

1. Разберите пример *Story.html*, в котором находится интерактивная история «Приключения студента». Принцип работы данного сценария: в каждой сцене присутствуют два варианта дальнейшего развития событий, каждый раз для перехода к следующей сцене пользователю приходится выбирать один из двух вариантов развития событий в текущей сцене. Содержимое веб-страницы меняется в зависимости от выбора пользователя.
2. Усовершенствуйте код страницы *Story.html* следующим образом:
 - Замените вложенные операторы *if..else* оператором *switch..case*;
 - Измените способ описания текущей сцены. Необходимо, чтобы описание сцены не появлялось в отдельном окне (*alert*), а находилось непосредственно на странице (*div*, *getElementById*, *innerHTML*). Для данной задачи подойдет функция *replaceNodeText(id, newText)*, из следующего пункта;
 - Замените кнопки перехода к следующей сцене на HTML-элементы, содержащие текст (*span*). Реализуйте изменение текста внутри кнопок при переходе к следующей сцене вашей истории (для этого напишите функцию *replaceNodeText(id, newText)*, которая изменяет текст в узле с указанным ID (*removeChild*, *appendChild*, *createTextNode*)). При наведении на кнопку указателя мыши стиль кнопки должен изменяться (*className*, *onmouseover*, *onmouseout*). Если из некоторых сцен возможен только один переход (например, первая сцена *scene0*) – то вторая кнопка должна быть скрыта (*visibility*, *hidden*, *visible*);
 - Создайте элемент *div* с *id="history"*, содержащий список сцен, посещённых пользователем по мере продвижения по сюжету (*p*). Каждая пройденная сцена должна добавляться к истории вместе с решением, которое к ней привело. Пример записи элемента истории: *decision 1 -> Сцена 2 : "You have arrived at a cute home"* (*createElement*, *createTextNode*, *appendChild*).
3. Создайте свою интерактивную историю (состоящую из 10 и более сцен), реализованную по данному алгоритму.
4. Оформите вашу историю с помощью фреймворка Bootstrap или своего набора CSS стилей.

Контрольные вопросы

1. Что такое модальное окно в Javascript?
2. Назовите отличия объявления переменных в Javascript через ключевые слова *let* и *var*.
3. Чем отличается оператор идентичности *===* от оператора эквивалентности *==* в Javascript?
4. Опишите принцип работы тернарного условного оператора.
5. Как ведёт себя функция в Javascript, если аргументов ей будет передано больше, чем у неё задано параметров? А если будет передано меньше?
6. Какой из двух типов циклов (*while* или *do...while*) всегда выполнится хотя бы один раз?
7. Опишите принцип работы инструкций *break* и *continue* в теле цикла.
8. Как будет работать цикл *for (var i = 0; i < 3; i++) { ... }*, если убрать одно из условий (объявление переменной, условие, счётчик)?