно быть понятным и доступным для всех участников проекта.

КЛАССИФИКАЦИЯ ВИДОВ ПРОТОТИПОВ СОВРЕМЕННЫХ ВЕБ-САЙТОВ

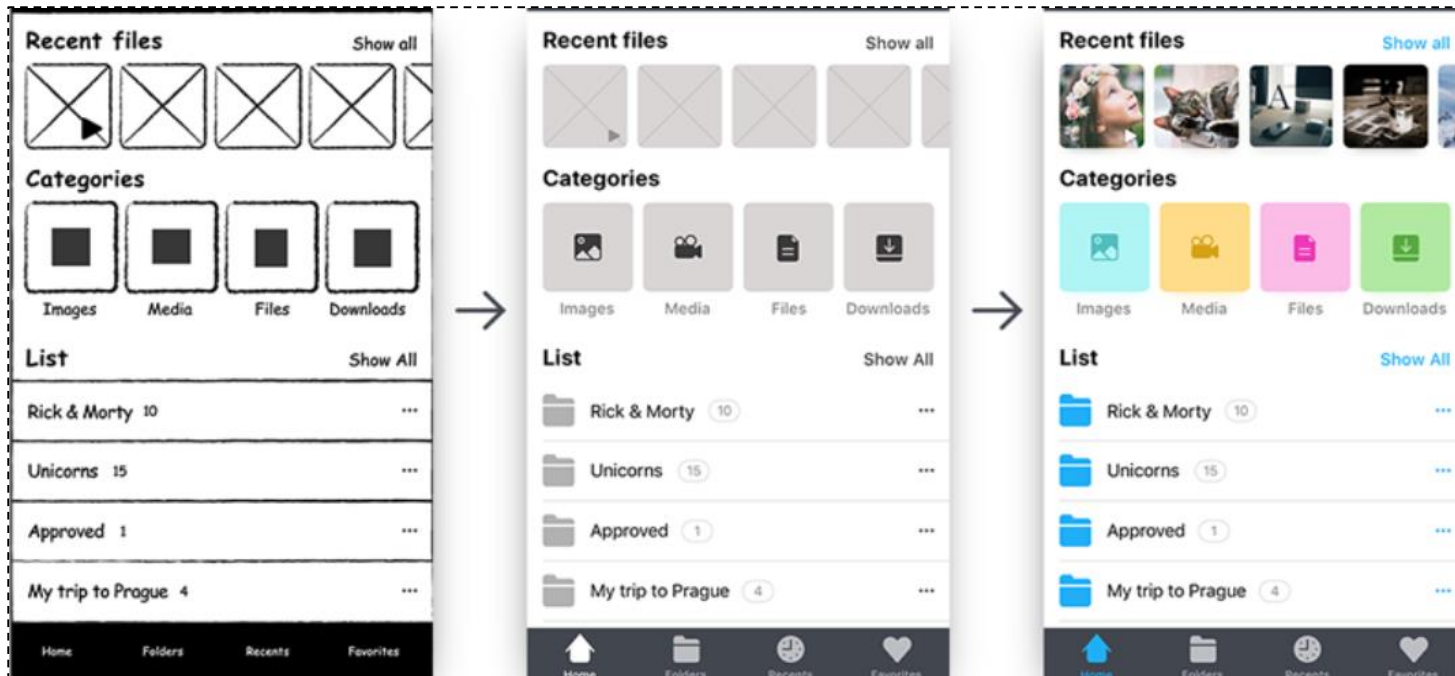




Визуализация эскиза прототипа современного веб-сайта на бумаге

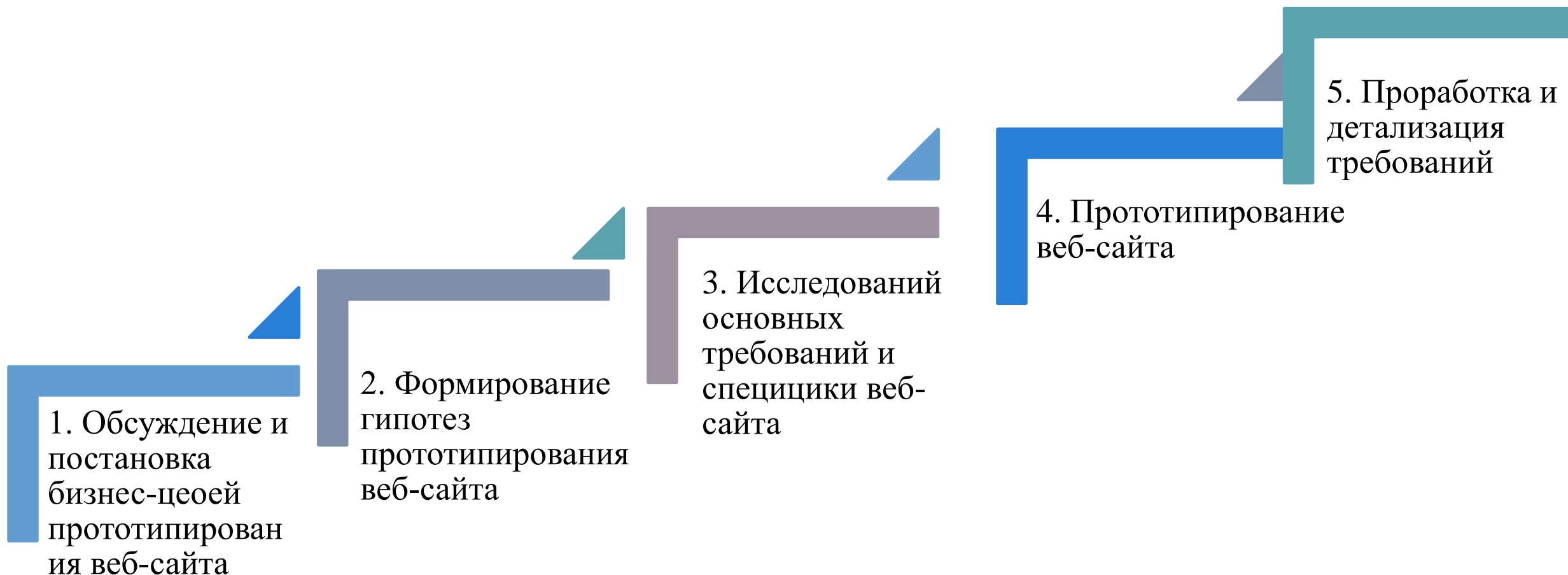


Визуализация прототипа современного веб-сайта при помощи эскиза с низкой детализацией



Визуализация прототипа современного веб-сайта при помощи эскиза с высокой детализацией

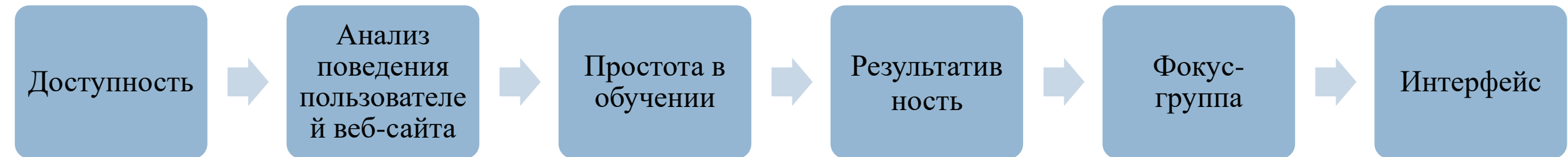
СТРУКТУРИЗАЦИЯ КЛЮЧЕВЫХ ЭТАПОВ РАЗРАБОТКИ ПРОТОТИПА СОВРЕМЕННОГО ВЕБ-САЙТА



КЛАССИФИКАЦИЯ ОСНОВНЫХ ПРОГРАММ ДЛЯ ПРОТОТИПИРОВАНИЯ СОВРЕМЕННЫХ ВЕБ- САЙТОВ



ОСНОВНЫЕ ЭЛЕМЕНТЫ ПРОЦЕССА РАЗРАБОТКИ И ПРОЕКТИРОВАНИЯ UX И UI ВЕБ-ДИЗАЙНА СОВРЕМЕННОГО ВЕБ-САЙТА



КЛЮЧЕВЫЕ БИЗНЕС-ЗАДАЧИ РЕШЕНИЕ КОТОРЫХ ОБЕСПЕЧИВАЕТ ПРОЕКТИРОВАНИЯ СОВРЕМЕННЫХ ВЕБ- САЙТОВ

Повышение вероятности достижения желаемого вида веб-ресурса.

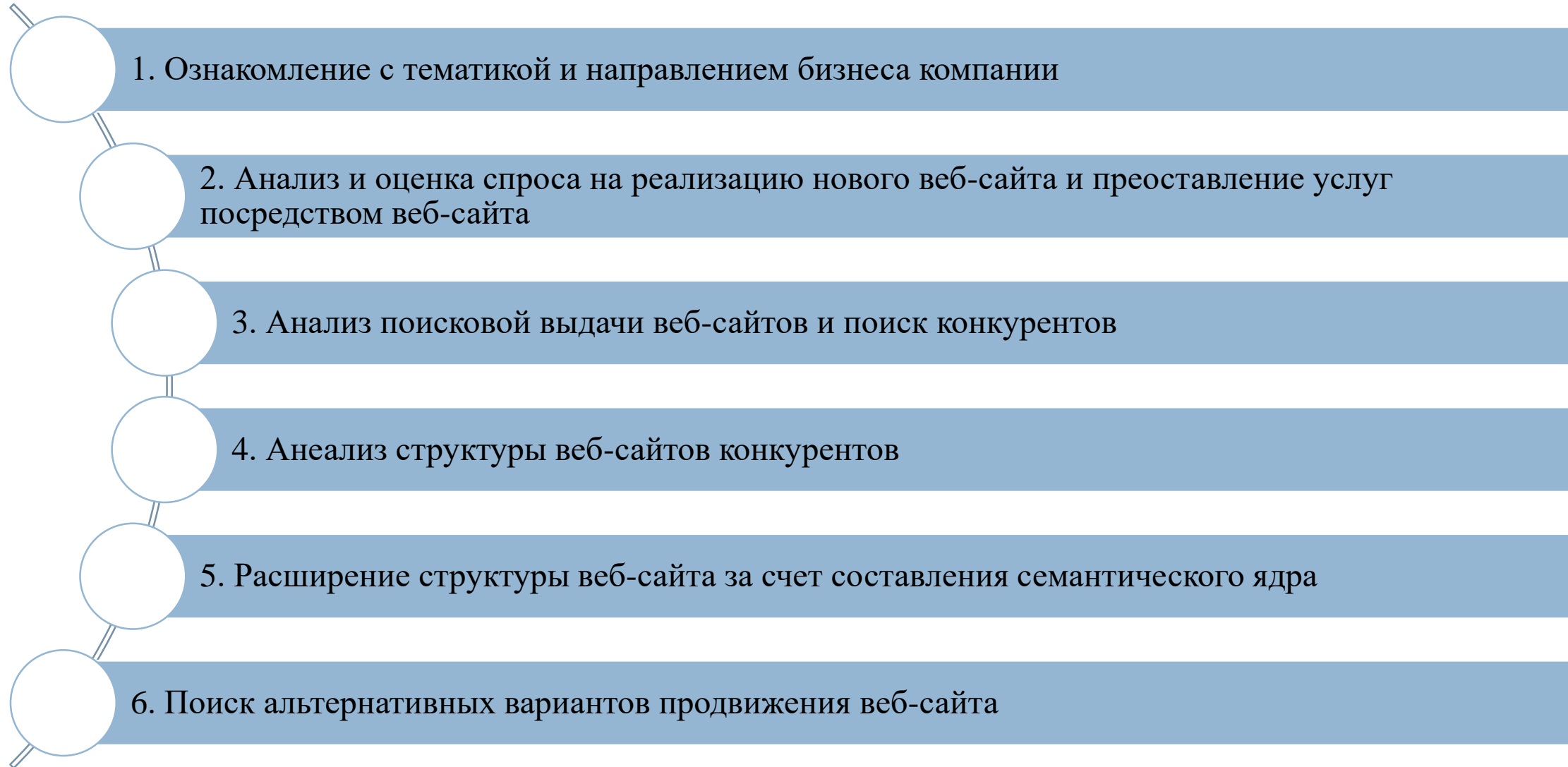
Четкая постановка задач, определение целевой аудитории и моделирование пользовательских путей может получить желаемый от разработки результат.

Логичное разделение работ по разработке веб-сайта .

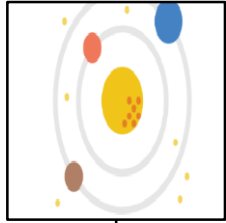
Прототип сайта – это самостоятельный документ, на основе которого заказчик уже может создать веб-сайт, самостоятельно или силами другой команды разработчиков.

Оптимизация времени и сохранность финансов.

Устранение ошибок при создании прототипа сайта – простой процесс, так как достаточно изменить некоторые части текста или графики. Реализация таких действий займет гораздо больше времени и будет дороже на этапе согласования. Для веб-программиста такие исправления будут более трудоемкими.

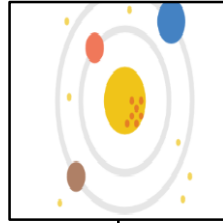


КЛЮЧЕВЫЕ ОСОБЕННОСТИ И ПРЕИМУЩЕСТВА СЕМАНТИЧЕСКОГО ПРОЕКТИРОВАНИЯ СОВРЕМЕННОГО ВЕБ-САЙТА



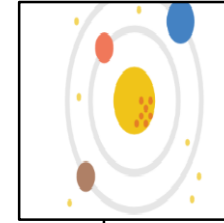
Преимущество для SEO специалистов

- готовое семантическое ядро;
- готовая стратегия продвижения в поисковых системах;
- расставлены приоритеты по подготовке контента;
- Спланирован процесс разработки и внедрения веб-сайта



Преимущества для WEB разработчика

- готовая структура веб-сайта;
- рекомендации SEO специалиста касательно разработки веб-сайта;
- минимизация ошибок;
- Уменьшение времени разработки веб-сайта.



Преимущества для владельца бизнеса

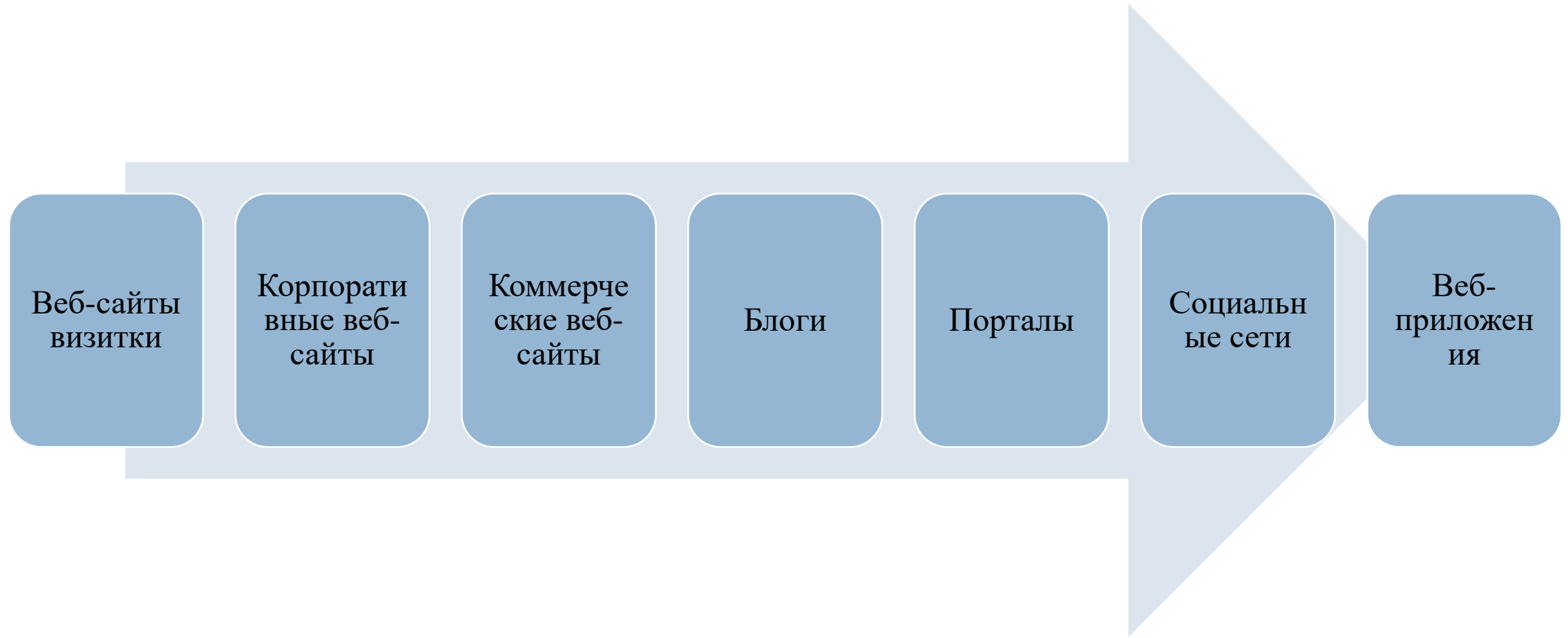
- удешевление разработки веб-проекта;
- ускорение процесса ожидания результативности внедрения;
- ускорение получение прибыли по результатам внедрения веб-сайта.

3. ФУНКЦИОНАЛЬНОСТЬ ВЕБ-САЙТА

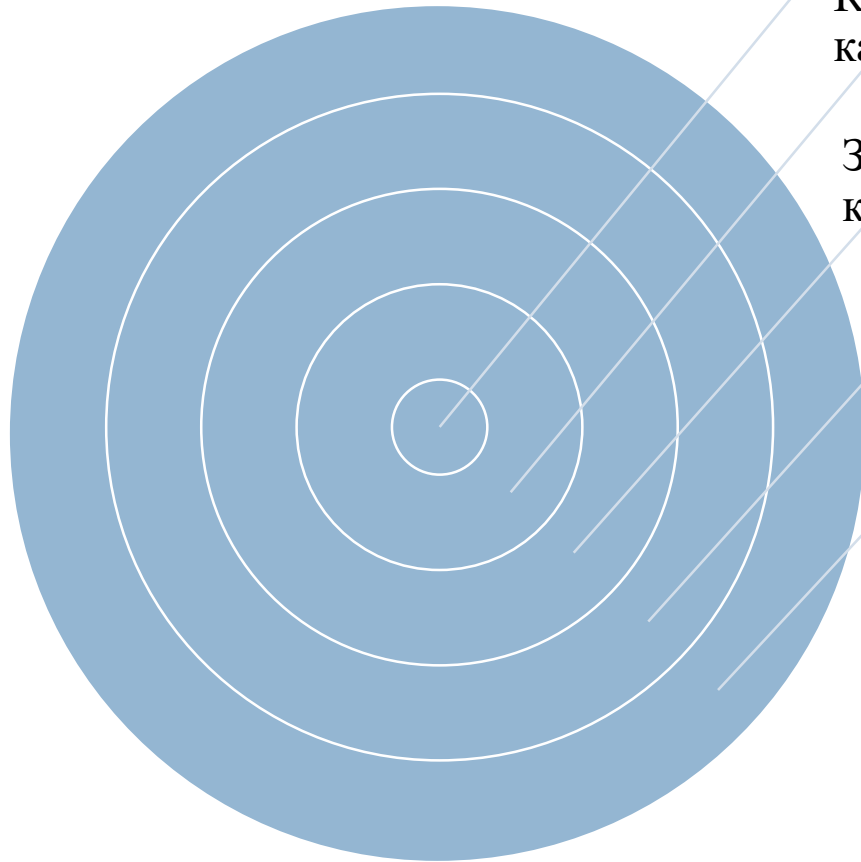
Ключевые характеристики и компоненты функциональности современного веб-сайта



КЛАССИФИКАЦИЯ СОВРЕМЕННЫХ ВЕБ-САЙТОВ В ЗАВИСИМОСТИ ОТ ИХ НАЗНАЧЕНИЯ И ФУНКЦИОНАЛЬНОСТИ



КЛЮЧЕВЫЕ ЭЛЕМЕНТЫ, КОТОРЫЕ ПОЗВОЛЯТ УЛУЧШИТЬ ФУНКЦИОНАЛЬНОСТЬ СОВРЕМЕННОГО ВЕБ- САЙТА



Понятный веб-дизайн и
пользовательский интерфейс

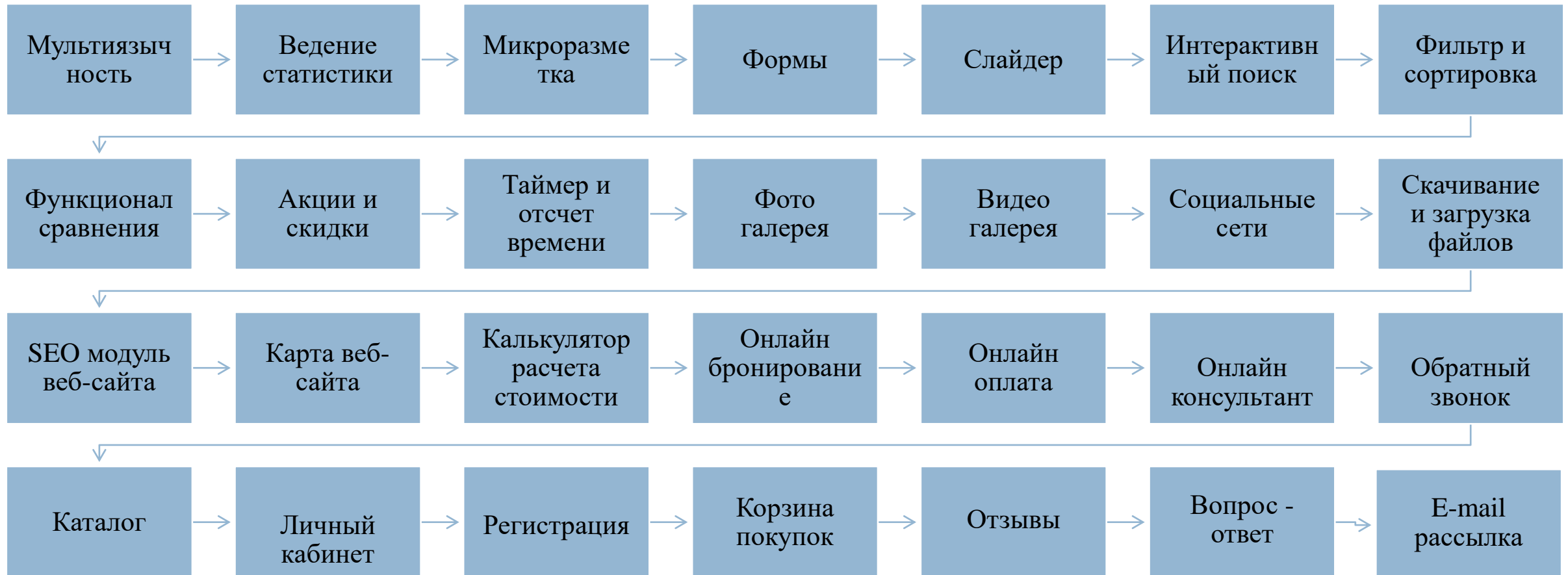
Качественное изображение и
качественное описание продуктов

Заметные, но не навязчивые призывы
к действию

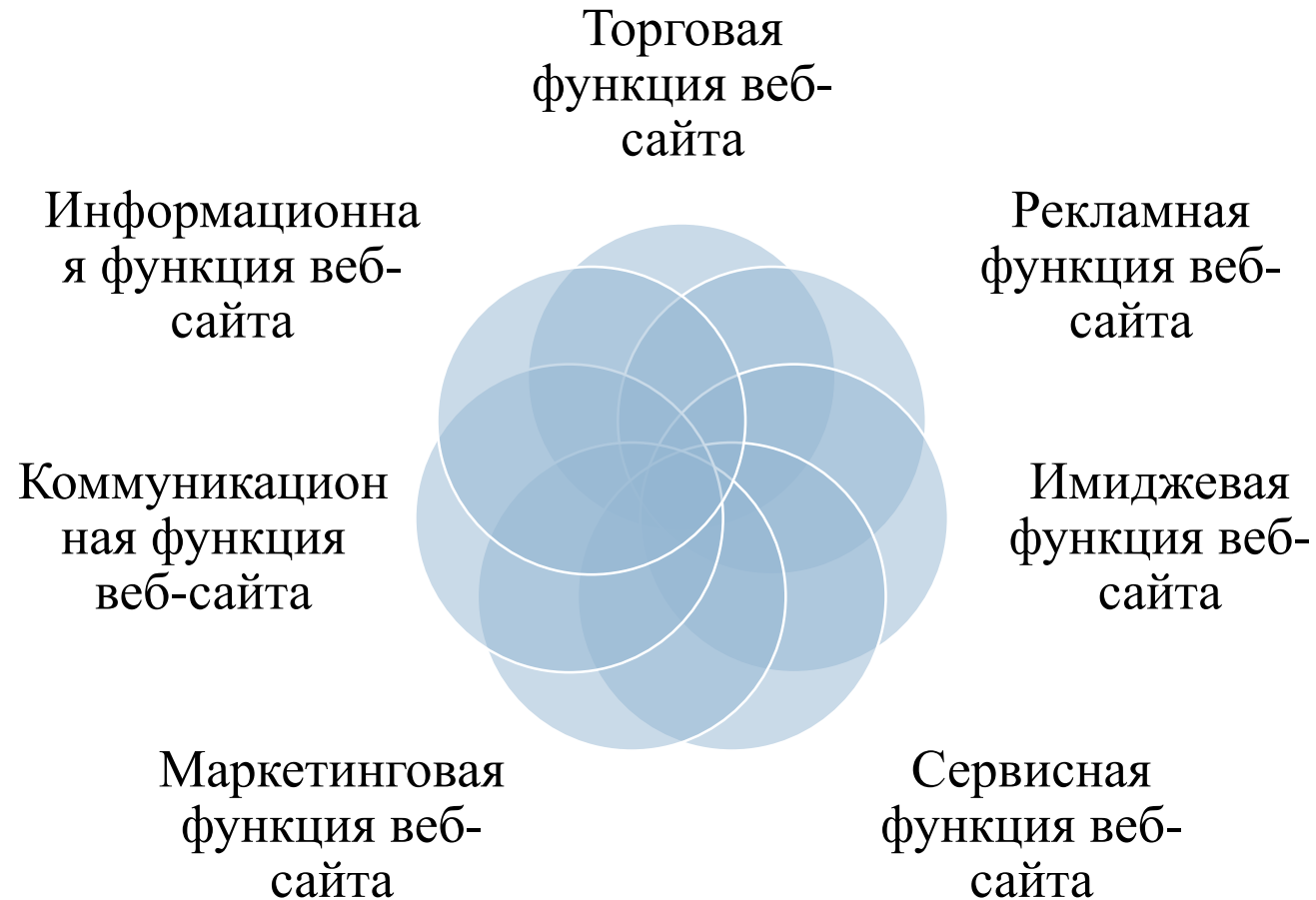
Ведение блога компании для
улучшения доверия к бренда

Простая и четкая навигация
веб-сайта

СТРУКТУРИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ФУНКЦИОНАЛА И ОСОБЕННОСТЕЙ НАСТРОЙКИ СОВРЕМЕННЫХ ВЕБ-САЙТОВ



КЛЮЧЕВЫЕ ФУНКЦИИ СОВРЕМЕННЫХ ВЕБ-САЙТОВ В ЗАВИСИМОСТИ ОТ НАПРАВЛЕНИЯ И ВИДА ДЕЯТЕЛЬНОСТИ КОМПАНИИ



4. ВЕРСТКА ВЕБ-САЙТА

Требования к коду, написанному для разработки сайта:

Первое требование: формат должен быть корректным: грубые смысловые ошибки не допускаются, так как это приводит к проблемам в продвижении сайта в поисковых системах, а эффект ощутим.

Второе требование: Сайт должен быть мультибраузерным: то есть выглядеть одинаково во всех популярных версиях браузеров, используемых целевой аудиторией и клиентами.

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ИСПОЛЬЗОВАНИЯ CSS ВЁРСТКИ СОВРЕМЕННЫХ ВЕБ-САЙТОВ

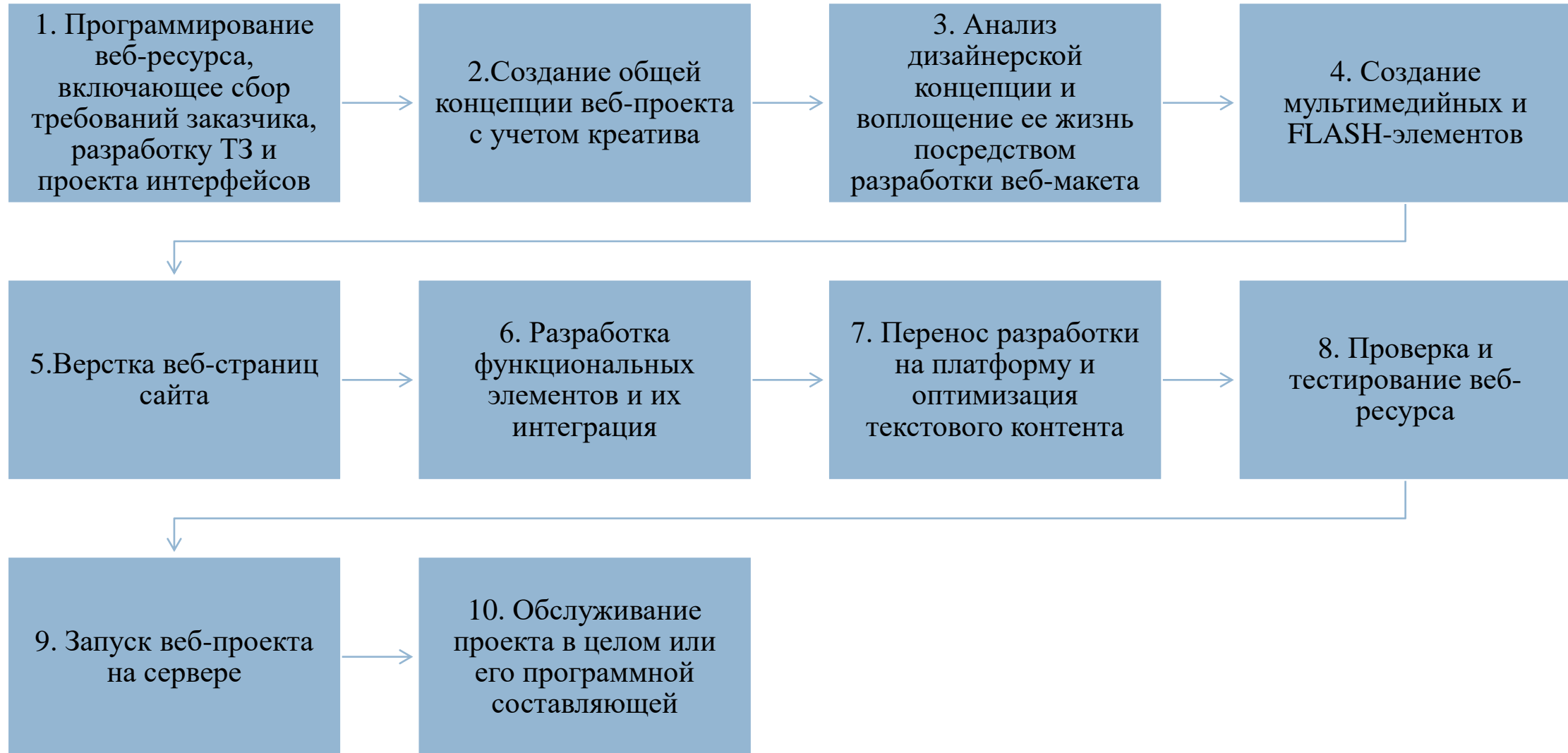
Преимущества

- Реализация нескольких веб-дизайнов страницы для разных устройств просмотра.
- Уменьшение времени загрузки страниц веб-сайта за счёт переноса правил представления данных в отдельный CSS-файл. В этом случае браузер загружает только структуру документа и данные, хранимые на странице, а представление этих данных загружается браузером только один раз и может быть закешировано.
- Простота последующего изменения веб-дизайна. Не нужно править каждую веб-страницу, а достаточно лишь изменить CSS-файл.
- Дополнительные возможности оформления веб-дизайна страниц веб-сайта.

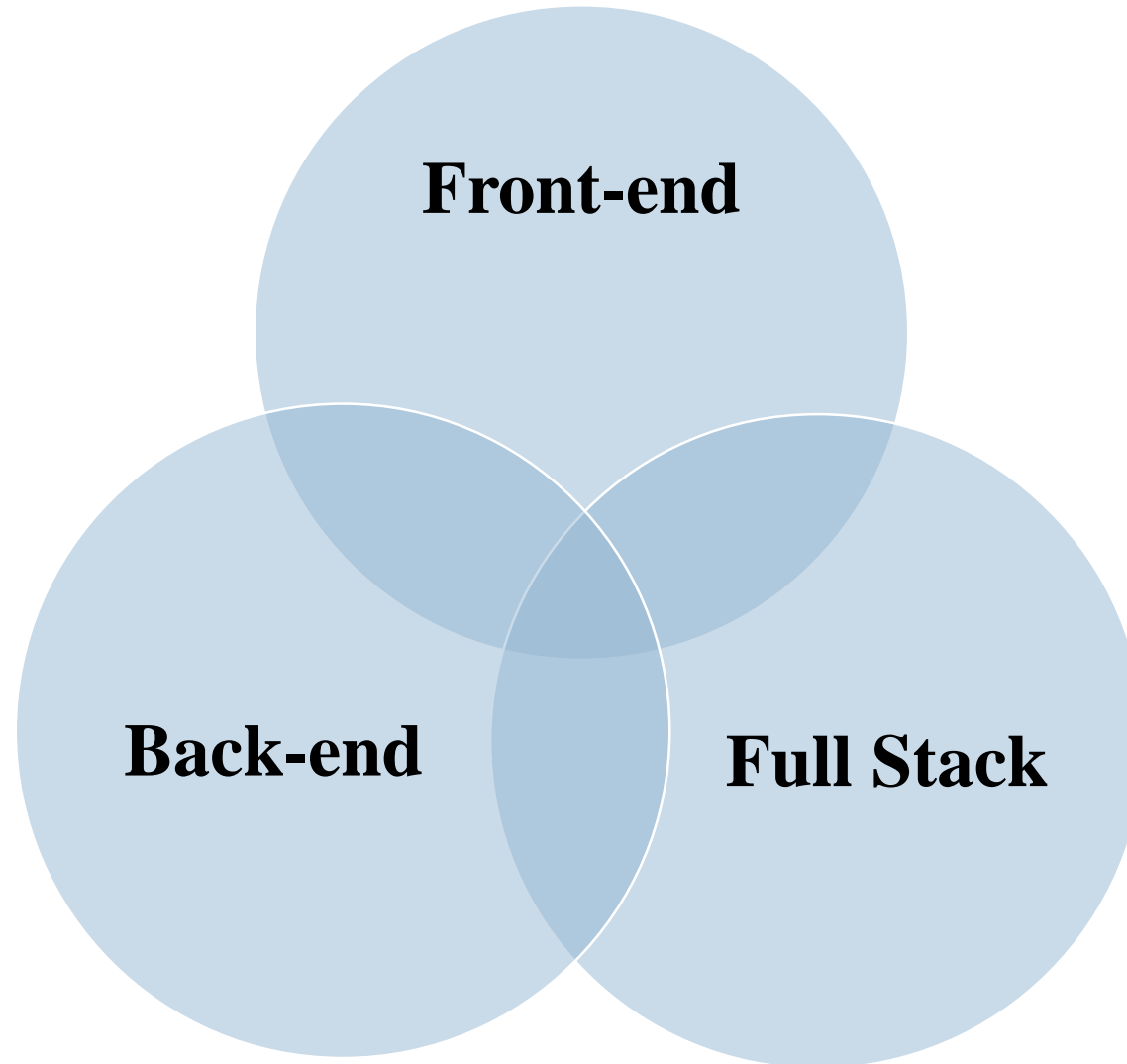
- Различное отображение вёрстки в различных браузерах, которые по-разному интерпретируют одни и те же данные CSS.
- Часто встречающаяся необходимость на практике исправлять не только один CSS-файл, но и теги HTML, которые сложным и ненаглядным способом связаны с селекторами CSS, что иногда сводит на нет простоту применения единых файлов стилей и значительно увеличивает время редактирования и тестирования.

Недостатки

КЛЮЧЕВЫЕ ЭТАПЫ ВЕБ-ПРОГРАММИРОВАНИЯ ВЕБ-САЙТОВ В СОВРЕМЕННЫХ УСЛОВИЯХ



КЛАССИФИКАЦИЯ ВИДОВ ВЕБ-РАЗРАБОТКИ В СОВРЕМЕННЫХ УСЛОВИЯХ



Разработка веб-интерфейса должна включать и основываться на ключевых компонентах:

HTML (язык гипертекстовой разметки) — язык разметки, используемый при создании и дизайне веб-страниц и веб-сайтов. Этот язык считается одним из старейших языков и широко используется при разработке веб-страниц для всех элементов и документов на веб-странице и их взаимодействия в структуре веб-страницы.

CSS (каскадные таблицы стилей) — язык форматирования веб-страниц, который заботится о внешнем виде и дизайне веб-сайтов. Он был специально разработан для отделения форматирования от содержания письменного документа. Это относится к цветам, шрифтам, изображениям и фонам, которые используются на страницах с полной гибкостью и легкостью. Эта технология связана с общим внешним видом веб-страниц, включая цвета, изображения и т. д. Она также адаптирует один и тот же документ в разных стилях и стилях. отображает передачу на экране или для чтения голосом.

JavaScript — язык программирования, который разработчики используют для создания интерактивных веб-страниц, чтобы оживить их. От обновления лент социальных сетей до отображения анимации и интерактивных карт — его функциональные возможности могут улучшить взаимодействие пользователя с веб-сайтом.

- **PHP.** Это скриптовый язык, в первую очередь предназначенный для разработки и программирования веб-приложений. Его также можно использовать для создания автономных программ, а не только для веба. Открытый исходный код и разработанный командой волонтеров, он является одним из самых популярных языков в мире.
- **Python.** Это простой в изучении, открытый исходный код, расширяемый, высокоуровневый, объектно-ориентированный язык программирования. Это интерпретируемый и универсальный язык, который широко используется во многих областях, таких как создание автономных программ с использованием графических интерфейсов и веб-приложений, и его можно использовать в качестве скриптового языка для управления производительностью многих программ с возможностью использовать его для создания простых программ для новичков и выполнения огромных проектов одновременно.
- **Ruby.** Это высокоуровневый интерпретируемый язык программирования общего назначения, который поддерживает множество парадигм программирования. Он был разработан с упором на производительность и простоту программирования. В Ruby все является объектом, включая примитивные типы данных, они динамически типизированы и используют сборку мусора и компиляцию JIT.
- **SQL.** Это язык программирования для хранения и управления информацией в реляционной базе данных. Реляционная база данных хранит информацию в виде таблицы, в которой строки и столбцы представляют различные атрибуты данных и различные отношения между значениями данных, с возможностью использования операторов SQL для хранения, обновления, удаления, поиска и извлечения информации из базы данных.
- **Go (Golang)** Это скомпилированный многопоточный язык программирования с открытым исходным кодом от Google. Он считается языком общего назначения, но в основном используется для разработки веб-сервисов и клиент-серверных приложений. Он стал одним из самых быстрорастущих серверных языков программирования благодаря своей простоте, а также способности работать с многоядерными системами, сетями и огромными базами кода.
- **Java.** Это высокоуровневый, основанный на классах, объектно-ориентированный язык программирования, разработанный так, чтобы иметь как можно меньше зависимостей от реализации.

| Элементы Back-end веб-разработки | Особенности |
|--|---|
| Серверная часть | <p>У каждого веб-сайта или веб-приложения есть две стороны - клиент и сервер. Клиент - это оболочка приложения или браузер, то есть то, с чем взаимодействует пользователь, и откуда отправляются запросы на сервер. А сервер - это по сути машина, которая обрабатывает эти запросы и возвращает нужную информацию. При этом один сервер может обрабатывать запросы множества клиентов, как и один клиент может обращаться ко многим серверам.</p> |
| База данных | <p>Сервер только обрабатывает информацию, но не хранит ее для этого есть специальные инструменты - базы данных. При входе в любую систему также используется база данных: в ней хранятся все логины и пароли, и когда вводятся данные при входе, сервер отправляет запрос в базу, чтобы проверить, если там такая пара логин-пароль; при положительном ответе – возможно зайти и получаете доступ в систему, в которую совершался вход.</p> |
| Application Programming Interface | <p>Application Programming Interface - это та часть сервера, с которой непосредственно взаимодействует клиент, когда речь идет только об одном сервере и одном клиенте в вакууме - все кажется довольно просто. Однако в на практике такое редко бывает и приходится интегрировать программы между собой, например, вставить Google-карты на веб-сайт, чтобы пользователь легко мог найти любой офлайн магазин компании и построить маршрут к нему.</p> |

СТРУКТУРА ОСНОВНЫХ КОМПОНЕНТОВ FULL-STACK ВЕБ-РАЗРАБОТКИ В СОВРЕМЕННЫХ УСЛОВИЯХ

Back-end

Project management

*Структура основных
компонентов Full-stack веб-
разработки в современных
условиях*

Front-end

DevOps

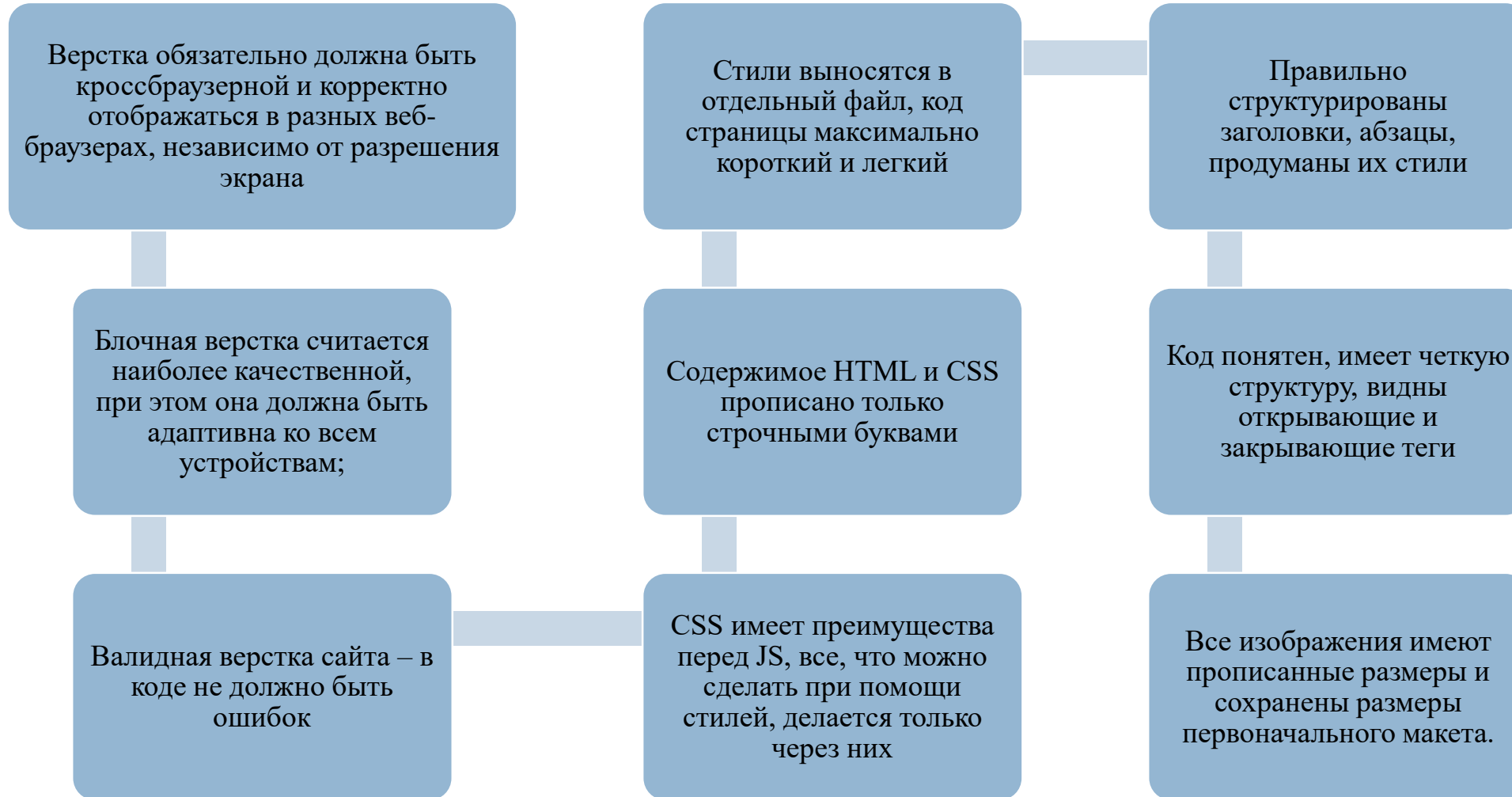
КЛАССИФИКАЦИЯ СОВРЕМЕННЫХ ВИДОВ ВЕРСТКИ ВЕБ-САЙТОВ КОМПАНИЙ В СОВРЕМЕННЫХ УСЛОВИЯХ

| Вид верстки веб-сайтов | Особенности верстки веб-сайтов |
|------------------------|--|
| Адаптивная верстка | Подразумевает корректное отображение на любых видах устройств. На сегодняшний день она наиболее популярная. Является неотъемлемой частью двух других видов, поэтому ее гораздо реже выделяют, как отдельный вид. |
| Табличная верстка | Когда научились верстать веб-сайты, появился этот метод. Табличная верстка веб-дизайна веб-сайта подразумевает использование таблиц, наполненных информацией. Внешне процесс напоминает работу в Excel. Данный метод считается уже устаревшим, но все равно активно используется. Дело в том, что он подходит для типовых проектов с возможностью детализации и индивидуализации. |
| Блочная верстка | Это современный универсальный вид верстки веб-сайтов, который активно используется для создания качественных веб-страниц. Принцип его заключается в том, что все элементы располагаются в блоках, или контейнерах. Они содержат необходимую информацию и сами по себе являются регулируемы. Можно задавать их размер, цвет и прочие параметры. Блочная верстка дизайн макета проще, имеет больше возможностей и способна реализовать любые идеи дизайнера. Код получается чище и легче, он удобно читается браузерами, что влияет на ранжирование сайта в поисковых системах. Кроме того, блоки гарантируют адаптивный дизайн, поэтому веб-сайт корректно отображается на различных устройствах. |

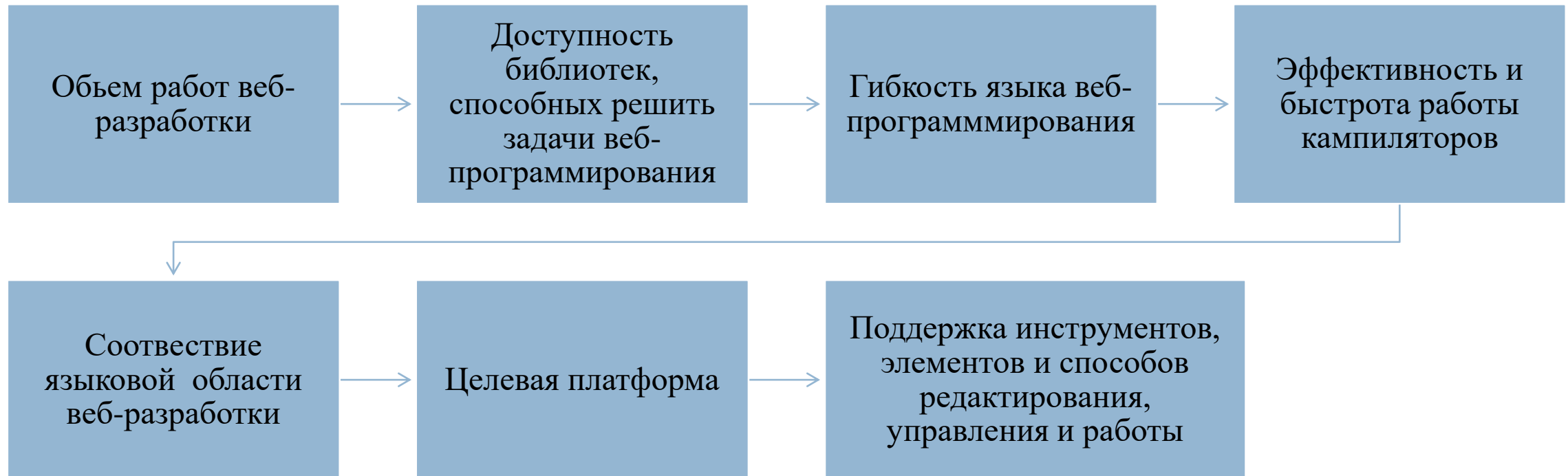
СТРУКТУРИЗАЦИЯ ОСНОВНЫХ ИНСТРУМЕНТОВ ДЛЯ ВЕРСТКИ ВЕБ-САЙТОВ КОМПАНИЙ В СОВРЕМЕННЫХ УСЛОВИЯХ

| Направления применения инструментов для верстки веб-сайтов | Инструменты для верстки веб-сайтов |
|---|---|
| Обработка изображений | Adobe Photoshop, Gimp, Krita и прочие приложения, позволяющие из макета взять элементы и работать с визуальным контентом. |
| Работа с кодом программирования | Notepad++, UltraEdit, Adobe DreamViewer, CSS3 Generator, SublimeText, Firebug, CoffeeCup HTML Editor, Winless, WinMerge и прочие. |
| Работа с JS | Front Page, NetBeans и много других инструментов |
| Проверка работоспособности | Crossbrowsertesting, Markup validtor, IE Tester, Validator.w3, Dr Watson, Css validator и прочие. |

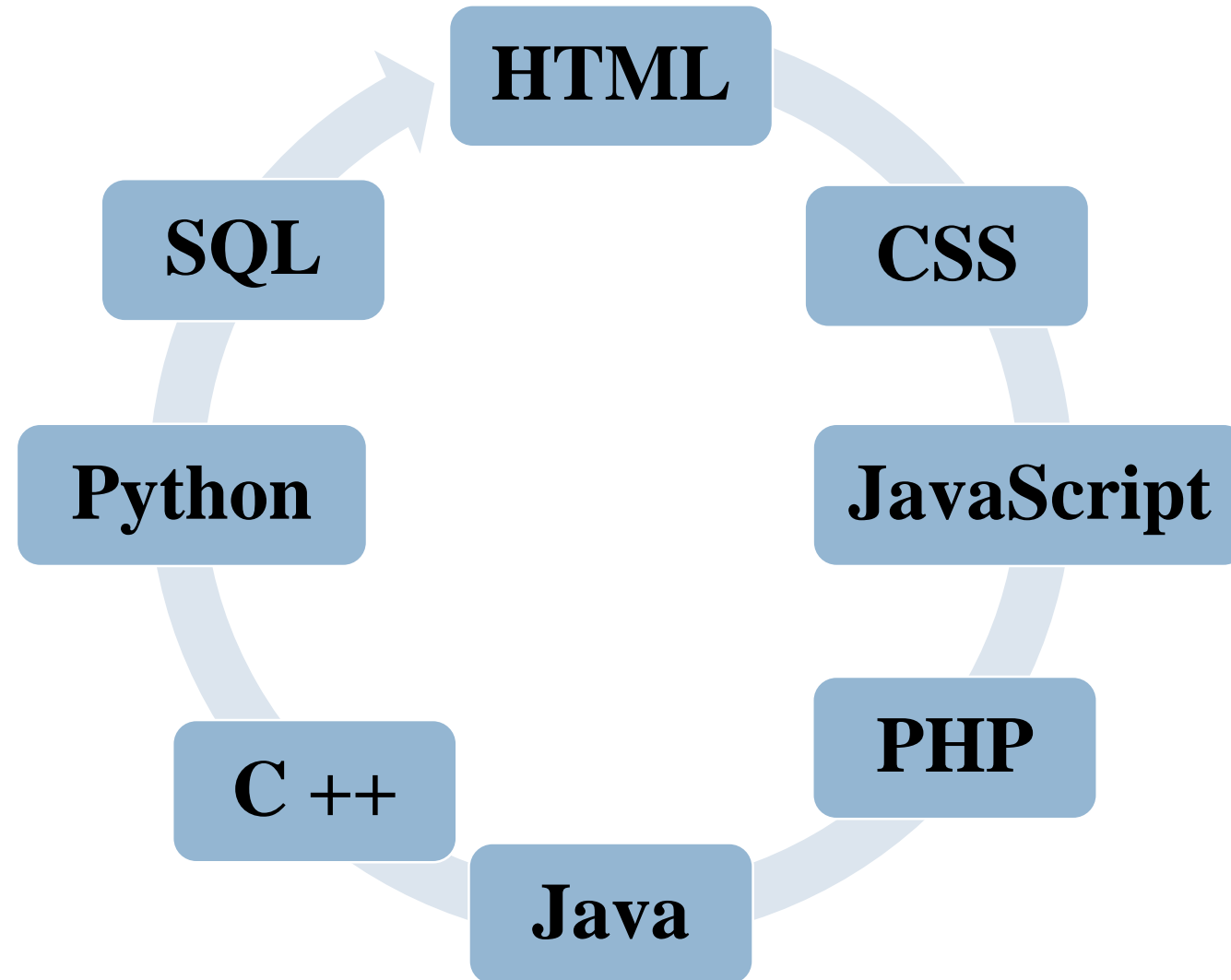
ОСНОВНЫЕ ПРИЗНАКИ ПРАВИЛЬНОЙ ВЕРСТКИ ВЕБ-САЙТА КОМПАНИЙ В СОВРЕМЕННЫХ УСЛОВИЯХ

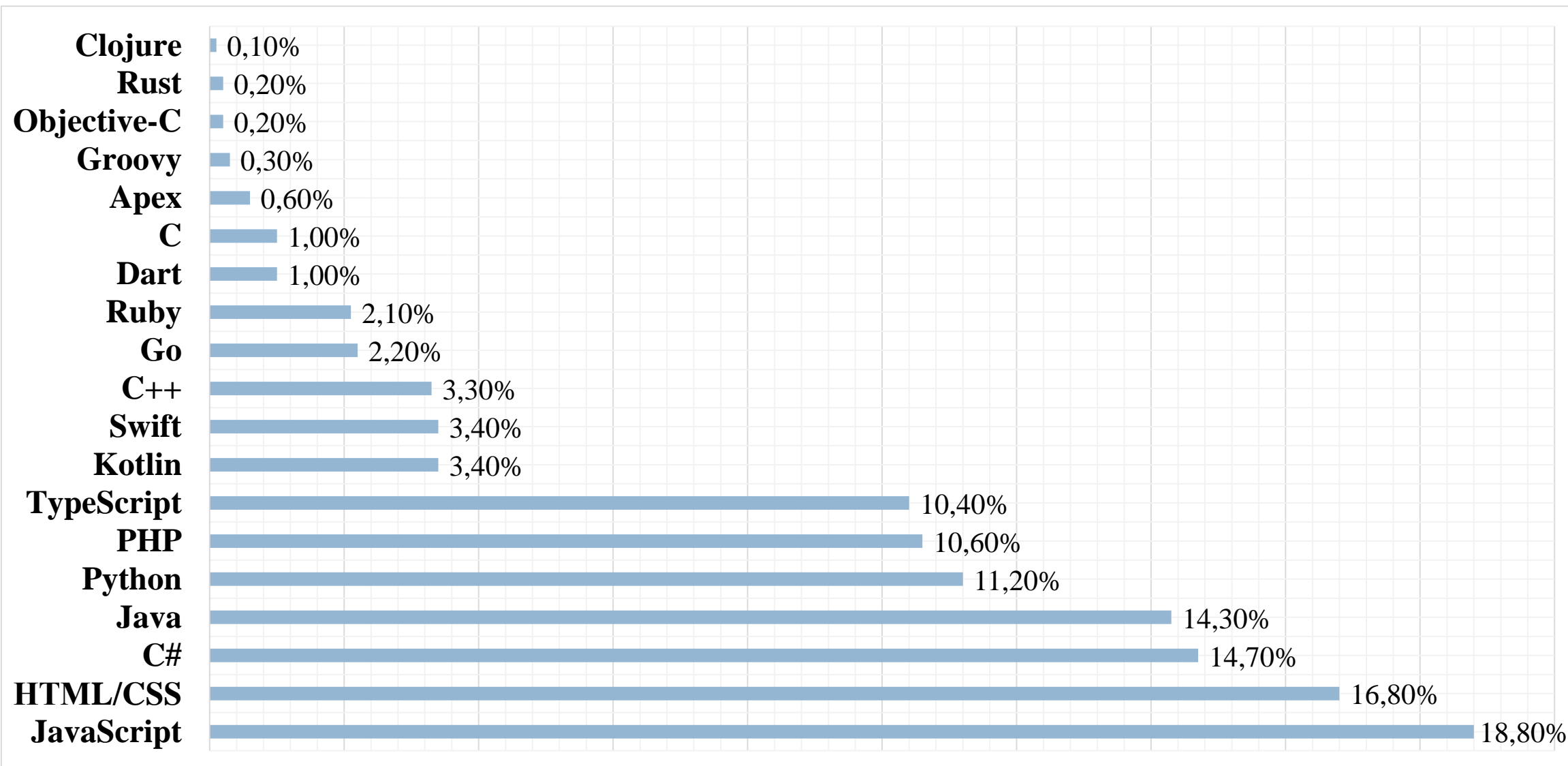


Ключевые факторы, которые влияют на принятие решения о выборе языка веб-программирования

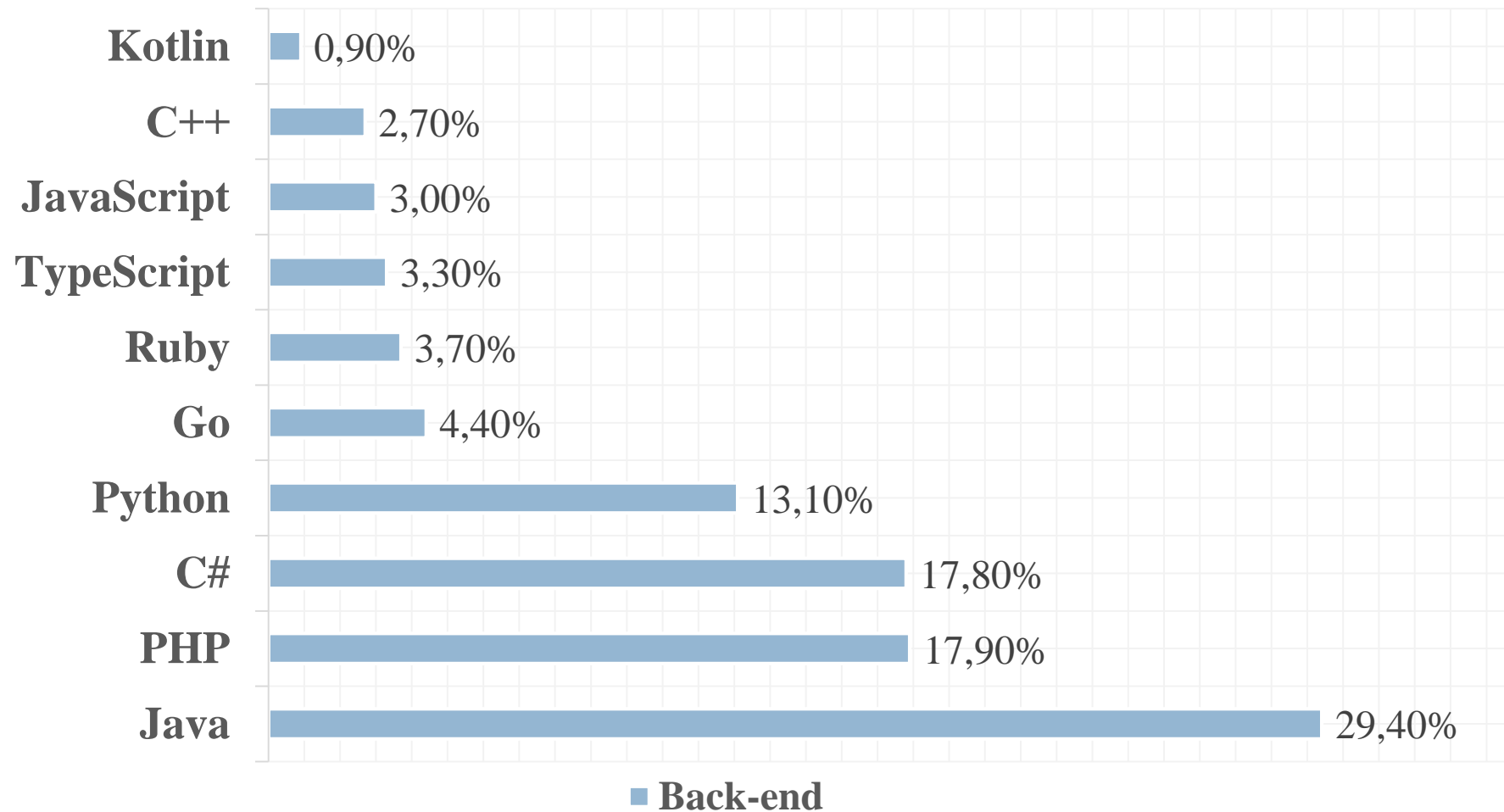


СТРУКТУРИЗАЦИЯ НАИБОЛЕЕ АКТУАЛЬНЫХ И ВОСТРЕБОВАННЫХ ЯЗЫКОВ ВЕБ-ПРОГРАММИРОВАНИЯ В СОВРЕМЕННЫХ УСЛОВИЯХ

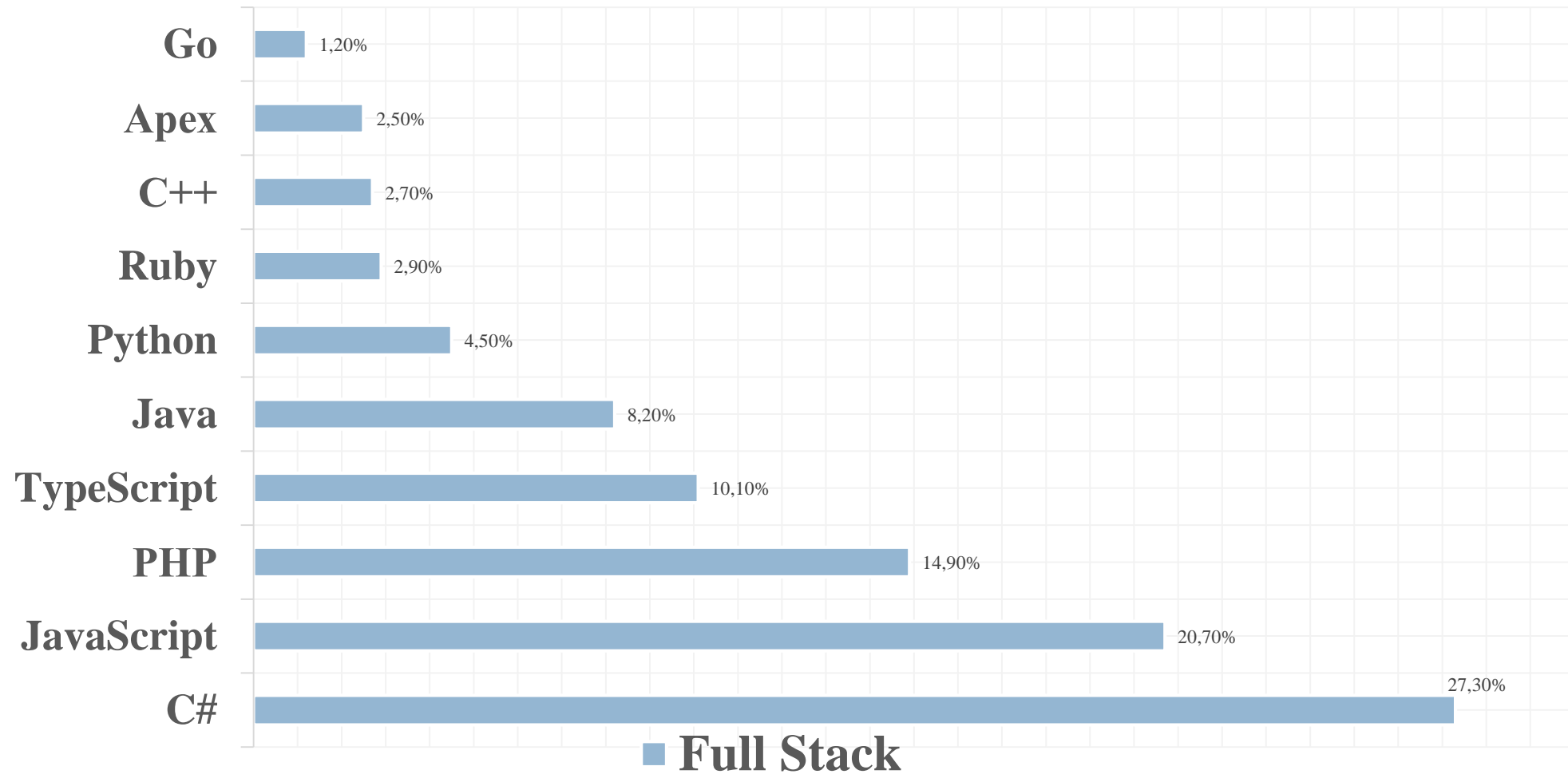


КОМПЛЕКСНА ОЦЕНКА КОММЕРЧЕСКОГО
ИСПОЛЬЗОВАНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В %.

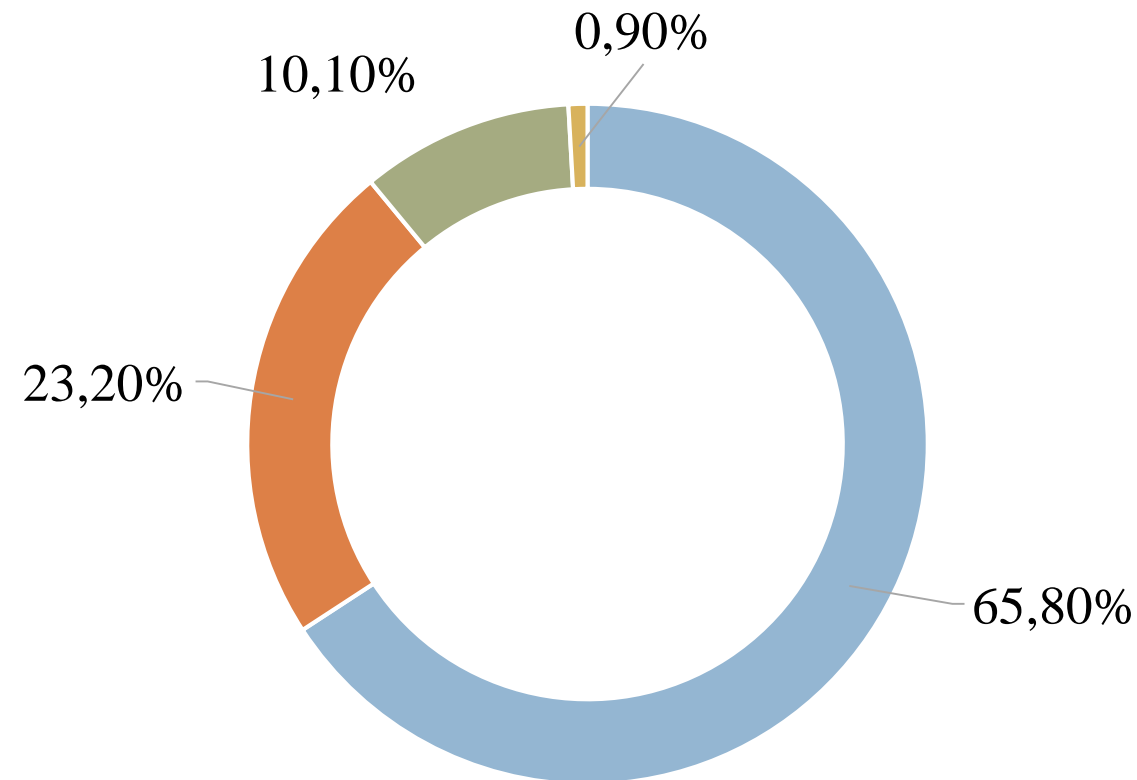
ИСПОЛЬЗОВАНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ЗА ОБЛАСТЬЮ ПРИМЕНЕНИЯ ВЕБ-РАЗРАБОТКИ



ИСПОЛЬЗОВАНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ЗА ОБЛАСТЬЮ ПРИМЕНЕНИЯ ВЕБ-РАЗРАБОТКИ



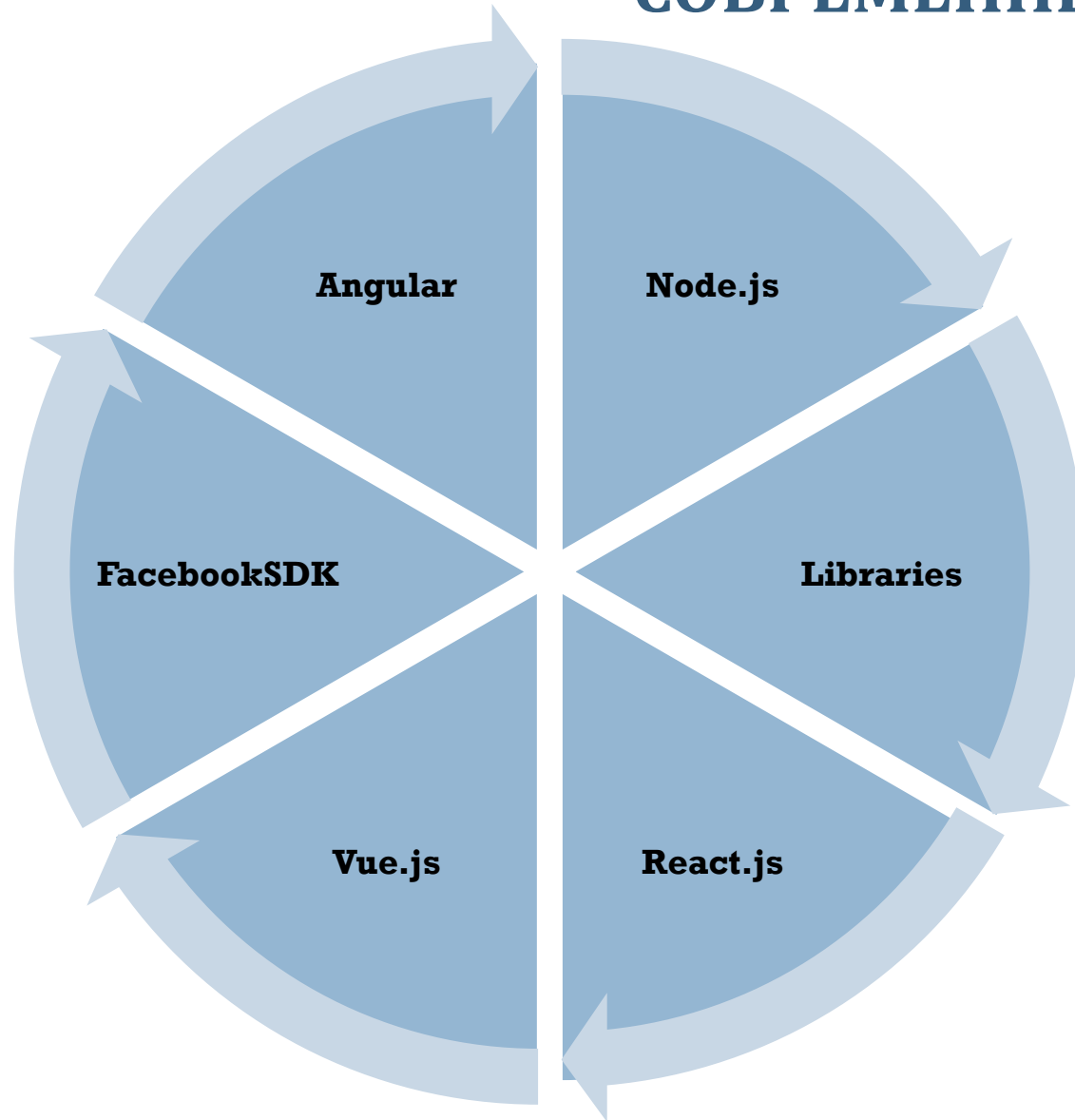
ИСПОЛЬЗОВАНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ЗА ОБЛАСТЬЮ ПРИМЕНЕНИЯ ВЕБ-РАЗРАБОТКИ



Front-end

■ JavaScript ■ TypeScript ■ HTML/CSS ■ C#

ОСНОВНЫЕ ВИДЫ FRAMEWORK И БИБЛИОТЕК, КОТОРЫЕ УПРОЩАЮТ ПРОЦЕСС ВЕБ-РАЗРАБОТКИ В СОВРЕМЕННЫХ УСЛОВИЯХ



AL-Ababneh 6. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ HTML

- Структуризация основных элементы HTML, которые используются при разработке современных веб-сайтов



| starting tag | end marking | Характеристика |
|-------------------------------------|-----------------------|--|
| <!DOCTYPE html> | | Это объявление типа страницы, сообщаемое веб-браузеру, что эта страница HTML5. |
| <"meta charset="utf-8> | | Эта строка определяет кодировку кода, который вы использовали для программирования страницы. UTF-8 - самая распространенная международная кодировка в Интернете. |
| <HTML> | </HTML> | Это корневой элемент любой HTML-страницы, содержащий весь остальной код страницы. |
| <HEAD> | </HEAD> | Это тег для информации о странице и содержит URL-адрес веб-сайта. |
| <TITLE> | </TITLE> | Используется для указания заголовка веб-страницы |
| <BODY> | </BODY> | Это тег, который относится ко всем визуальным элементам веб-сайта. |

AL-Ababneh ФОРМАТИРОВАНИЕ ТЕКСТА И ЦВЕТА

Bold font

** ... **
** ... **

Bold Text

** Bold Text **

Strong Text

** Strong Text **

Italic font

<I> ... </I>
** ... **

Italic Text

<I> Italic Text </I>

Emphasized Text

** Emphasized Text **

Font underlined

<U> ... </U>

Undelined Text

<U> Undelined Text </U>

Font with uppercase letters

^{...}

Superscript Text

^{Superscript Text}

Font with small letters

_{...}

Subscript Text






_{Subscript Text}






By adding the BGCOLOR property to the <BODY> t

```
<BODY BGCOLOR="#FFFFFF"
BACKGROUND="image.jpg">...</BODY>
```

Specify the background color

```
<BODY BGCOLOR="FFFFFF"> ... </BODY>
```

| HEX | Color |
|---------|---|
| FF0000# |  |
| FFFF00# |  |
| #00FF00 |  |
| #00FFFF |  |
| #0000FF |  |

| HEX | Color |
|---------|---|
| FF0000# |  |
| FFFF00# |  |
| #00FF00 |  |
| #00FFFF |  |
| #0000FF |  |

Determine the type of font we want

```
<FONT FACE=" Times New Roman">... Text ...</FONT>
```

Determine the font color

```
<FONT <FONT COLOR="#FF0000">... Text ...</FONT>
```

Determine font size

```
<FONT SIZE="4">... Text ...</FONT>
```

.Determine the font type and its characteristics for the entire page

```
<BASEFONT Name="Arial" COLOR="#FF0000" SIZE="5">
```

ПАРАГРАФЫ

```
<html>
<body>
  <p>This is a paragraph
  This is another paragraph.</p>
</body>
</html>
```

This is a paragraph This is another paragraph.

```
<html>
<body>
  <p>This <br> is a paragraph
  This is another paragraph.</p>
</body>
</html>
```

This is a paragraph This is another paragraph.

```
<html>
<body>
  <h2>HTML Unordered List</h2>
  <ul>
    <li>Chocolate Cake</li>
    <li>Black Forest Cake</li>
    <li>Pineapple Cake</li>
  </ul>
  <hr>
  <h2>HTML Nested Unordered List</h2>
  <ul>
    <li>Chocolate Cake
    <ul>
      <li>Chocolate Velvet Cake</li>
      <li>Chocolate Lava Cake</li>
    </ul>
    </li>
    <li>Black Forest Cake</li>
    <li>Pineapple Cake</li>
  </ul>
</body>
```

HTML Unordered List

- Chocolate Cake
- Black Forest Cake
- Pineapple Cake

HTML Nested Unordered List

- Chocolate Cake
 - Chocolate Velvet Cake
 - Chocolate Lava Cake
- Black Forest Cake
- Pineapple Cake

ЗАГОЛОВКИ

```
<html>
<body>
  <h1>Heading level 1</h1>
  <h2>Heading level 2</h2>
  <h3>Heading level 3</h3>
  <h4>Heading level 4</h4>
  <h5>Heading level 5</h5>
  <h6>Heading level 6</h6>
</body>
</html>
```

Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

МУЛЬТИМЕДИА:

```


```

```

```

Рассмотрим пример вставки картинок на веб-страницу:

```
<html>
<body>
  
    
    
</body>
</html>
```



Для размещения видео необходимо обращать внимание на разрешения.

```
<html>
<body>
  
    
    
</body>
</html>
```



Important tag properties <video >

| | |
|---|--|
| width | To specify the width of the video window, the number we pass to it is the width in pixels. |
| height | To specify the length of the video window, the number we pass to it is the length in pixels. |
| control | Shows control buttons in the video window. |
| For the <source> tag, these are its most important properties: | |
| src | Select video path to display. |
| type | Select video type to display. |

ССЫЛКИ

```
<a href="Document Link">Link Text</a>
```

| | |
|----------------|--|
| Unvisited Link | Links that we haven't visited yet are shown in blue and underlined |
| Visited Link | Links that we have previously visited appear in purple and are underlined. |
| Active Link | The active link is displayed in red and underlined. |

AL-Ababneh ВАЖНЫЕ СВОЙСТВА ТЕГА <VIDEO >

| | |
|-------------------------------------|---|
| <table> </table> | This is the main tag that can be placed to tell the browser to display the data in the table. |
| <tr> </tr> | Used to add a row to a table. |
| <th> </th> | Used to add a field to a line representing a title, i.e. the text you paste is shown in bold and in the middle. |
| <td> </td> | Used to add a field to a string that represents ordinary information, i.e. the text that is inserted into it. |
| border | Used to display the rows of a table, the number we assign to it is the size of the rows that will be placed between the rows and columns. <TABLE GRAM="5"> |
| width | It is used to specify the width of the table on its own, not for the browser. The number we give it represents the width of the table. <TABLE WIDTH="80%"> |
| height | It is used to specify the length of the table on its own, not for the browser. The number we give it represents the length of the table. <TABLE HEIGHT="500"> |
| CELLSPACING | Specifies the space between each table cell <TABLE CELLSPACING="10"> |
| CELLPADDING | Specifies the distance between the border and the beginning of the text in each cell. Or let's say: specify the size of the margins for the cells of the table. <TABLE CELLPADDING="10"> |
| ALIGN | Specifies the left or right alignment of the table horizontally on the page. <TABLE ALIGNMENT="Left"> <TABLE ALIGNMENT="Right"> |
| BGCOLOR | Used to specify the background color of the table. <TABLE BGCOLOR="#00FFFF"> |

```
<html>
<body>
<form>
  <label for="user-name">Username:</label>
  <input type="text" name="username" id="user-name">
</form>
</body>
</html>
```

Username:

```
<html>
<body>
  <form action="examples/pages/Confirmed.php" method="post">
    <label>Username: <input type="text" name="username"></label>
    <label>Password: <input type="password"
name="userpass"></label>
    <input type="submit" value="Submit">
  </form> </body></html>
```

Username:

Password:

```
<html>
<body>
  <form>
    <input type="checkbox" name="sports" value="soccer"
id="soccer">
    <label for="soccer">Soccer</label>
    <input type="checkbox" name="sports" value="cricket"
id="cricket">
    <label for="cricket">Cricket</label>
    <input type="checkbox" name="sports" value="baseball"
id="baseball">
    <label for="baseball">Baseball</label>
  </form> </body></html>
```

Soccer Cricket Baseball

```
<html>
<body>
  <form>
    <input type="radio" name="gender" value="male" id="male">
    <label for="male">Male</label>
    <input type="radio" name="gender" value="female"
id="female">
    <label for="female">Female</label>
  </form>
</body>
</html>
```

Male Female

Фрейм:

<FRAMESET> ... <FRAMESET>
Used instead of <BODY> tags ... </BODY>

So the frame master file does not include a <body> tag

ОСОБЕННОСТИ УПРАВЛЕНИЯ СВОЙСТВАМИ HTML- ФРЕЙМОВ

| | |
|--|---|
| Позволяет определить два вертикальных окна, каждое из которых занимает 50% размера экрана. | <code><FRAMESET COLS="50%,50%"> </FRAMESET></code> |
| Позволяет показывать три окна, размеры которых составляют 20%, 50% и 30% соответственно от размера экрана. | <code><FRAMESET COLS="20%,50%,30%"> </FRAMESET></code> |
| Позволяет определить три вертикальных фрейма, первый 200 px и второй 300px. Что касается третьего *, то есть он не ограничивается конкретным размером, а будет оставшимся размером экрана (пока мы не знаем разрешение экрана, которое использует посетитель сайта) | <code><FRAMESET COLS="200,300,*"> </FRAMESET></code> |
| Определяет четыре кадра, размер первого - 200 пикселей, третьего - 15% размера экрана, четвертого - 20% размера экрана, четвертого - размер остального экрана. | <code><FRAMESET COLS="200*,15%,20%"> </FRAMESET></code> |
| Выделяет три кадра, первый имеет размер 150 пикселей. Оставшееся пространство делится на основании того, что третий кадр в два раза больше (*2), чем размер второго кадра (*) | <code><FRAMESET COLS="150*,2*"> </FRAMESET></code> |

ОСОБЕННОСТИ УПРАВЛЕНИЯ СВОЙСТВАМИ ROWS HTML- ФРЕЙМОВ

| | |
|---|--|
| Позволяет определить две горизонтальные рамки, каждая по 50% высоты экрана. | <code><FRAMESET ROWS="50%,50%"> </FRAMESET></code> |
| Позволяет определить три горизонтальные рамки, высота которых составляет 20 %, 50 % и 30 % соответственно от высоты экрана. | <code><FRAMESET ROWS="20%,50%,30%"> </FRAMESET></code> |
| Позволяет определить три горизонтальных кадра, первый будет иметь высоту 50 пикселей, второй -120 пикселей, а третий - оставшуюся высоту экрана. | <code><FRAMESET ROWS="50,120,*"> </FRAMESET></code> |
| Позволяет определить четыре горизонтальных кадра, высота первого составляет 50 пикселей, третьего - 15% высоты экрана, четвертого - 20% высоты экрана, а высота второго будет остатком высоты экрана. | <code><FRAMESET ROWS="50*,15%,20%"> </FRAMESET></code> |
| Позволяет задать два кадра, высота второго в два раза больше высоты первого | <code><FRAMESET COLS="*,2*"> </FRAMESET></code> |

7. ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVASCRIPT

- 1. Добавьте код JavaScript внутрь веб-страницы. Мы можем добавить код JavaScript непосредственно в веб-страницы, поместив его как содержимое внутри элемента `<script>`. Элемент `<script>` указывает, что веб-браузер содержит исполняемый скрипт, а не HTML-код.

```
<html>
<head>
<meta charset="utf-8">
<title> JavaScript</title>
</head>
<body>
  <script>
var greet = "Hello World!";
document.write(greet);
  </script>
</body>
</html>
```

Hello World!

- 2. Вызов внешнего файла JavaScript необходимо поместить сам код JavaScript в отдельный файл.js, а затем вызвать этот файл на веб-странице через атрибут `src` тега `<script>`

```
<html>
<head>
<meta charset="utf-8">
<title> JavaScript </title>
</head>
<body>
<button type="button" id="myBtn">Click
Me<button>
<script src="js/hello.js"></script>
</body> </html>
```

Click Me

- 3. Размещайте код JavaScript непосредственно внутри тегов

onload**onkeypress****onmouseover****onclick**

Важно не размещать слишком много встроенного кода JavaScript, так как это создаст беспорядок между HTML и JavaScript и затруднит поддержку вашего кода:

```
<html>
<head>
<meta charset="UTF-8">
<title>Inlining JavaScript</title>
</head>
<body>
<button onclick="alert ('Hello
World!)">Click Me</button>
</body> </html>
```

Click Me

Понимание синтаксиса JavaScript

Это набор правил, определяющих построение правильно структурированной программы JavaScript. Язык JavaScript состоит из операторов, которые размещаются внутри элемента `<script>` на веб-странице или во внешнем файле JavaScript с расширением `.js`.

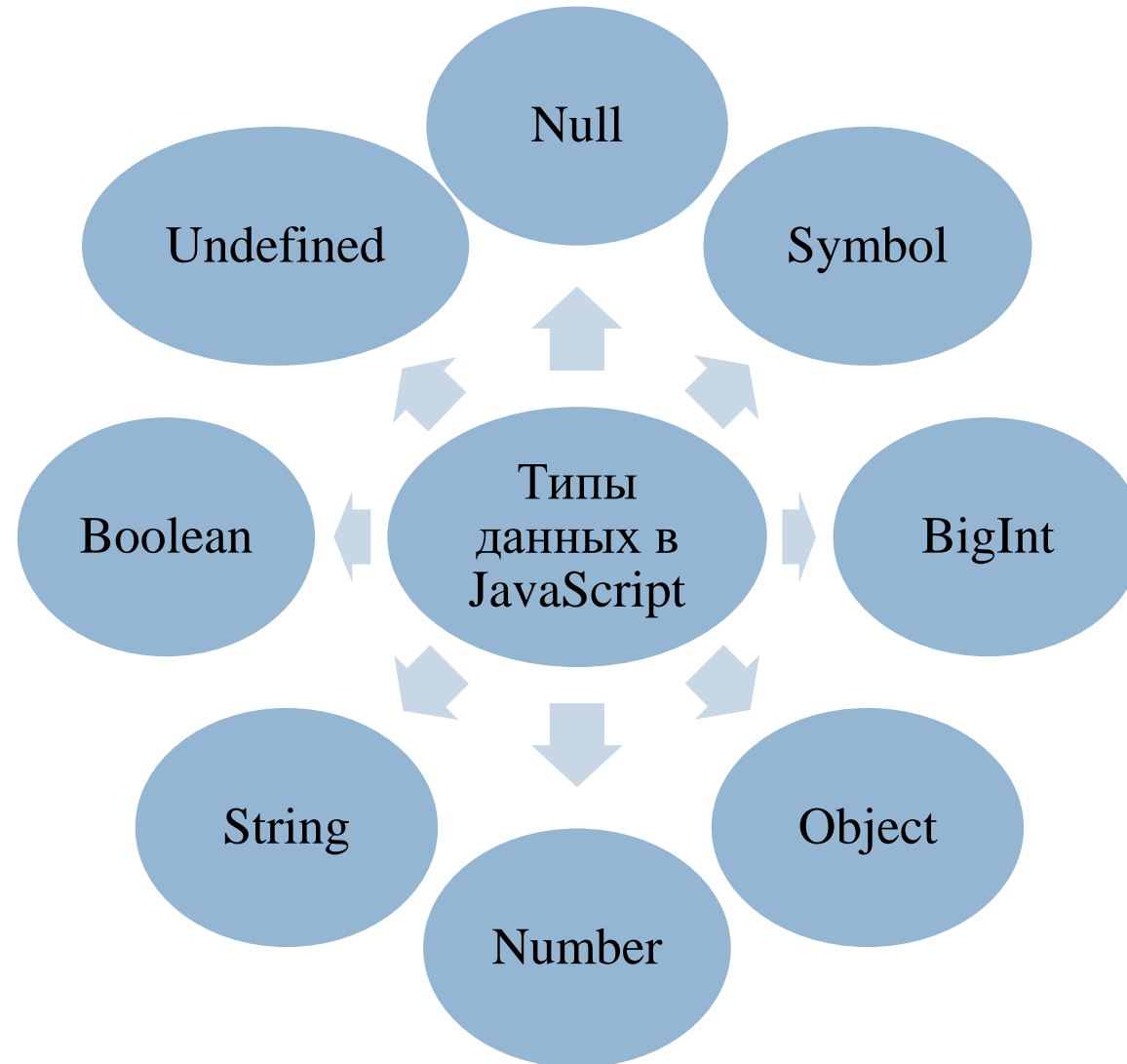
```
var x = 5;
```

```
var y = 10;
```

```
var sum = x + y;
```

```
document.write(sum); print the value of the variable //
```

СТРУКТУРИЗАЦИЯ ОСНОВНЫХ ТИПОВ ДАННЫХ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В JAVASCRIPT В СОВРЕМЕННЫХ УСЛОВИЯХ



- **String** представляет собой тип данных, который используется для представления текстовых данных (т. е. последовательности символов). Строки создаются с использованием одинарных или двойных кавычек, заключающих один или несколько символов

```
<html>
<head>
<meta charset="utf-8">
<title> String </title>
</head>
<body>
<script>
var a = 'Hi there!';
var b = "Hi there!";
document.write(a + "<br>");
document.write(b);
</script>
</body>
</html>
```

```
Hi there!
Hi there!
```

- **Number data** представляет собой данные числового типа, которые используются для представления положительных или отрицательных чисел с десятичным разрядом или без него или чисел, которые записаны с использованием экспоненциальной записи, такой как $1,5e-4$ (эквивалентно $1,5 \times 10^{-4}$).

```
<html>
<head>
<meta charset="utf-8">
<title> Number</title>
</head>
<body>
<script>
var a = 25;
var b = 80.5;
var c = 4.25e+6;
var d = 4.25e-6;
document.write(a + "<br>");
document.write(b + "<br>");
document.write(c + "<br>");
document.write(d);
</script>
</body>
</html>
```

```
25
80.5
4250000
0.00000425
```

- **Undefined:** Неопределенный тип данных может иметь только одно значение, специальное значение `undefined`. Если переменная объявлена, но значение не указано, ее значение по умолчанию равно `undefined`

```
<html>
<head>
  <meta charset="utf-8">
  <title> Undefined</title>
</head>
<body>
  <script>
    var a;
    var b = "Hello World!"
    document.write(a + "<br>");
    document.write(b);
  </script>
</body>
</html>
```

```
undefined
Hello World!
```

- **Null data** является специальным типом данных, который может иметь только одно значение `null`, `null` означает, что значения нет, но пустая строка (`""`) или значение `0` не является `null`, `null` - это просто ничто. Переменную можно очистить от текущего содержимого, установив для нее значение `null`

```
<html>
<head>
  <meta charset="utf-8">
  <title> Null</title>
</head>
<body>
  <script>
    var a = null;
    document.write(a + "<br>"); //
null
    var b = "Hello World!"
    document.write(b + "<br>"); //
Hello World!
    b = null;
    document.write(b) // null
  </script>
</body>
</html>
```

```

null
Hello World!
null
```

СТРУКТУРИЗАЦИЯ ОСНОВНЫЕ МАТЕМАТИЧЕСКИХ
ОПЕРАТОРОВ В JAVASCRIPT

| Результат | Пример | Оператор | Описание |
|---|----------------------------|---|----------|
| Сумма x и y | $x + y$ | Addition | + |
| Вычесть y из x | $x - y$ | Subtraction | - |
| Произведение x на y | $x * y$ | Multiplication | * |
| Разделить x на y | x / y | Division | / |
| Остаток от деления x на y | $x \% y$ | Modulus | % |
| $X = y$ | $x = y$ | Assign | = |
| $X = x + y$ | $x += y$ | Add and assign | += |
| $X = x - y$ | $x -= y$ | Subtract and assign | -= |
| $X = x * y$ | $x *= y$ | Multiply and assign | *= |
| $X = x / y$ | $x /= y$ | Divide and assign quotient | /= |
| $X = x \% y$ | $x \% = y$ | Divide and assign modulus | %= |
| Увеличивает x на единицу, затем возвращает значение x | авансовое увеличение | Pre-increment | ++x |
| Возвращает значение x, затем увеличивает x на единицу | последующее увеличение | Post-increment | x++ |
| Вычитает единицу из значения x, затем возвращает значение x | Предварительное уменьшение | Pre-decrement | --x |
| Возвращает значение x, затем вычитает единицу из значения x | Последующее снижение | Post-decrement | x-- |
| Истина, если оба x и y верны | $x \&\& y$ | And | && |
| Истина, если либо x, либо y верны | $x \ \ y$ | Or | \ \ |
| Истина, если x ложно | $!x$ | Not | ! |
| Верно, если x равно y | $x == y$ | Equal | == |
| Истина, если x равно y, и две переменные имеют один и тот же тип данных | $x === y$ | Identical | === |
| Истина, если x не равно y | $x != y$ | Not equal | != |
| True, если x не равно y или это два разных типа данных | $x !== y$ | Not identical | !== |
| Верно, если x меньше y | $x < y$ | Less than | < |
| Истина, если x больше, чем y | $x > y$ | больше Greater than | > |
| Истина, если x меньше или равно y | $x <= y$ | меньше или равно Less than or equal to | <= |
| Истина, если x больше или равен y | $x >= y$ | больше или равно Greater than or equal to | >= |

КЛАССИФИКАЦИЯ СОБЫТИЙ НА ЧЕТЫРЕ ОСНОВНЫЕ ГРУППЫ:

1. Mouse Events событие мыши запускается, когда пользователь щелкает элемент или наводит указатель мыши на элемент. Вот некоторые из наиболее важных событий мыши и их обработчики событий.

2. Click (onclick) событие происходит, когда пользователь нажимает на элемент на веб-странице, обычно это элементы формы или ссылки. Возможно обработать событие Click с помощью обработчика события onclick. В примере будет показано предупреждающее сообщение при нажатии на элементы:

```
<html>
<head>
  <meta charset="utf-8">
  <title>onclick</title>
</head>
<body>
  <button type="button" onclick="alert('You
have clicked a button!');"> Press here
</button>
  <a href="#" onclick="alert('You have clicked
a link!');"> Press here </a>
</body>
</html>
```

Click here Click here

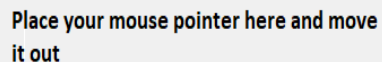
3. . *Mouseover (onmouseover)* это событие возникает, когда пользователь наводит указатель мыши на элемент, при этом событие Mouseover обрабатывается с помощью обработчика событий onmouseover. Этот пример покажет нам предупреждающее сообщение, когда вы наведете указатель мыши на элементы:

```
<html>
<head>
  <meta charset="utf-8">
  <title>onmouseover</title>
</head>
<body>
  <button type="button" onmouseover="alert('You
have placed mouse pointer over a button!');"> ضع مؤشر
هنا الفأرة</button>
  <a href="#" onmouseover="alert('You have right-
clicked a link!');"> ضع مؤشر الفأرة هنا</a>
</body>
</html>
```

Place your mouse cursor here

4. *The mouseout (onmouseout)* когда пользователь перемещает указатель мыши за границы элемента, мы можем обработать событие mouseout с помощью обработчика события onmouseout. Пример покажет нам предупреждающее сообщение, когда произойдет событие выхода курсора мыши::

```
<html>
<head>
  <meta charset="utf-8">
  <title>onmouseover</title>
  <style>
    div {
      min-height : 200px;
    }
    button {
      border : 5px solid red;
      font-size : 30px;
      padding : 30px;
    }
    a {
      border : 5px solid blue;
      font-size : 30px;
      padding : 25px;
    }
  </style>
</head>
<body>
  <div>
    <button type="button" onmouseout="alert('You have
moved out of the button!')"> Place your mouse here and move
it out </button>
  </div>
  <div>
    <a href="#" onmouseout="alert('You have moved out of
the link!')"> Place your mouse here and move it out </a>
  </div>
</body>
</html>
```




5. *Strings*: это событие клавиатуры запускается, когда пользователь нажимает или отпускает клавишу на клавиатуре. Вот некоторые из наиболее важных событий клавиатуры и их обработчики событий.

```
<html>
<head>
  <meta charset="utf-8">
  <title>تخطي علامات الاقتباس</title>
</head>
<body>
  <script>
    var str1 = 'it\'s okay';
    var str2 = "He said \"Goodbye\"";
    var str3 = 'She replied \'Calm down,
please\'';
    document.write(str1 + "<br>");
    document.write(str2 + "<br>");
    document.write(str3);
  </script>
</body>
</html>
```

it's okay
He said "Goodbye"
She replied 'Calm down, please

| | |
|---|----|
| Заменяется кодом новой строки .Newline. | \n |
| Заменяет символ tab. | \t |
| Его заменяет возврат каретки, который перемещает курсор в начало строки, не переходя на следующую строку. | \r |
| Заменяется символом возврата backspace | \b |
| Заменяются одиночной обратной косой чертой(\) | \\ |

Управляющие последовательности на практике в JavaScript работают по следующему принципу:

```
<html>
<head>
  <meta charset="utf-8">
  <title>Escape Sequences</title>
</head>
<body>
  <script>
    var str1 = "The quick brown fox \n jumps over the lazy dog.";
    document.write("<pre>" + str1 + "</pre>");
    var str2 = "C:\Users\Downloads";
    document.write(str2 + "<br>");
    var str3 = "C:\\Users\\Downloads";
    document.write(str3);
  </script>
</body>
</html>
```

```
The quick brown fox
jumps over the lazy dog.
C:UsersDownloads
C:\Users\Downloads
```

8. ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP

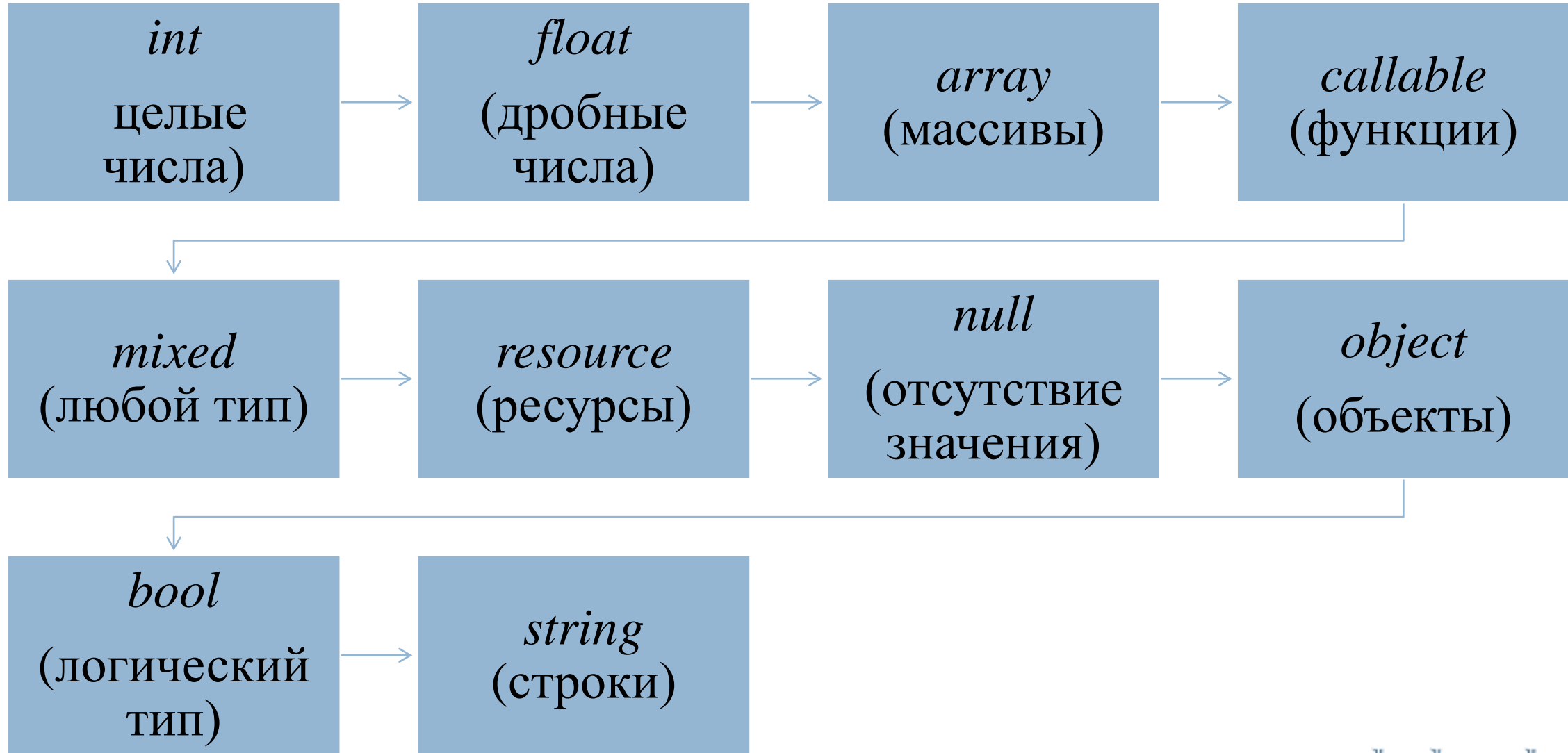
Каждая буква этой аббревиатуры относится к одной из этих программ:

1. Буква W обозначает Windows.
2. Буква A обозначает Apache.
3. Буква M обозначает MySQL.
4. Буква P обозначает PHP.

Важность установки локального сервера Localhost на устройстве:

1. Изучите программирование веб-сайтов со стороны сервера.
2. Изучите базы данных MySQL.
3. Изучите среды программирования, например, Codeigniter.
4. Попробуйте любой скрипт, требующий хостинга.
5. Попробуйте некоторые настройки на сайте, прежде чем работать над исходным веб-сайтом.
6. Экспериментируйте или проектируйте шаблоны и плагины WordPress.
7. Написание профессионального контента.

КЛАССИФИКАЦИЯ БАЗОВЫХ ТИПОВ, ДАННЫХ РНР, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В СОВРЕМЕННЫХ УСЛОВИЯХ



ТИПИЗАЦИЯ ТИПОВ, ДАННЫХ РНР В СОВРЕМЕННЫХ УСЛОВИЯХ

*Скалярные
типы:*

- boolean
- integer
- float
- string

*Смешанные
типы:*

- array
- object

*Специальные
типы:*

- resource
- NULL

Псевдотипы:

- mixed
- number
- callback

It starts with the tag

```
<?php
```

It ends with the tag

```
?>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta charset="UTF-8">
</head>
<body>
<h1> My first php page </h1>
<?php
  echo " Hello World ";
?>
</body>
</html>
```

My first PHP page
Hello World!

▪ ***Comments***

- Заметка в одну строку

single line note //

single line note#

- Примечание с более чем одной строкой

Note with more than one line /*

It contains more information

It will be ignored by the server

```
*/
```

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment

/* This is a
multi-line comment */
?>

</body>
</html>
```

▪ Правила именованных переменных PHP:

- Переменная начинается со знака доллара \$, за которым следует имя переменной;
- Переменная должна начинаться с буквы или подчеркивания (_);
- Имя переменной не может начинаться с цифры. Поместите _ перед числом, чтобы оно было принято;
- Имя должно состоять из букв алфавита и цифр или знака (_) Имена переменных чувствительны к орфографии, например, \$y не то же самое, что \$Y. Каждая переменная имеет свое собственное значение

```
<?php
$txt=" Hello world!";
$x=8;
$y=14;
?>
```

Основные типы функций PHP

Это функции, лежащие в основе языка, и это то, что PHP предоставляет для непосредственного использования.

Built-in Functions

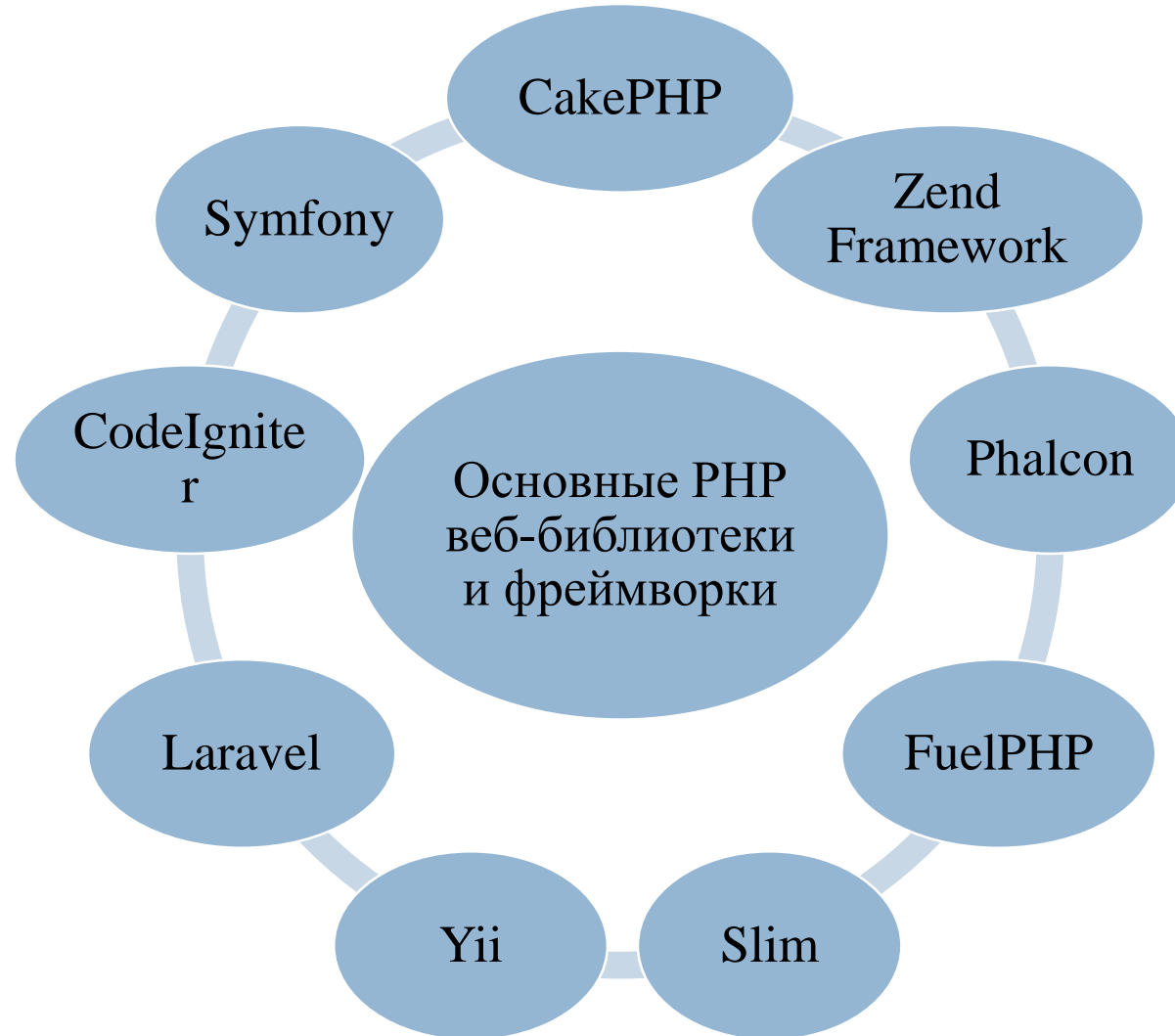
Функции, которые создаются и используем в программных проектах

User Defined Functions

Дополнительные библиотечные функции, такие как библиотека изображений GD, должны быть установлены и включены, чтобы иметь возможность их использовать.

| Название функции PHP | Описание и характеристика функции PHP по работе с переменными |
|--------------------------|--|
| boolval | Возвращает логическое значение переменной |
| debug_zval_dump | Выводит строковое представление внутренней структуры zval |
| floatval | Возвращает значение переменной в виде числа с плавающей точкой |
| get_debug_type | Возвращает имя типа переменной в виде, подходящем для отладки |
| get_defined_vars | Возвращает массив всех определённых переменных |
| get_resource_id | Возвращает целочисленный идентификатор для данного ресурса |
| get_resource_type | Возвращает тип ресурса |
| gettype | Возвращает тип переменной |
| intval | Возвращает целое значение переменной |
| is_array | Определяет, является ли переменная массивом |
| is_bool | Проверяет, является ли переменная булевой |
| is_callable | Проверяет, что значение может быть вызвано как функция в текущей области видимости |
| is_countable | Проверить, что содержимое переменной является счётным значением |
| is_numeric | Проверяет, является ли переменная числом или строкой, содержащей число |
| is_scalar | Проверяет, является ли переменная скалярным значением |
| unserialize | Создаёт PHP-значение из хранимого представления |
| serialize | Генерирует пригодное для хранения представление переменной |
| var_dump | Выводит информацию о переменной |
| var_export | Выводит или возвращает интерпретируемое строковое представление переменной |

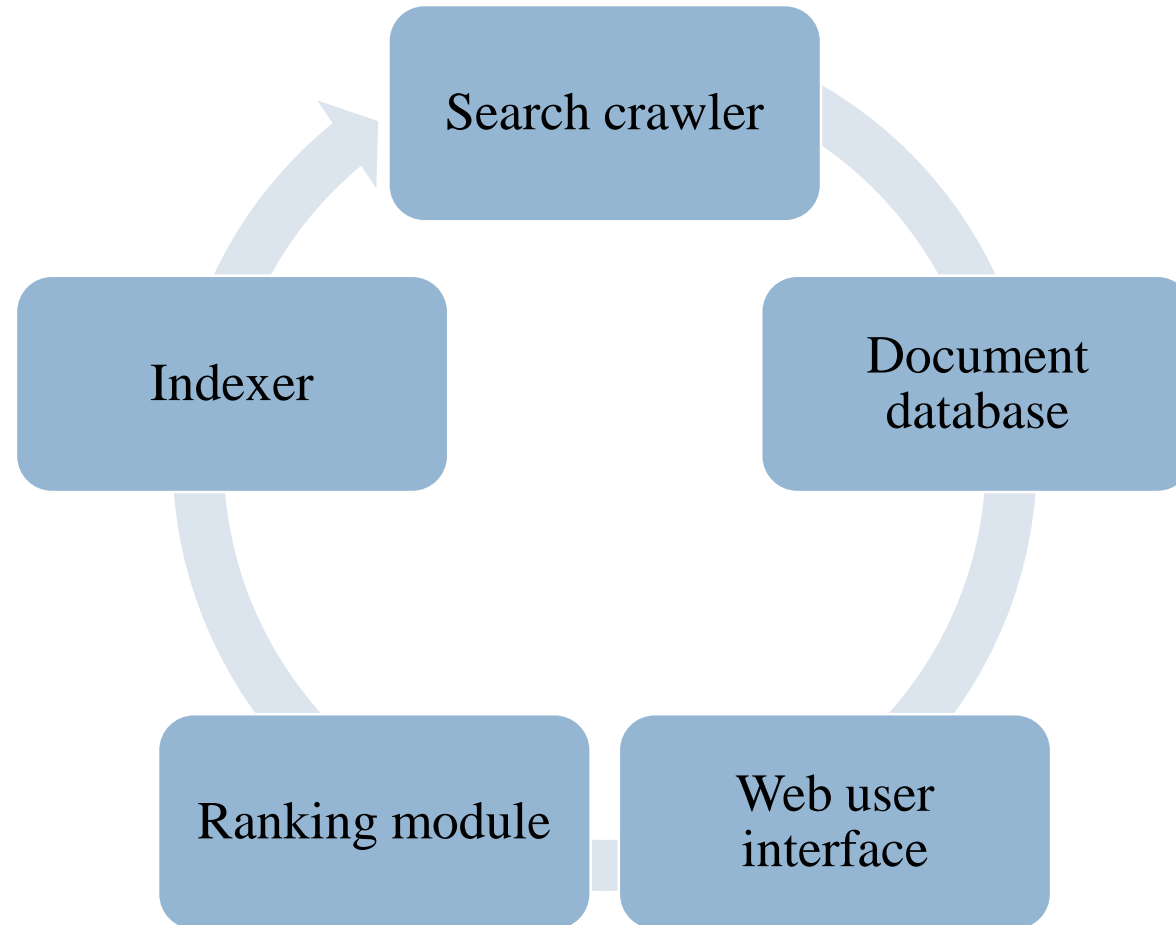
СТРУКТУРИЗАЦИЯ ОСНОВНЫХ PHP ВЕБ-БИБЛИОТЕК И ФРЕЙМВОРКОВ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В СОВРЕМЕННЫХ УСЛОВИЯХ



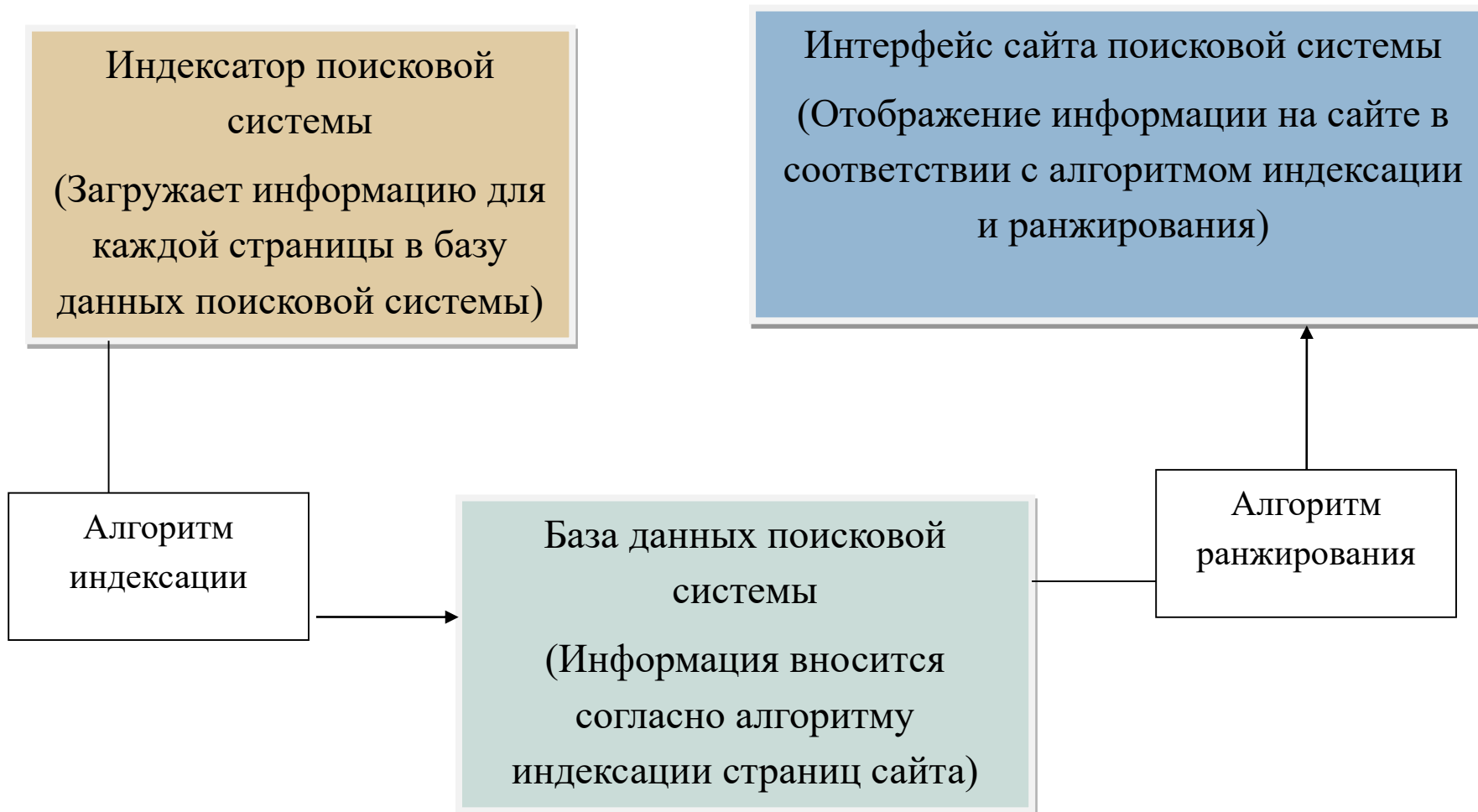
| PHP Веб-библиотека/фреймворка и их характеристика и особенности | |
|---|---|
| Laravel | Бесплатный опенсорсный PHP-фреймворк наиболее часто используемый веб-разработчики. Одна из главных причин этого заключается в том, что Laravel позволяет разрабатывать сложные веб-приложения проще, быстрее и безопаснее, чем любые другие фреймворки. |
| Yii | Простой, но высокопроизводительный универсальный фреймворк, основанный на компонентной структуре. Он известен благодаря высокой производительности, но, в первую очередь, он знаменит своей простотой. Он позволяет использовать код сторонних разработчиков, а имеющийся в нём генератор кода Gii позволяет быстро создавать базовые структуры, на основе которых можно строить собственные решения. |
| CodeIgniter | Фреймворк, который известен тем, что для его приведения в рабочее состояние требуется минимальное количество настроек. Размер фреймворка, включая документацию, не превышает 2 Мб, но он, несмотря на такую компактность, предлагает разработчикам множество стандартных модулей для создания компонентов, отличающихся надёжностью и подходящих для многократного использования. Это делает CodeIgniter отличным выбором для тех, кто создаёт динамические веб-сайты. |
| Symfony | Включает в себя встроенную систему тестирования и основан на обширном MVC-фреймворке. Это делает его идеальным выбором для крупномасштабных веб-проектов, создаваемых на уровне организаций. Также поддерживает работу с различными базами данных. |
| CakePHP | Встроенная подсистема, называемая «Components and Helpers», облегчает работу веб-разработчиков, избавляя от необходимости делать многое самостоятельно, а выбирать из набора библиотек, реализующих много полезных возможностей. Подходит для реализации веб-проектов, в которых реализован какой-то редкий или нестандартный функционал. |
| Zend Framework | Позволяет загружать необходимые компоненты в виде отдельных библиотек. При этом можно, без особых сложностей, интегрировать в проекты и внешние библиотеки. При его использовании крайне просто организовать многократное использование кода. |
| Phalcon | Высокопроизводительный фреймворк, размещаемый в памяти и охватывающий все уровни веб-разработки. Помимо скорости, качество организована система управления ресурсами, и универсальная автоматизированная система загрузки ресурсов. |
| FuelPHP | Позволяет создавать веб-проекты разных масштабов. Он отличается надёжной и хорошо продуманной системой безопасности с поддержкой Output Encoding, с защитой от CSRF- и XSS-атак. В нём имеется уникальная утилита командной строки, но этим его полезные возможности не ограничиваются. Среди них - весьма продвинутая встроенная ORM. Среди других его возможностей-поддержка разработки RESTful-API, хорошая система маршрутизации, встроенные механизмы защиты от уязвимостей. |
| Slim | Это один из лучших фреймворков для начинающих. Его очень легко освоить, он отличается дружелюбной и простой документацией. На него стоит взглянуть тем, кого интересует разработка RESTful-API. Он облегчает решение таких задач, как маршрутизация, шифрование куки-файлов, HTTP-кеширование на стороне клиента. |

9. ПРОДВИЖЕНИЕ И ПОИСКОВАЯ ОПТИМИЗАЦИЯ ВЕБ-САЙТА

АРГУМЕНТАЦИЯ СТРУКТУРЫ ПОИСКОВОЙ СИСТЕМЫ



СХЕМАТИЧЕСКОЕ ОТРАЖЕНИЕ РАБОТЫ ПОИСКОВЫХ СИСТЕМ В МИРЕ



СТРУКТУРИРОВАНИЕ ТИПОВ ПОИСКОВЫХ СИСТЕМ В ЗАВИСИМОСТИ ОТ МЕХАНИЗМА РАБОТЫ

1. Search engines using the scanner-crawler mechanism

2. Search engines operating using the mechanism of directories that are controlled by a person

3. Search engines working using a hybrid mechanism

4. Search engines working with other search engines

КЛАССИФИКАЦИЯ ТИПОВ ПОИСКОВЫХ СИСТЕМ В СОВРЕМЕННОМ МИРЕ

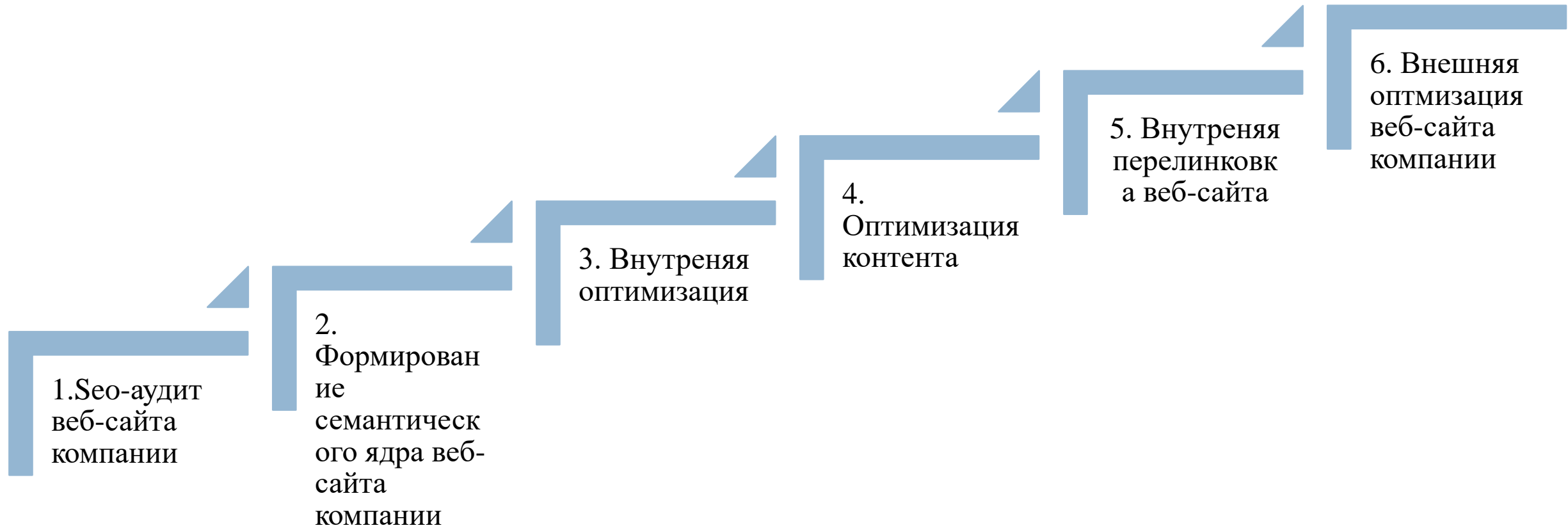
Global
Search
Engines

- Universal
- Specialized
- Thematic

Local search
engines

- local
- search by individual resource

КЛЮЧЕВЫЕ ЭТАПЫ SEO-ОПТИМИЗАЦИИ ВЕБ-САЙТА КОМПАНИЙ В СОВРЕМЕННЫХ УСЛОВИЯХ



ЭЛЕМЕНТЫ ВНУТРЕННЕЙ ПОИСКОВОЙ ОПТИМИЗАЦИИ САЙТА ОРГАНИЗАЦИИ



СТРУКТУРИЗАЦИЯ КЛЮЧЕВЫХ ЭЛЕМЕНТОВ КОНТЕНТА, КОТОРЫЕ ВЛИЯЮТ НА ОПТИМИЗАЦИЮ ВЕБ-САЙТА И ЕГО РАНЖИРОВАНИЕ



10. ТРЕНДЫ ВЕБ-ДИЗАЙНА И РАЗРАБОТКИ ВЕБ-САЙТОВ

ОСНОВНЫЕ ОСОБЕННОСТИ СОВРЕМЕННОГО ВЕБ-ДИЗАЙНА В СЕТИ ИНТЕРНЕТ

Простота в адаптации.

Плоский веб-сайт можно легко адаптировать под смартфон, планшет и ПК

Практичность.

Принцип такого веб-дизайна состоит в том, что дизайнер полностью минимизирует количество стилей, анимаций и веб-сайт загружается быстрее.

Легкость использования.

Пользователь может быстро найти нужную информацию на веб-сайте. Кроме того, воспринимается она гораздо проще и понятнее

Красота.

Плоские веб-сайты выглядят довольно привлекательно, стильно и действительно, они цепляются.

- 1. Одностраничные веб-сайты: это сайты, которые используют одну страницу в Интернете для отображения контента проекта или организации, и контент разделен на разделы и не содержит традиционных меню вкладок, которые являются разными страницами, и поэтому посетитель веб-сайта может получить доступ ко всему контенту, прокручивая длинную страницу, чтобы перемещаться между информацией, прокручивая, а не нажимая и просматривая каждую страницу по отдельности.
- 2. Графический веб-дизайн. Графический веб-дизайн постоянно развивается, и разработчики могут положиться на графический дизайн, чтобы создать уникальный визуальный опыт для пользователей. В 2023 году основное внимание будет уделяться добавлению движущихся и 3D-визуальных элементов в дизайн.
- 3. Простота. Некоторые веб-сайты просто хотят простоты, учитывая их целевую демографическую группу. Если большинство ваших пользователей — пожилые люди, вы можете сохранить его традиционным и простым, чтобы они могли легко найти нужные им продукты.
- 4. Прокручивающиеся переходы. Эффективный веб-дизайн позволяет онлайн-пользователям лучше взаимодействовать с содержимым веб-сайта. Один из способов увеличить количество онлайн-взаимодействий — передавать рефералы.
- 5. Простота в цветовых схемах: цвета играют важную роль в улучшении пользовательского опыта на веб-сайтах, поскольку выбор подходящего цвета обеспечивает простоту использования, а также ясность и легкость чтения информации, отображаемой на экране.
- 6. Используйте светлый фон веб-сайта. За последние несколько лет темный режим стал очень популярной тенденцией дизайна. Однако в 2022 году эта тенденция встречалась все реже. Светлые цвета займут свое место в дизайне веб-сайтов, например, веб-сайты с полностью белым или другим нейтральным фоном.

- 7. Использование микроанимации веб-сайта. Использование небольшой анимации является одной из важнейших тенденций в веб-дизайне, которая в последнее время широко используется на веб-сайтах. Это простые анимации, специально разработанные для того, чтобы направлять покупателя во время взаимодействия с цифровым продуктом. Они полностью интегрированы с пользовательским опытом для упрощения процесса продажи.
- 8. Сложные фильтры веб-сайта. Поскольку популярность онлайн-режима растет во всех областях мировой экономики, количество услуг и товаров, предлагаемых в Интернете, растет, и часто оно включает в себя большое количество похожих товаров и услуг.
- 9. Интерактивный поиск веб-сайта. Виджет поиска также можно добавить на боковую панель веб-сайта электронной коммерции. Часто, когда вы разрабатываете большой и сложный функциональный веб-сайт, этого недостаточно.
- 10. Функции приложений в мобильной версии веб-сайта. Хотя количество клиентов, совершающих покупки со смартфонов, уже давно превысило количество тех, кто пользуется компьютером, переход был медленным.
- 11. Быстро загружаемый веб-сайт: С обновлением алгоритма Google Core Web Vitals в 2021 году скорость загрузки мобильных страниц стала важнейшим фактором ранжирования веб-сайта на страницах результатов поиска, поскольку сверхбыстрое время загрузки является одним из важнейших критериев веб-дизайна.
- 12. Веб-программирование PWA (Progressive Web Applications), это тип прикладного программного обеспечения, разработанного с использованием популярных веб-технологий, таких как HTML и JavaScript.

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА ПРОГРЕССИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ PWA В СОВРЕМЕННЫХ УСЛОВИЯХ

Удешевление
процесса веб-
разработки,
поддержки и
обновления

Сокращение
сроков веб-
разработки и
выхода релиза

Возможность
полноценной
работы в
автономном
режиме

Увеличение
скорости
загрузки и
экономия
памяти гаджетов

Повышение
уровня
конверсии и
вовлеченности
аудитории

Технологии дополненной реальности можно использовать непосредственно на веб-сайтах в браузере, поэтому это направление получило название WebAR.

Это нововведение улучшает взаимодействие с клиентами и имеет другие важные преимущества, особенно для коммерческих ИТ-продуктов:

- Пользователи могут примерить товар перед покупкой
- Улучшается пользовательский опыт
- Онлайн-покупки становятся более осязаемыми
- Сокращение времени обслуживания клиентов

ТЕНДЕНЦИИ РАЗВИТИЯ ДИЗАЙНА ВЕБ-САЙТОВ:

- **13. Голосовой поиск:** Важно отметить, что оптимизацию голосового поиска можно просто определить как процесс оптимизации веб-страниц для отображения в голосовом поиске. Устройства, использующие распознавание голоса, быстро набирают популярность благодаря голосовым помощникам и Интернету. Технологии развивались настолько экспоненциально, что к следующему году эти устройства смогут распознавать голоса разных людей и предоставлять персонализированный опыт на основе искусственного интеллекта.
- **14. Оптимизация голоса:** Оптимизация голоса также важна для веб-сайтов, и крупные компании хотят воспользоваться этой возможностью, потому что она определит будущее Интернета.
- **15. Машинное обучение:** Инновационный подход также должен заключаться в извлечении персонализированного контента с помощью машинного обучения. Персонализация контента с помощью машинного обучения, также известная как предиктивная персонализация контента, представляет собой передовой подход на основе ИИ к отображению контента, который лучше всего подходит для каждого пользователя. Методы машинного обучения (ML) используются при разработке веб-сайтов для предоставления пользователям улучшенного опыта.

ТЕНДЕНЦИИ РАЗВИТИЯ ДИЗАЙНА ВЕБ-САЙТОВ:

- **16. Безопасность данных:** С развитием Интернета, веб-приложений и веб-сайтов киберпреступников становится все больше. Любое веб-приложение, обрабатывающее большие объемы информации, привлекательно для мошенников. Для обеспечения безопасности разрабатываемые веб-сервисы и веб-сайты должны придерживаться принципов безопасности:
- Не отказывайтесь от тестирования безопасности веб-ресурса. Его можно проводить в ходе разработки веб-проекта. Это позволяет исключить утечку информации и важно тестировать любые изменения в приложении.
- Используйте ресурсы для мониторинга веб-сайтов. С помощью алгоритмов можно отслеживать все запросы и находить, распознавать подозрительные действия, уведомляя владельца. Своевременная реакция поможет быстро отреагировать и защитить программу.
- Уделяйте внимание выбору сторонних сервисов. Веб-разработчики должны убедиться, что поставщику таких сервисов можно доверять.
- Шифрование конфиденциальной информации. Если киберпреступник сможет получить доступ к базе данных, он не сможет извлечь выгоду из таких данных в зашифрованном виде.

ТЕНДЕНЦИИ РАЗВИТИЯ ДИЗАЙНА ВЕБ-САЙТОВ:

- 17. Технологии блокчейн: Технология блокчейн, которая представляет собой зашифрованную систему хранения данных, набирает обороты. В отличие от традиционных систем, она хранит информацию в блоках, которые связаны друг с другом в цепочку. Одним из многочисленных преимуществ этой технологии является то, что она делает транзакции более безопасными и безошибочными. Эта технология также поддерживает цифровую валюту биткойн. Использование этой криптовалюты резко возросло за последнее десятилетие из-за решения крупных платежных систем принимать биткойн для своих транзакций.
- 18. Кибербезопасность: В последние годы кибербезопасность стала главным приоритетом в веб-разработке. Безопасность стала одной из главных проблем пользователей Интернета, поскольку кибератаки, такие как потеря информации, кража личных данных; взлом, шпионаж и другие вредоносные программы, могут дискредитировать веб-сайт.

СПАСИБО ЗА
ВНИМАНИЕ

