enum ScrollRestoration { "auto", "manual" };

[Exposed=Window]

interface History {

readonly attribute unsigned long length;

attribute ScrollRestoration scrollRestoration;

readonly attribute any state;

undefined go(optional long delta = 0);

undefined back();

undefined forward();

undefined pushState(any data, DOMString unused, optional USVString? url = null);

undefined replaceState(any data, DOMString unused, optional USVString? url = null);

};

**history.length**

Возвращает общее количество записей истории сеанса для текущего проходимого объекта навигации.

**history.scrollRestoration**

Возвращает режим восстановления прокрутки записи журнала активных сеансов.

**history.scrollRestoration = value**

Установите режим восстановления прокрутки записи истории активного сеанса равным значению.

**history.state**

Возвращает классическое состояние API истории записи истории активного сеанса, десериализованное в значение JavaScript.

**history.go()**

Перезагружает текущую страницу.

**history.go(delta)**

Возвращает назад или вперед указанное количество шагов в общем списке записей истории сеанса для текущего проходимого объекта навигации.

Нулевая разница приведет к перезагрузке текущей страницы.

Если дельта находится вне зоны действия, ничего не происходит.

**history.back()**

Возвращает на один шаг назад в общем списке записей истории сеанса для текущего проходимого навигационного объекта.

Если предыдущей страницы нет, ничего не делает.

**history.forward()**

Перемещается на один шаг вперед в общем списке записей истории сеанса для текущего проходимого навигационного объекта.

Если следующей страницы нет, ничего не делается.

**history.pushState(data, "")**

Добавляет новую запись в историю сеансов с классическим состоянием API истории, настроенным на сериализацию данных. URL-адрес активной записи истории будет скопирован и использован в качестве URL-адреса новой записи.

(Второй параметр существует по историческим причинам и не может быть опущен; передача пустой строки является традиционной.)

**history.pushState(data, "", url)**

Добавляет новую запись в историю сеансов с состоянием classic history API, установленным на сериализацию данных, и с URL-адресом, установленным на url.

Если URL-адрес текущего документа не может быть переписан на url-адрес, будет выдано исключение DOMException "SecurityError".

(Второй параметр существует по историческим причинам и не может быть опущен; передача пустой строки является традиционной.)

**history.replaceState(data, "")**

Обновляет классическое состояние API истории записи истории активного сеанса до структурированного клона данных.

(Второй параметр существует по историческим причинам и не может быть опущен; передача пустой строки является традиционной.)

**history.replaceState(data, "", url)**

Обновляет состояние classic history API записи истории активного сеанса до структурированного клона данных, а его URL-адрес - до url-адреса.

Если URL-адрес текущего документа не может быть переписан на url-адрес, будет выдано исключение DOMException "SecurityError".

(Второй параметр существует по историческим причинам и не может быть опущен; передача пустой строки является традиционной.)

У документа есть объект истории, объект History.

Шаги по получению истории заключаются в том, чтобы вернуть объект истории этого связанного документа.

Каждый объект истории имеет состояние, изначально нулевое.

Каждый объект истории имеет длину, неотрицательное целое число, изначально равную 0.

Каждый объект истории имеет индекс, неотрицательное целое число, изначально равное 0.

**Шаги получения length равны:**

Если связанный с этим релевантным глобальным объектом документ не полностью активен, то возникает исключение DOMException "SecurityError".

Возвращает эту длину.

**Шаги получения scrollRestoration равны**

Если связанный с этим релевантным глобальным объектом документ не полностью активен, то возникает исключение DOMException "SecurityError".

Верните режим восстановления прокрутки активной записи истории сеансов этого узла, доступного для навигации.

**Шаги настройки scrollRestoration следующие:**

Если связанный с этим релевантным глобальным объектом документ не полностью активен, то возникает исключение DOMException "SecurityError".

Установите режим восстановления прокрутки активной записи истории сеансов этого узла, доступного для навигации, на заданное значение.

**Шаги получения state следующие:**

Если связанный с этим релевантным глобальным объектом документ не полностью активен, то возникает исключение DOMException "SecurityError".

Возвращает это состояние.

**go(delta)** метод заключается в том, чтобы дельта пересекла эту заданную дельту.

**back()** метод заключается в том, чтобы дельта-пересечь это заданное значение -1.

**forward()** метод заключается в том, чтобы дельта пересекла это заданное значение +1.

Для дельта-обхода объекта истории history задается целочисленная дельта:

Пусть document будет документом, связанным с соответствующим глобальным объектом истории.

Если документ не полностью активен, то генерируется исключение DOMException "SecurityError".

Если дельта равна 0, то перезагрузите узел документа, доступный для навигации.

Просматривайте историю с помощью дельты, заданной для узла документа, доступного для навигации, и с параметром sourceDocument, установленным в document.

**pushState(data, unused, url)** метод заключается в выполнении шагов push/replace state для общей истории с учетом этого, данных, URL и "push".

**replaceState(data, unused, url)** метод заключается в том, чтобы выполнить шаги по отправке/замене состояния общей истории с учетом этого, данных, URL-адреса и "заменить".