**Вопрос 1** **Чем объект Event отличается от стандартных объектов событий?**

Нет дополнительных полей и методов.

Некоторые поля можно изменять.

Содержит имя DOM-элемента.

Внутри обработчика имеет указатель this.

**Ответ:** Некоторые поля можно изменять

**Вопрос 2** **Свойство объекта Event, которое содержит тип (название) прошедшего события:**

type()

data()

result()

target()

**Ответ:** type()

**Вопрос 3** **Свойство which() объекта Event содержит:**

Код функции, которая выполнится после нажатия кнопки на клавиатуре.

Код кнопки, которая была зажата во время вызова данного события.

Код функции, которая выполнится, после того как кнопка на клавиатуре будет отпущена.

Код функции, которая выполнится после нажатия клавиши мыши на выбранном элементе.

**Ответ:** Код кнопки, которая была зажата во время вызова данного события

**Вопрос 4** **Что будет выведено в консоль, после компиляции данного кода?**

**var myP = document.getElementById("elem");**

**myP.addEventListener("click", function()**

**{**

**console.log(this.id);**

**});**

Значение текущего элемента.

Ошибка на этапе выполнения.

ID элемента, который вызвал обработчик событий.

ID текущего элемента.

**Ответ:** ID элемента, который вызвал обработчик событий

**Вопрос 5** **Как работает метод blur()?**

Привязывает функцию к выбранному элементу.

Привязывает или вызывает функцию, которая выполняется, когда выбранный элемент станет активным.

Привязывает или вызывает функцию, которая выполняется, когда выбранный элемент загрузится.

Привязывает или вызывает функцию, которая выполняется, когда выбранный элемент перестанет быть активным.

**Ответ:** Привязывает или вызывает функцию, которая выполняется, когда выбранный элемент перестанет быть активным

**Вопрос 6** **Что такое событие?**

Сигнал от браузера, когда кликнули левой кнопкой мыши по элементу.

Сигнал от браузера при нажатии кнопки на клавиатуре.

Сигнал от браузера, когда кликнули правой кнопкой мыши по элементу.

Сигнал от браузера, который происходит в момент какого-то действия на определенный объект.

**Ответ:** Сигнал от браузера, который происходит в момент какого-то действия на определенный объект

**Вопрос 7** **Что произойдет при нажатии правой кнопки мыши?**

**var zag = document.querySelector("p");**

**var button = document.querySelector("button"); zag.addEventListener("mousedown", function()**

**{**

**console.log("Обработчик заголовка.");**

**});**

**button.addEventListener("mousedown", function(event)**

**{**

**console.log("Обработчик кнопки.");**

**if (event.which == 3)**

**event.stopPropagation();**

**});**

Запуск обработчика кнопки.

Ошибка на этапе компиляции.

Запуск обработчика заголовка.

Запуск обоих обработчиков.

**Ответ:** Запуск обработчика кнопки

**Вопрос 8** **Объект Event – это:**

Обработчик, который назначен для удаляемого события.

Объект обрабатываемого события.

Объект, который содержит данные об обработчике событий.

Объект, который содержит данные о событии, которое уже произошло.

**Ответ:** Объект, который содержит данные о событии, которое уже произошло

**Вопрос 9** **Свойство объекта Event, которое содержит имя DOM-элемента:**

target()

data()

currentTarget()

attrName()

**Ответ:** currentTarget()

**Вопрос 10** **Выберите правильные варианты, создания события необходимо ввести:**

.on(events[,selector][,data])

.on(events[,selector][,data], handler)

.trigger(eventType, [extraParameters])

.bind(eventType, function)

**Ответ:** .on(events[,selector][,data])   
.on(events[,selector][,data], handler)

**Вопрос 11** **Выберите правильные варианты ответов.**

**Методами для удаления обработчиков событий являются:**

trigger()

die()

unbind()

toggle()

**Ответ:** die()   
unbind()

**Вопрос 12** **В чем заключается действие такого понятия, как «всплытие события»?**

Регистрация своего второго аргумента как функции, которая вызывается, когда описанное в первом аргументе событие случается.

Генерация события только для текущего элемента.

Генерация событий не только для текущего элемента, но и для всех его потомков, включая родителя.

События, зарегистрированные на узлах, которые в свою очередь имеют свои дочерние узлы, получают события своих детей.

**Ответ:** Генерация событий не только для текущего элемента, но и для всех его потомков, включая родителя

**Вопрос 13** **Метод для установки обработчика событий на один или несколько выбранных элементов страницы:**

.on()

.bind()

.trigger()

.target()

**Ответ:** .on()

**Вопрос 14** **С помощью метода off() можно:**

Удалить обработчики событий, которые были созданы с помощью метода one().

Удалить все обработчики заданного события.

Сразу удалить несколько разных обработчиков событий, которые были установлены на разные типы событий.

Удалить обработчики событий, которые были созданы с помощью метода bind().

**Ответ:** Сразу удалить несколько разных обработчиков событий, которые были установлены на разные типы событий

**Вопрос 15** **Выберите метод, который используется для остановки выполнения действий по умолчанию:**

unbind()

stopDefault()

triggerHandler()

preventDefault()

**Ответ:** preventDefault()

**Вопрос 16** **Какое действие выполняет метод unload()?**

Запускает обработчик события без его выполнения.

Устанавливает обработчик ошибки при загрузке элементов.

Устанавливает обработчик, с помощью которого происходит выход со страницы.

Устанавливает обработчик, с помощью которого происходит запуск этого события.

**Ответ:** Устанавливает обработчик, с помощью которого происходит выход со страницы.

**Вопрос 17** **Выберите правильные варианты использования метода mouseover():**

.mouseover()

.mouseover(eventData, handler(eventObject))

.mouseover(eventObject, handler(eventData))

.mouseover(handler)

**Ответ:** .mouseover()   
.mouseover(eventData, handler(eventObject))

**Вопрос 18** **Данная строка кода обозначает:**

**$("#foo").unbind(".myEvents");**

Отключение всех обработчиков указанного пространства имен, независимо от типа события.

Открепление события с указанным пространством имен.

Открепление конкретного события с указанным пространством имен.

Отключение обработчика внутри самого себя.

**Ответ:** Отключение всех обработчиков указанного пространства имен, независимо от типа события

**Вопрос 19** **Свойство result() объекта Event содержит:**

Значение, которое вернул предыдущий обработчик этого события.

Обработчика, назначенного ранее для события.

Время произведения события.

Значение заданного события.

**Ответ:** Значение, которое вернул предыдущий обработчик этого события

**Вопрос 20** **Недостаток метода bind():**

Если параметры не указаны, будут удалены все обработчики.

Событие вызывается только для первого элемента.

При добавлении новых элементов после назначения обработчика, для новых событий обработчики назначаться не будут.

Используется только для динамического назначения обработчиков событий.

**Ответ:** При добавлении новых элементов после назначения обработчика, для новых событий обработчики назначаться не будут

**Вопрос 21** **Метод jQuery.proxy() создает:**

По заданной функции другую функцию, которая будет иметь переменную this с заданным значением.

Объект, который содержит данные о текущем событии.

Обработчики событий для каждого из элементов, содержащихся в элементе jQuery.

Обработчик события для потомка выбранного элемента.

**Ответ:** По заданной функции другую функцию, которая будет иметь переменную this с заданным значением

**Вопрос 22** **Метод, с помощью которого можно установить обработчик отправки формы:**

submit()

select()

load()

toggle()

**Ответ:** submit()

**Вопрос 23** **Методом для установки обработчика выделения текста является:**

select()

blur()

mousemove()

change()

**Ответ:** select()

**Вопрос 24** **Когда используется метод .one()?**

При вызове указанного события у элемента.

При удалении обработчика событий.

При установке обработчика событий на один или несколько выбранных элементов страницы.

При установке обработчика событий, который сработает лишь один раз.

**Ответ:** При установке обработчика событий, который сработает лишь один раз

**Вопрос 25** **Для чего нужен метод mousedown()?**

**(выберите несколько ответов).**

Для установки обработчика нажатия кнопки мыши.

Для установки обработчика клика мышью по элементу.

Для запуска события.

Для установки обработчика выхода курсора из области элемента.

**Ответ:** Для установки обработчика нажатия кнопки мыши   
Для запуска события

**Вопрос 26** **Обратиться к свойствам и методам объекта Event можно, с помощью:**

Имени объекта через оператор ->.

Объекта, который был передан параметром в обработчик события.

Обращаться к свойствам объекта Event нельзя.

Указателя this.

**Ответ:** Объекта, который был передан параметром в обработчик события

**Вопрос 27** **«Всплытие события» можно прервать, с помощью:**

Метода stopPropagation().

Метода off().

Метода preventDefault().

Метода click().

**Ответ:** Метода stopPropagation()

**Вопрос 28** **Когда запускается событие “scroll”?**

Когда страница закрывается.

При получении элементом фокуса.

Когда элемент прокручивается.

Когда заканчивается загрузка страницы.

**Ответ:** Когда элемент прокручивается

**Вопрос 29** **После создания обработчика событий вызывается метод bind(), который:**

Удаляет обработчики событий для всех элементов.

Делает привязку обработчика события к определенному событию для выбранного элемента.

Устанавливает обработчик события на выбранный элемент страницы.

Позволяет получить доступ к элементу.

**Ответ:** Делает привязку обработчика события к определенному событию для выбранного элемента