

1. Основные принципы (особенности) JS

2. JSP Action

1. JavaScript интерпретируется браузером, код выполняется по мере его чтения, выполняется на стороне клиента. Динамическая типизация. Прототипное наследование. Функция как объекты первого класса. Асинхронное выполнение. Все идентификаторы регистровые. В названиях переменных можно использовать буквы, подчеркивание, символ доллара, арабские цифры. Названия переменных не могут начинаться с цифры. Для оформления одноточных комментариев используются `//`, многострочные и внутрострочные комментарии начинаются с `/*` и заканчиваются `*/`.

2. JSP Action - это специальные теги или инструкции, встраиваемые в JSP-страницу. `<jsp:include>`: включать содержимое другой JSP-страницы в текущую страницу. `<jsp:forward>`: для перенаправления запроса на другую JSP-страницу или ресурс. `<jsp:useBean>`: создавать или получать JavaBean объекты, которые могут использоваться для хранения данных. `<jsp:setProperty>`: для установки свойств JavaBean объектов. `<jsp:getProperty>`: для получения свойств JavaBean объектов.

1. Библиотека JQuery, её назнач и основ API

2. Архитектура Model 1 и Model 2

1. JS-библиотека, предназначенная для разработки DHTML и AJAX-приложений. Ключевым элементом API является функция (объект) `$`. Основные API включают селекторы (например, `$(element)`), методы манипуляции DOM (например, `append()`), `addClass()`, события (например, `click()`), `on()`), асинхронные запросы (например, `$.ajax()`), и анимацию (например, `animate()`). JQuery упрощает взаимодействие с веб-страницей, улучшает кроссбраузерную совместимость и ускоряет разработку.

2. Model 1 логика приложения и визуальный слой находятся в одном месте. Предназначена для проектирования приложений небольшого масштаба и сложности. Model 1 редко используется в реальных проектах современного веб-разработки, так как не обеспечивает четкого разделения между представлением и логикой приложения. Model 2 является более современной и популярной архитектурой. Она разделяет приложение на модель, представление, и контроллер. Модель - за бизнес-логику и взаимодействие с базой данных. Представление - за отображение данных и взаимодействие с пользователем. Контроллер управляет потоком данных между моделью и представлением, обрабатывает запросы от клиента. Является более распространенной и рекомендуемым подходом для разработки веб-приложений.

1. LESS, SASS, SCSS

2. JSTL - что это, зачем это.

1. LESS, SASS и SCSS - препроцессоры CSS, которые предоставляют дополнительные функции в сравнении с обычным CSS. LESS использует синтаксис, который напоминает обычный CSS, но добавляет переменные, миксины, операторы и вложенность. LESS-файлы требуют компиляции в обычный CSS перед отправки на клиентский браузер. SASS предлагает два синтаксиса: SASS (с более компактным синтаксисом, без фигурных скобок и точек с запятой) и SCSS (синтаксис, близкий к обычному CSS с фигурными скобками и точками с запятой). Оба синтаксиса поддерживают переменные, миксины и другие расширения. SCSS также поддерживает все функции SASS, включая переменные и миксины. SASS и SCSS-файлы также требуют предварительной компиляции в CSS перед использованием в браузере.

2. Расширение JSP, добавляющее возможность использования дополнительных тегов, решающих типовые задачи. Примеры задач: Условная обработка; Создание циклов, вывод массивов / коллекций; Поддержка интернационализации. Рекомендуется использовать их вместе с EL вместо скриптовых. Основные теги создания циклов, определения условий, вывода информации на страницу и т. д. `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>`
// Теги для работы с XML-документами `<%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml"%>`
// Теги для работы с базами данных `<%@ taglib prefix="s" uri="http://java.sun.com/jsp/jstl/sql"%>`
// Теги для форматирования и интернационализации // информация (i10n и i18n) `<%@ taglib prefix="f" uri="http://java.sun.com/jsp/jstl/fmt"%>`

1. Rest и RPC

2. Диспетчеризация запросов в веб-приложениях на Java. Интерфейс RequestDispatcher

1. Representational State Transfer - подход к архитектуре сетевых протоколов, обеспечивающих доступ к информационным ресурсам. Данные должны передаваться в виде небольшого числа стандартных форматов (HTML, XML, JSON). Сетевой протокол должен поддерживать кеширование, не должен зависеть от сетевого слоя, не должен сохранять информацию о состоянии между парами «запрос-ответ». RPC протокол взаимодействия между клиентом и сервером, при котором клиент может вызывать удаленные процедуры на сервере.

2. Сервлеты могут делегировать обработку запросов другим ресурсам (сервлетам, JSP и HTML-страницам). Два способа получения RequestDispatcher — через ServletRequest (абсолютный или относительный URL) и ServletContext (только абсолютный URL). Интерфейс RequestDispatcher определяет два метода для перенаправления запросов и используется для выполнения этой операции. `forward` - этот метод используется для перенаправления запроса от одного сервлета к другому. `include` - этот метод также используется для перенаправления запроса, но в отличие от `forward`, не завершает выполнение текущего сервлета и результат обработки добавляется к текущему ответу.

1. Fast CGI взаимодействие с веб-сервером

2. FreeMarker. Архитектура, принцип работы, использование в веб приложениях.

1. Веб-сервер взаимодействует с процессом через UNIX Domain Sockets или TCP/IP (а не через `stdin` и `stdout`). FastCGI предполагает использование отдельных процессов для каждого внешнего приложения, поддерживает постоянное соединение между веб-сервером и внешним приложением.

2. FreeMarker- Компилирующий обработчик шаблонов, написан на Java. Свободное ПО, распространяется по лицензии BSD. Разработчик создает шаблон, который определяет структуру страницы и вставки для данных. Приложение предоставляет данные, которые должны быть вставлены в шаблон. Шаблонный движок FreeMarker объединяет данные и шаблон, заменяя вставки в шаблоне значениями данных. После объединения данных и шаблона, генерируется контент, который может быть возвращен клиенту как HTML-страница. FreeMarker используется в веб-приложениях для генерации HTML-страниц, JSON-ответов и другого динамического контента.

1. Jakarta EE архитектура, компоненты, контейнер

2. Синтаксис шаблонов, модель данных FreeMarker

1. Jakarta EE - «Надстройка» над Java SE. Архитектура ключевые компоненты: клиенты; компоненты приложения такие как сервлеты, EJB, JMS, CDI; серверный контейнер (который предоставляет среду выполнения для компонентов приложения, обеспечивает множество служб, таких как управление жизненным циклом компонентов, безопасность, управление транзакциями, пулы соединений); также предоставляет инфраструктуру, включая JDBC для доступа к базам данных, JNDI для доступа к ресурсам и EIS, JTA для управления транзакциями. Компоненты: Сервлеты, EJB, JMS, CDI, JPA, JSP, JCA. Принципы: Inversion of Control (IoC) + Contexts & Dependency Injection (CDI); Location Transparency. 2. Шаблон может содержать следующие элементы: Статический HTML; Обращения к модели данных: `Welcome ${user}!` Директивы: `<#if animals.python.price != 0>` Python are not free today! `</#if>` Вызовы встроженных функций: `animals?filter(it -> it.protected)`. Модель данных FreeMarker - древовидная объектная структура, данные из которой шаблон использует при формировании HTML. Элементы дерева - Java Beans. При выводе в HTML все объекты преобразуются в строки. Сложность иерархии может быть любой.

1. SuperAgent

2. Что такое шаблон проектирования...

1. SuperAgent - библиотека браузеров, предоставляющая API для выполнения HTTP-запросов. Можно настраивать заголовки запроса, включая заголовки аутентификации и пользовательские заголовки, передавать данные в теле запроса, включая JSON, формы и файлы; обрабатывать ответы от сервера, включая разбор JSON и других форматов данных. `request.post('/api/pe?').send({ name: 'Manny', species: 'cat' }).set('X-API-Key', 'foobar').set('Accept', 'application/json').end(function(err, res){ if (err || !res.ok) alert('Oh no! Error!'); else alert('yay got ' + JSON.stringify(res.body)); });`

2. Шаблон проектирования или паттерн - повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Абстрактный шаблон, который может быть адаптирован и применен к разным ситуациям. Примеры включают Singleton, Factory, Observer, Strategy и многие другие. Шаблоны проектирования применяются на уровне кода и предлагают решения для конкретных задач внутри приложения. Шаблоны проектирования применяются для решения локальных задач, таких как создание объектов, управление состоянием. Позволяют избежать «типовых» ошибок при разработке типовых решений. Позволяют кратко описать подход к решению задачи — программистам, знающим шаблоны, проще обмениваться информацией. Легче поддерживать код — его поведение более предсказуемо.

1. Структура HTML документа

2. Сервлеты - особенности, преимущества и недостатки относительно CGI, FastCGI

1. Документ состоит из элементов. Начало и конец элемента обозначаются тегами. Теги могут иметь атрибуты. Элементы могут быть вложенными. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN http://www.w3.org/TR/html4/strict.dtd.>` `<html>`: Корневой элемент. `<head>`: Содержит метainформацию о документе и заголовков. `<meta>`: Метатеги для определения кодировки. `<title>`: Заголовок документа. `<link>`: Подключение внешних CSS стилей. `<script>`: Вставка скриптов. `<body>`: Основное содержимое документа.

2. Сервлеты — это серверные сценарии, написанные на Java. Жизненным циклом сервлетов управляет веб-контейнер (он же контейнер сервлетов). В отличие от CGI, запросы обрабатываются в отдельных потоках (а не процессах) на веб-контейнере. Выполняются быстрее, чем CGI-сценарии. Хорошая масштабируемость. Надежность и безопасность (реализованы на Java). Платформенно-независимы. Множество инструментов мониторинга и отладки. Слабое разделение уровня представления и бизнес-логики. Возможны конфликты при параллельной обработке запросов.

1. PHP особенности синтаксиса, использование в веб-приложениях

2. LongPolling и Websockets зачем нужны, сходства и различия

1. PHP код встраивается непосредственно в HTML документы, обрамляясь открывающим тегом `<?php` и закрывающим тегом `?>`. Переменные начинаются с символа доллара (`$`). Инструкции разделяются символом `:`. Динамически типизированный язык. `integer`, `float`, `double`, `boolean`, `string` и `NULL`. Диапазоны числовых типов зависят от платформы. Не скалярные типы - ресурс (например, дескриптор файла), массив и объект. Псевдотипы — `mixed`, `number`, `callback` и `void`. Суперглобальные массивы `$_GLOBALS`, `$_SERVER`, `$_ENV`, `$_GET`, `$_POST`, `$_FILES`, `$_COOKIE`, `$_REQUEST`, `$_SESSION`. Реализованы все основные механизмы ООП — инкапсуляция, полиморфизм и наследование. Можно объявлять финальные и абстрактные методы и классы. Множественное наследование не поддерживается, но есть интерфейсы и механизм особенностей (traits).

2. Long Polling и WebSockets - это технологии для обмена данными между клиентом и сервером в реальном времени. Сходства: Обеспечивают двустороннюю связь между клиентом и сервером. Поддерживают обновление данных без перезагрузки страницы. Различия: Long Polling - клиент отправляет запрос серверу и ждет ответа, что может создавать задержку. WebSocket — протокол полнодуплексной связи поверх TCP-соединения. Позволяет серверу отправлять данные браузеру без дополнительного запроса со стороны клиента. Альтернатива — AJAX + Long Polling.

1. Конфигурация PHP, способы интеграции PHP с веб-приложением

2. ServletContext - что это и для чего применяется

1. Настройки PHP хранятся в файле `php.ini`. Можно подключать дополнительные модули, расширяющие возможности языка. Использование интерпретатора PHP: С помощью SAPI / ISAPI; С помощью CGI / FastCGI; Через интерфейс командной строки; FastCGI - это более современный и гибкий способ интеграции PHP. Можно настроить PHP как FastCGI-процесс, который взаимодействует с веб-сервером через протокол FastCGI.

2. API, с помощью которого сервлет может взаимодействовать со своим контейнером. Доступ к методам осуществляется через интерфейс `javax.servlet.ServletContext`. У всех сервлетов внутри приложения общий контекст. В контекст можно помещать общую для всех сервлетов информацию (методы `getAttribute` и `setAttribute`). Если приложение — распределенное, то на каждом экземпляре JVM контейнером создается свой контекст.

1. Как сервлеты обрабатывают HTTP запрос

2. Многоуровневая архитектура: элементы, зачем нужна

1. Браузер формирует HTTP-запрос и отправляет его на сервер. Веб-контейнер создает объекты `HttpRequest` и `HttpResponse`. Сервер при запуске инициализирует сервлет, вызывая метод `init()`. После инициализации сервлет ожидает входящих HTTP-запросов от клиентов. Для каждого запроса вызывается метод `service()`, который перенаправляет выполнение соответствующего метода, такого как `doGet()`, `doPost()`, `doPut()`, `doDelete()`. Сервлет формирует ответ и записывает его в поток вывода `HttpResponse`.

2. Многоуровневая архитектура - это подход к проектированию программных систем, который разделяет систему на несколько уровней, каждый из которых выполняет определенные функции. Основные элементы многоуровневой архитектуры включают: Presentation; Application Logic; Data; Business; Service. Повышает модульность, совместимость и масштабируемость приложений, делая их более устойчивыми и легкими в сопровождении.

1. Цикл жизни сервлета

2. Дialeкты и процессоры Thymeleaf и стандартный диалект

1. Сервлет инициализируется `init()` один раз при его загрузке в веб-контейнер. Каждый запрос обрабатывается с использованием метода `service()`, который вызывает соответствующий метод (`doGet()`, `doPost()`, и т. д.). При завершении работы сервлета или при выгрузке веб-приложения, метод `destroy()` вызывается, что позволяет освободить ресурсы.

2. Диалект состоит из одного или нескольких процессоров (processors). Процессор - объект, который применяет некоторую логику к формируемому на основе шаблона артефакту. «Из коробки» Thymeleaf содержит стандартный диалект (Standard Dialect), которого достаточно для решения большинства типовых задач. Standard Dialect содержит набор процессоров, предназначенных для решения типовых задач. Большая часть процессоров стандартного диалекта — процессоры атрибутов (Attribute Processors). Процессоры атрибутов обрабатывают дополнительные («нестандартные») атрибуты тегов Thymeleaf — модульный движок. Модуль Thymeleaf называется диалектом (dialect).

1. AJAX и DHTML - описание, сходства и различия

2. Проблемы при параллельной обработке запросов в JSP, как этого можно избежать

1. AJAX (Asynchronous Javascript and XML) — подход к построению интерактивных пользовательских интерфейсов веб-приложений. Основан на «фоновом» обмене данными браузера с веб-сервером. При обмене данными между клиентом и сервером веб-страница не перезагружается полностью. Использует технологии динамического обращения к серверу, без перезагрузки всей страницы полностью, например: с использованием XMLHttpRequest; через динамическое создание дочерних фреймов; через динамическое создание тега <script>. Использование DHTML для динамического изменения содержания страницы. Dynamic HTML — способ создания интерактивного веб-сайта, использующий сочетание: статичного языка разметки HTML; выполняемого на стороне клиента скриптового языка JavaScript; CSS (каскадных таблиц стилей); DOM (объектной модели документа).

2. Параллельные запросы могут конфликтовать за общие ресурсы, приводя к блокировкам и задержкам. Использование глобальных переменных может привести к состоянию гонки и ошибкам. Решения: Использовать синхронизацию для общих ресурсов. Избегать глобальных переменных и использовать локальные для каждого запроса. А так же использовать атомарные переменные

1. HTTP методы

2. Thymeleaf. Особенности архитектуры. Отличия от Freemarker.

1. GET - получения данных с сервера. Идемпотентны и могут содержать параметры только в URL. POST - создания или обновления ресурсов. Могут содержать тело запроса. PUT - обновления или создания ресурса. Обычно идемпотентны. DELETE - удаление ресурса на сервере. Идемпотентны. PATCH - частичное обновление ресурса на сервере. HEAD - похож на GET, но сервер отвечает только заголовками без тела ответа. OPTIONS - получение информации о возможных методах. CONNECT - установление сетевого соединения с ресурсом.

2. Thymeleaf - компилирующий обработчик шаблонов. Thymeleaf шаблоны могут быть HTML, что облегчает их редактирование. Thymeleaf поддерживает разные языки шаблонов, включая HTML, XML, JavaScript и др. Отличия от Freemarker: Thymeleaf использует более натуральный синтаксис в HTML, а Freemarker использует свой собственный синтаксис. Thymeleaf обеспечивает удобную и читаемую разработку веб-страниц, в то время как Freemarker более общего назначения и может использоваться в разных сценариях.

1. CSS : назначение, правила, приоритеты

2. MVC : назначение, элементы, примеры реализации

1. CSS - это язык, используемый для определения внешнего вида и форматирования веб-страниц. CSS предназначен для разделения структуры (HTML) и стиля веб-страницы. Он облегчает управление дизайном и обеспечивает согласованный внешний вид для всех страниц сайта. Таблица стилей состоит из набора правил. Каждое правило состоит из набора селекторов и блока определений. Если к одному элементу «подходят» сразу несколько стилей, применен будет наиболее приоритетный. Приоритеты рассчитываются таким образом (от большего к меньшему): свойство задано при помощи !important; стиль прописан напрямую в теге; наличие идентификаторов (#id) в селекторе; количество классов (.class) и псевдоклассов (:pseudoclass) в селекторе; количество имен тегов в селекторе.

2. MVC - это шаблон проектирования, который используется для разделения приложения на три основных компонента: Model(бизнес-логика и данные приложения), View(отображение данных пользователем) и Controller(обрабатывает запросы пользователя, взаимодействует с моделью и обновляет представление). MVC помогает создавать структурированные и легко поддерживаемые приложения. Предназначен для проектирования достаточно сложных веб-приложений.

1. Структура http-запроса

2. Типы данных в PHP

1. HTTP-запрос представляет собой сообщение, отправляемое клиентом серверу. В стартовую строку входит: метод, который указывает, какое действие должен выполнить сервер; URI представляет собой адрес ресурса, к которому выполняется запрос; версия протокола. Заголовки представляют собой пары «ключ-значение»(Host; User-Agent; Accept; Connection). Тело запроса содержит данные, отправляемые на сервер. Это обычно используется при методе POST, когда данные формы отправляются на сервер.

2. PHP — язык с динамической типизацией; при объявлении переменных их тип не указывается. 6 скалярных типов данных — integer, float, double, boolean, string и NULL. Диапазоны числовых типов зависят от платформы. 3 нескаларных типа — ресурс (например, дескриптор файла), массив и объект. 4 псевдотипа — mixed, number, callback и void. Предопределенные массивы, имеющие глобальную область видимости: \$GLOBALS — массив всех глобальных переменных. \$SERVER — параметры, которые ОС передает серверу при его запуске. \$ENV — переменные среды ОС. \$GET, \$POST — параметры GET- и POST-запроса. \$FILES — сведения об отправленных методом. \$COOKIE — массив cookies. POST файлах. \$REQUEST — содержит элементы из массивов. \$_SESSION — данные HTTP-сессии.

1. JSP - элементы

2. CGI обработка запроса, преимущества и недостатки

1. 2 варианта синтаксиса — на базе HTML и XML. Обозначаются тегами <% %> (HTML-вариант): <html> <%- scripting element -%> </html>. Существует 5 типов JSP-элементов: Комментарий — <%- Comment -%>; Директива — <%@ directive %>; Объявление — <%! decl %>; Скриплет — <% code %>; Выражение — <%= expr %>. Поддерживаются 3 типа комментариев: HTML-комментарии: <!-- -->; JSP-комментарии: <%- -%>. Java-комментарии: <% /* comment */ %>. <%@ DirectiveName [attr="value"]* %>. <%! JavaClassDeclaration %>. <% scriptlet %>. <%= JavaExpression %>.

2. Пользователь в браузере отправляет HTTP-запрос к веб-серверу, указывая путь к CGI-скрипту и метод. Каждый запрос обрабатывается отдельным процессом CGI-программы. Веб-сервер принимает запрос и запускает CGI-скрипт. CGI-скрипт формирует HTTP-ответ, включая заголовки и тело ответа, и выводит его через stdin и stdout. Программы могут быть написаны на множестве языков программирования. «Падение» CGI-сценария не приводит к «падению» всего сервера. Исключены конфликты при параллельной обработке нескольких запросов. Хорошая поддержка веб-серверами. Высокие накладные расходы на создание нового процесса. Плохая масштабируемость. Слабое разделение уровня представления и бизнес-логики. Могут быть платформо-зависимыми.

1. Filter пост предобработка

2. Thymeleaf выражения. Стандартные выражения

1. В контексте веб-приложений, фильтры представляют собой компоненты, которые позволяют выполнить дополнительную обработку запросов и ответов до того, как они достигнут сервера или после его выполнения. Пример предобработки — допуск к странице только авторизованных пользователей. Пример постобработки — запись в лог времени обработки запроса. Фильтр будет выполнен при обработке каждого запроса, проходя через него. В методе doFilter(), можно выполнять предобработку данных перед передачей запроса и ответа по цепочке (chain.doFilter(req, resp)). Также можно выполнять пост-обработку данных после выполнения сервера.

2.Значения атрибутов присваиваются путём вычисления выражений (expressions). \${variable}: для вывода значения переменной. #{Selection}. #{Message}. @Link). {Fragment} theach: для итерации по коллекциям. th:if и th:unless: задавать условия. th:src, th:href, th:alt: для задания соответствующих значений атрибутов. th:attr: задать значения пользовательских атрибутов. th:text: текстовое содержание HTML-элементов. th:utext: выводит HTML-разметку как текст. th:text=""#{message.code}": для локализации текстовых сообщений. th:action: URL-адрес для действия формы. th:object: Устанавливает объект модели для формы. th:include: Включение других шаблонов. th:replace: Замена содержимого.

1. DOM и BOM

2. Управление сессиями. HttpSession

1. DOM — это платформо-независимый интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-документов. Стандартизирована W3C. Документ в DOM представляет собой дерево узлов. Узлы связаны между собой отношением «родитель-потомок». Используются для динамического изменения страниц HTML. BOM — прослойка между ядром и DOM. Основное предназначение — управление окнами браузера и обеспечение их взаимодействия. Каждое из окон браузера представляется объектом window. Возможности BOM: управление фреймами; поддержка задержки в исполнении кода и закидывания с задержкой; системные диалоги; управление адресом открытой страницы; управление информацией о браузере; управление информацией о параметрах монитора; ограниченное управление историей просмотра страниц; поддержка работы с HTTP cookie.

2. Управление сессиями - этот аспект позволяет приложениям отслеживать состояние и взаимодействие с каждым пользователем на веб-сайте. HttpSession - это интерфейс, который предоставляет средства для создания и управления сессиями пользователей. Сессия создается, когда пользователь впервые взаимодействует с веб-приложением, и ему присваивается уникальный идентификатор сессии.

1. ECMA script преимущества 6 и 7 версий

2. GoF паттерны. Что это такое? Основные виды, примеры

1. ES6 Добавляет в синтаксис языка множество новых возможностей. Для работы в старых браузерах может потребоваться специальная программа — транслятор (transpiler). Для работы в старых браузерах может потребоваться специальная программа — транспилер. Два новых ключевых слова let и const. Новый этап деструктуризации. var x = (function () {}); Новые ключевые слова extends и Super. for (const element of array). Некоторые из изменений в ES7 включают оператор возведения в степень и метод Array.prototype.includes. ES7 представил ключевые слова async и await, которые упрощают работу с асинхронным кодом и делают его более понятным.

2. GoF паттерны - это набор шаблонов проектирования. Порождающие паттерны (Singleton, Factory Method, Abstract Factory) помогают решать задачи, связанные с созданием сущностей или групп похожих сущностей. Структурные паттерны (Adapter, Decorator, Composite) заботятся о том, как сущности могут использоваться друг друга и как из классов и объектов образуются более крупные структуры. Поведенческие паттерны (Observer, Strategy, Chain of Responsibility) - поведенческие паттерны проектирования отвечают за эффективное взаимодействие объектов и шаблоны для обмена сообщениями.

1. Преимущества и недостатки ajax

2. Директива page: назначение, особенности, атрибуты

1. AJAX обеспечивает следующие преимущества при реализации в правильных местах: пользователь с AJAX может сделать много без обновления страницы, которая делает веб-приложения ближе к обычным настольным приложениям. Лучшая производительность - путем обмена только требуемыми данными внутри сервера, AJAX сохраняет пропускную способность и увеличивает скорость приложений. Значительный недостаток AJAX - пробелы в безопасности. Каждый пользователь может легко посмотреть исходный код в браузере. Невозможность интеграции с инструментами браузера. В процессе динамического формирования страниц браузер не может отображать их в истории посещения.

2. Позволяет задавать параметры, используемые контейнером при управлении жизненным циклом страницы. Обычно расположена в начале страницы. На одной странице может быть задано несколько директив page с разными указаниями контейнеру. Синтаксис: <%@ page attribute="value"%> Она может быть использована для указания различных атрибутов, таких как кодировка, импорт пакетов, тип контента и другие. buffer — параметр буферизации; autoFlush — автоматическая выгрузка при переполнении буфера; contentType — Type+кодировка; errorPage — при RuntimeException; isErrorPage; extends — для наследования; import; info — строка на getServletInfo(); isThreadSafe — блок параллельной обработки нескольких запросов; language — ЯП; session — создавать ли предопределенную session; isElIgnored — вычисляются ли EL выражения

1. Коды состояния HTTP

2. JSP Expression Language, что такое, зачем нужно, чем отличается от JSP, как обрабатывается веб контейнером

1. Состоят из 3-х цифр. Первая цифра — класс состояния: «1» — Informational — информационный; «2» — Success — успешно; «3» — Redirection — перенаправление; «4» — Client error — ошибка клиента; «5» — Server error — ошибка сервера.

2. JSP Expression Language - это язык выражений, используемый в технологии JSP для вставки и обработки данных в веб-страницах. Позволяет использовать на страницах арифметические и логические выражения. Поддерживается «из коробки», можно отключить в настройках конкретной страницы и / или приложения. JSP EL отличается от обычных JSP-скриплетов тем, что он предоставляет более простой способ вставки данных в веб-страницы, не требуя явного написания Java-кода. Он обрабатывается веб-контейнером автоматически при выполнении JSP-страницы, и его результаты вставляются в HTML-код страницы перед отправкой клиенту.

1. ECMA script

2. Правила трансляции JSP

1. ECMAScript является ядром js. Встраиваемый расширяемый не имеющий средств ввода/вывода язык программирования. Объектный тип данных - Object. 15 различных видов инструкций. Блок не ограничивает область видимости функции. Если переменная объявляется вне функции, то она попадает в глобальную область видимости Он поддерживает динамическую типизацию, поддерживается практически всеми современными браузерами. Функция - это тоже объект.

2. Код JSP-страницы транслируется в Java-код сервера с помощью компилятора JSP-страниц Jasper. Директивы управляют процессом трансляции страницы. На фазе трансляции каждый тип данных в странице JSP интерпретируется отдельно. Шаблонные данные трансформируются в код, который будет помещать данные в поток, возвращающий данные клиенту. Элементы JSP трактуются: директивы, используемые для управления тем, как Web-контейнер переводит и выполняет страницу JSP; скриптовые элементы вставляются в класс сервера страницы JSP; элементы в форме <jsp:XXX ... /> конвертируются в вызов метода для компонентов JavaBeans или вызовы API Java Servlet.

1. FastCGI. Плюсы, минусы, отличия от CGI.

2. Суперглобальные массивы в PHP (SuperGlobal massive)

1. FastCGI - это клиент-серверный протокол взаимодействия между веб-сервером и приложением. FastCGI использует многопоточность, что позволяет уменьшить потребление памяти и достичь более высокой скорости. FastCGI может кэшировать некоторые промежуточные данные. Все запросы могут обрабатываться одним процессом CGI-программы (фактическая реализация определяется программистом). Веб-сервер взаимодействует с процессом через UNIX Domain Sockets или TCP/IP (а не через stdin и stdout). Главное отличие FastCGI от стандартного CGI заключается в том, что FastCGI поддерживает постоянное подключение процессов приложения между запросами.

2. Суперглобальные массивы в PHP представляют собой массивы, доступные в любой области видимости скрипта. \$GET - содержит данные, переданные в GET запросе. \$POST - используется для получения данных из форм и других HTTP POST-запросов. \$COOKIE - содержит данные, переданные клиентом в виде cookie. \$SESSION: Этот массив позволяет хранить данные между запросами. \$REQUEST: Массив, объединяющий данные из \$GET, \$POST и \$COOKIE. \$SERVER - содержит данные о сервере и окружении выполнения скрипта(адрес сервера, метод запроса, HTTP заголовки). \$ENV: содержит переменные окружения, переданные сервером.

1. Правила построения html-форм

2. Конфигурация сервлетов. Файл web.xml

1. Предназначены для обмена данными между пользователем и сервером. Документ может содержать любое число форм, но одновременно на сервер может быть отправлена только одна из них. Вложенные формы запрещены. Границы формы задаются тегами `<form>...</form>`. Метод HTTP задаётся атрибутом `method` тэга `<form method="GET" action="URL">...</form>`

2. Корневой элемент XML-файла web.xml, который определяет структуру и параметры веб-приложения. `servlet` - конфигурация сервлета. Внутри него могут быть указаны имя сервлета, класс сервлета и др. `servlet-mapping` - Элемент, который связывает URL-шаблон с определенным сервлетом, определяя, какой сервлет будет обрабатывать запросы, соответствующие этому URL. `servlet-name` - имя сервлета, которое используется для связывания с сервлетом. `url-pattern` - элемент, определяющий URL-шаблон, который будет использоваться для маппинга сервлета на конкретные URL-адреса. `init-param` - определяет параметры и значения, которые могут быть использованы сервлетом при инициализации. `load-on-startup` указывает, что сервлет должен быть инициализирован при запуске приложения. `filter` - используемый для определения фильтра и его параметров. `filter-mapping` - связывает фильтр с URL-шаблоном.

1. Реализация объектно-ориентированных программ в PHP

2. Предопределенные переменные JSP

1. Полная поддержка появилась в PHP5. Реализованы все основные механизмы ООП — инкапсуляция, полиморфизм и наследование. Поля и методы могут быть `private`, `public` и `protected`. Можно объявлять финальные и абстрактные методы и классы (аналогично Java). Множественное наследование не поддерживается, но есть интерфейсы и механизм особенностей (traits). Обращение к константам, статическим свойствам и методам класса осуществляется с помощью конструкции «::<». Передача объектов по ссылке.

2. В процессе трансляции контейнер добавляет в метод `_jspService` ряд объектов, которые можно использовать в скриптах и выражениях: `application`; `config`; `exception`; `out`; `page`; `PageContext`; `request`; `response`; `session`. `Exception` — используется только на страницах-перенаправлениях. `Page` — API для доступа к экземпляру класса сервлета, в который транслируется JSP. `PageContext` — контекст JSP-страницы.

1. HTTP, реализация и особенности

2. Жизненный цикл JSP

1. Протокол прикладного уровня. Основа — технология «клиент-сервер». Может быть использован в качестве «транспорта» для других протоколов прикладного уровня. Основной объект манипуляции — ресурс, на который указывает URI. Stateless-протокол (состояние не сохраняется). Для реализации сессий используются cookies. Серверы, такие как Apache, Nginx, и Microsoft IIS, реализуют протокол HTTP для обслуживания клиентских запросов.

2. JSP контейнер проверяет код JSP страницы, парсит ее для создания кода сервлета. Трансляция `jsp` в код сервлета, компиляция сервлета, загрузка класса сервлета, создание экземпляра сервлета, вызов `jspInit()`, внедрение конструкторов без параметров созданных классов для инициализации в памяти классов, длительный жизненный цикл обработки запросов клиента JSP страницей, вызов `jspService()`, вызов `jspDestroy()`.