1. Что такое jsp? Каков ее состав и назначение?

**Java Server Pages** представляет технологию, которая позволяет создавать динамические веб-страницы.

JSP позволяет разработчику:

* получать данные из веб-страницы в Java-код;
* отправлять данные из Java кода на веб-страницу;
* писать Java-код, прямо внутри html (однако злоупотреблять этим не стоит).

Необходимость знания JSP можно оценить довольно высоко по нескольким причинам:

* JSP — одна из основных Java web-технологий;
* JSP широко используется в большинстве компаний и проектов;
* JSP бесшовно интегрируется с сервлетами Java внутри контейнера сервлетов.

Технология Java Server Pages содержит четыре ключевых компонента:

1. **Директивы** *(directive)* представляют собой сообщения для контейнера JSP, дающим возможность определить параметры страницы, подключения других ресурсов, использовать собственные нестандартные библиотеки тегов.
2. [Действия actions](https://java-online.ru/jsp-actions.xhtml) инкапсулируют функциональные возможности в предопределенных тегах, которые можно встраивать в JSP-страницу. JSP actions часто выполняются на основе информации, посылаемой на сервер в составе запроса от определенного клиента. Действия также могут создавать объекты Java для использования их в скриптлетах JSP.
3. [Скриптлеты scriptlets](https://java-online.ru/jsp-syntax.xhtml#scriptlets) позволяют вставлять код Java в страницы JSP, который взаимодействует с объектами страницы при обработке запроса.
4. [Библиотеки тегов](https://java-online.ru/jsp-taglib.xhtml) *(tag library)* являются составной частью механизма расширения тегов, допускающего разработку и использование собственных тегов.

2. Расскажите о жизненном цикле jsp?В чем отличие jsp и servlet?

Жизненный цикл. Первое обращение

1) браузер => JSP

2) JSP-engine

a) анализирует JSP

b) создает сервлет с кодом на основе JSP

c) статическое содержимое в методы вывода помещает в \_jspService() -- сервлет будет ответственен за генерацию статических элементов, определенных во время разработки

d) динамические элементы => в java-код

3) код сервлета компилируется \*.class. и загружается в контейнер

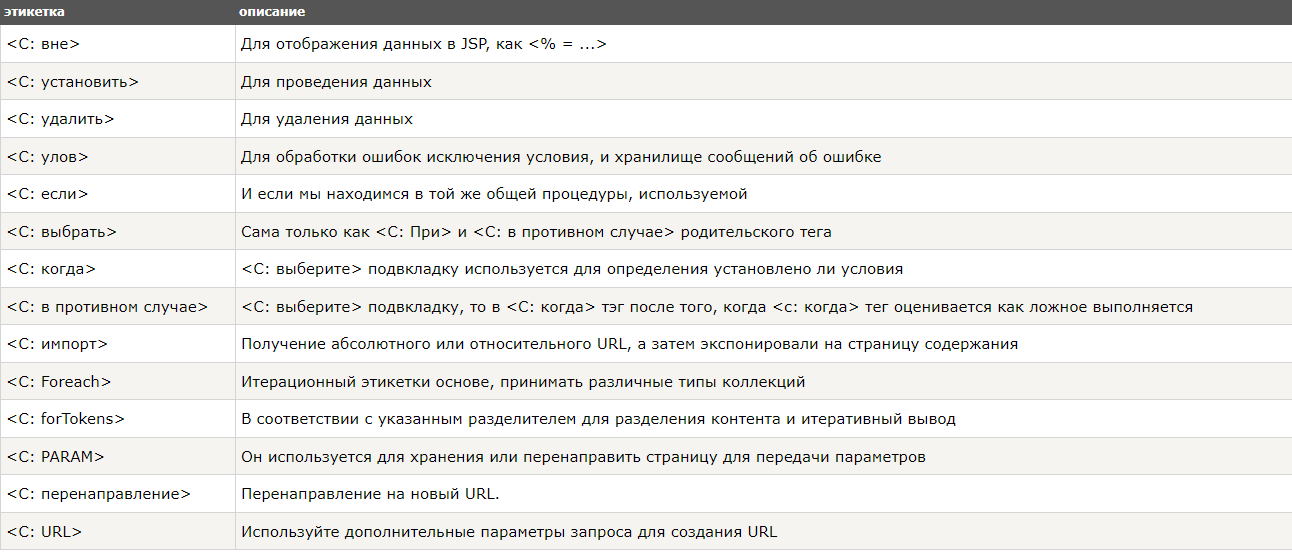
4) выполняется метод init() сервлета

5) вызываeтся метод \_jspService() = > сервлет логически исполняется = > response;

6) статический HTML + графика + результат исполнения динамических элементов JSP = > пересылаются браузеру через HttpServletResponse

Ключевое **отличие**: «**Jsp** (**Java** **Server** **Pages**)» - это технология на стороне сервера, тогда как «**Servlet**» - это небольшая программа, работающая на сервере.

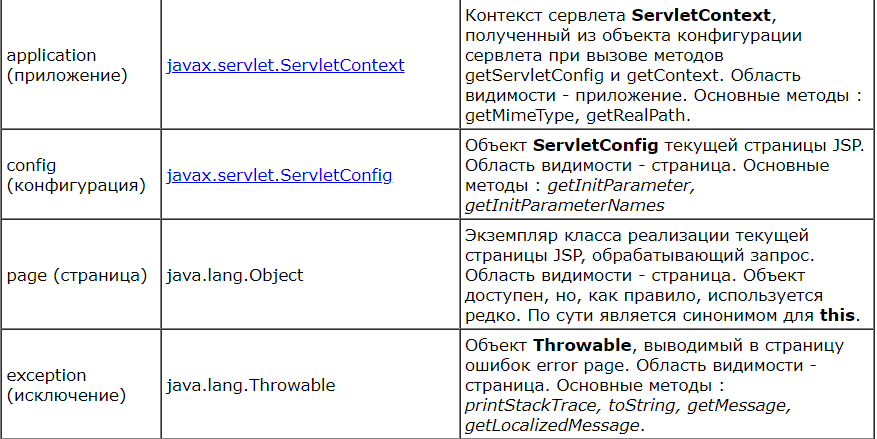
3. Перечислите теги jsp и поясните их назначение.



4. Перечислите неявные объекты jsp.

Неявные объекты (**implicit objects**) - это объекты, автоматически доступные как часть стандарта JSP без их специального объявления или импорта. Эти объекты, список которых представлен в таблице, можно использовать в коде JSP страницы.





5. Какие области видимости для переменных jsp существуют?

В JSP предусмотрены следующие области действия переменных (объектов):

request — область действия запроса;

session — область действия сессии;

application — область действия приложения;

page — область действия страницы.

6. Что такое PageContext?

Класс **PageContext**-это абстрактный класс, предназначенный для расширения с целью обеспечения его зависимой от реализации реализации соответствующими средами выполнения JSP engine. Экземпляр **PageContext** получается классом реализации JSP путем вызова метода JspFactory.getPageContext() и освобождается вызовом JspFactory.releasePageContext().

Экземпляр класса **PageContext** предоставляет доступ ко всем пространствам имён, ассоциированным с JSP-страницей, к различным атрибутам страницы, а также слой поверх деталей реализации.

7. Что такое EL, как он используется ?

Язык выражений JSP (EL) позволяет легко получать доступ к данным приложения, хранящимся в компонентах JavaBeans. JSP EL позволяет вам создавать выражения **(а)** арифметические и **(б)** логические. В выражении JSP EL вы можете использовать **целые числа, числа с плавающей запятой, строки, встроенные константы true и false** для логических значений и null.

Expression Language (EL) ► Назначение: упрощения доступа к данным (атрибутам, параметрам и т. п.), хранящимся в различных областях видимости: page, request, session, application и вычисления простых выражений ${имя\_атрибута}

Типы EL операторов Стандартные операторы отношения: == (или eq), != (или ne), < (или lt), > (или gt), <= (или le), >= (или ge) Арифметические операторы: +, -, \*, / (или div), % (или mod) Логические операторы: && (или and), || (или or), ! (или not) Оператор empty

8. Как задать и настроить error page?

Файл web.xml позволяет указать, какие страницы html или jsp будут отправляться пользователю при отправке статусных кодов ошибок. Для этого в web.xml применяется элемент **<error-page>**.

Внутри этого элемента с помощью элемента **<error-code>** указывается статусный код ошибки, который надо обработать. А элемент **<location>** указывает на путь к странице html или jsp, которая будет отправляться пользователю.

<error-page>

<error-code>500</error-code>

<location>/server\_error.html</location>

</error-page>

<error-page>

<error-code>404</error-code>

<location>/file\_not\_found.html</location>

</error-page>

9. Расскажите о взаимодействии jsp-servlet-jsp.

Нередко страница jsp обрабатывает запрос вместе сервлетом. В этом случае сервлет определяет логику, а jsp - визуальную часть. И при обработке запроса сервлет может перенаправить дальнейшую обработку странице jsp. Соответственно может возникнуть вопрос, как передать данные из сервлета в jsp?

Есть несколько способов передачи данных из сервлета в jsp, которые заключаются в использовании определенного контекста или scope. Есть несколько контекстов для передачи данных:

* **request** (контекст запроса): данные сохраняются в HttpServletRequest
* **session** (контекст сессии): данные сохраняются в HttpSession
* **application** (контекст приложения): данные сохраняются в ServletContext

Данные из контекста запроса доступны только в пределах текущего запроса. Данные из контекста сессии доступны только в пределах текущего сеанса. А данные из контекста приложения доступны постоянно, пока работает приложение.

Но вне зависимости от выбранного способа передача данных осуществляется с помощью метода **setAttribute(name, value)**, где name - строковое название данных, а value - сами данные, которые могут представлять различные данные.

Наиболее распространенный способ передачи данных из сервлета в jsp представляют атрибуты запроса. То есть у объекта HttpServletRequest, который передается в сервлет, вызывается метод **setAttribute()**. Этот метод устанавливает атрибут, который можно получить в jsp.