**Написать на JS функцию, которая на странице заменяет все тестовые поля ввода на кнопки**

1. function replaceAllTextInputs(){
2. Array.from(document.querySelectorAll("input[type=text], textarea"))
3. .forEach(input => {
4. var button = document.createElement('button');
5. button.innerHTML = "I am button";
6. input.replaceWith(button);
7. });
8. }

**Написать JS-функцию, которая запрещает вводить любые символы, кроме цифр и букв латинского алфавита**

Изображение выглядит как текст

Автоматически созданное описание

**Написать CSS правило, которое будет обводить все картинки в классе news в рамку при наведении мышю**

1. .news img:hover{
2. border: 2px solid #ff0000;
3. }

**Написать css правило, которое повернёт все картинки в форме с id=sampleForm на 90 градусов по часовой стрелке**

#sampleForm {  
 transform: rotate(90);  
}

**Реализовать функцию на JavaScript, которая будет закрывать текущее окно, если в нем открыт** [**https://www.google.ru**](https://www.google.ru)

1. function close\_window() {
2. if (window.location.href===”https://www.google.ru”) {
3. window.close();
4. }
5. }

**Написать js функцию, которая заменяет содержимое <div> с именем класса “nyan” на изображение по ссылке:** [**http://www.example.com/nyancat.gif**](http://www.example.com/nyancat.gif)

1. function replaceWithNyan(){
2. let img = document.createElement('img');
3. img.src = 'http://www.example.com/nyancat.gif';
4. Array.from(document.querySelectorAll('.nyan')).forEach(
5. div => {
6. div.innerHTML = '';
7. div.appendChild(img);
8. }
9. );
10. }

**Правило css, меняющее цвет фона на желтый, если ссылка посещена и не лежит в классе “news”**

1. \*:not(.news)>a:visited {
2. color: yellow;
3. }

**php скрипт, который достаёт из get запроса имя и фамилию и приветствует пользователя, выводя html страницу**

1. <?php
2. $name = $\_GET['name'];
3. echo 'Hello ' . ($name?$name:'noname') . '!';
4. **?>**

**написать на php класс**

1. class User{
2. private $userName;
3. private $age;
5. function \_\_constuctor($userName, $age){
6. $this->userName = $userName;
7. $this->age = $age;
8. }
10. function whois(){
11. return 'Name: ' . $this->userName . '\nAge: ' . $this->age;
12. }
13. }

**сервлет перенаправляющий все запросы на страницу google**

1. class ToGoogleServlet extends HttpServlet{
2. public void service(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
3. response.sendRedirect("http://google.com1;")
4. }
5. }

**Написать сервлет, который принимает из http запроса параметр name и выводит его. Если параметр не обнаружен то вывести Anonymous user**

1. class HelloServlet extends HttpServlet{
2. public void service(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
3. response.setContentType("text/html; charset=UTF-8");
5. String name = request.getParameter("name");
6. PrintWriter out = response.getWriter();
8. name = name==null?"Anonymous user":name;
9. out.println("<h1>Hello " + name + "</h1>");
10. out.close();
11. }
12. }

**Страница JSP, проверяющая есть ли /какой-то параметр/ в запросе и если нету - выводящая сообщение об ошибке**

1. <%-- Страница JSP, проверяющая есть ли /какой-то параметр/ в запросе и если нету - выводящая сообщение об ошибке  --%>
3. <%@ page contentType="text/html; charset=UTF-8" language="java"%>
4. <%= request.getParameter("parameterName")==null?"400 - your did not set parameter":request.getParameter("parameterName")  %>

**Код фильтра запросов, запрещающий доступ к приложению неавторизированным пользователям(у неавт пол в запросе отсутствует заголовок x-application-user)**

1. class AuthFilter extends ServletFilter{
2. public void init(FilterConfig config){}
4. public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException{
5. String auth = request.getHeader("x-application-user");
6. if (auth == null || auth.equals("")){
7. PrintWriter out = response.getWriter();
8. response.setContentType("text/html; charset=UTF-8");
9. out.println("WTF?! You are not authorized");
10. }else{
11. chain.doFilter(request, response);
12. }
14. }
16. public void destroy(){}
17. }

**Код jsp-страницы показывающий содержимое корзины юзера. Содержимое корзины - коллекциия объектов класса ShoppingItem который содержит имя, стоимость и количество заказанного товара - хранится в отдельном managed bean.**

1. <%@ import="my.package.ShoppingItem" %>
2. <%@ import="java.util.Collection" %>
3. <%@ contentType="text/html;charset=UTF-8" language="java"%>
4. <jsp:useBean id="managed" class="my.package.ManagedBean" scope="session">
5. <html>
6. <head>
7. <meta charset="utf-8">
8. </head>
9. <body>
10. <table>
11. <tr>
12. <th>name</th>
13. <th>price</th>
14. <th>count</th>
15. </tr>
16. <%
17. Collection<ShoppingItem> basket = mananged.getBasket();
18. for (ShoppingItem position: basket){%>
19. <tr>
20. <td><%= position.getName() %></td>
21. <td><%= position.getPrice() %></td>
22. <td><%= position.getCount() %></td>
23. </tr>
24. <%}%>
25. </table>
26. </body>

**написать css правило, которое при клике на ссылку добавляет ей подчеркивание, всем кроме ссылок в теге h1**

1. \*:not(h1)>a:active {
2. text-decoration: underline;
3. }