

1. Сколько есть видов условных операторов?

Ответ: Бывают нескольких видов:

- условный оператор **if** (с одной ветвью)
- условный оператор **if...else** (с двумя ветвями)
- условный оператор **else if...** (с несколькими ветвями)
- тернарный оператор (**?:**)
- оператор выбора **switch**

2. Перепишите пример про определение времени суток через тернарный оператор

```
let date = new Date();
```

```
let time = date.getHours();
```

```
if (time < 10) {  
  alert( "Доброе утро!" );  
} else {  
  alert( "Добрый день!" );  
}
```

Ответ:

3. Выведется ли **alert**?

```
if ("0") {  
  alert( 'Привет' );  
}
```

Ответ: да.

4. Чему будет равно условие (правда или ложь) в этих случаях, если **x = 6** and **y = 3** ?

**Ответы**

- $(x < 10 \ \&\& \ y > 1)$  логическое **И** *true*
- $(x == 5 \ || \ y == 5)$  логическое **ИЛИ** *false*
- $!(x == y)$  логическое **НЕ** *true*

5. Назовите три способа назначения обработчиков событий. Какой из них самый универсальный?

Ответ:

- Использование атрибута HTML: Обработчик может быть назначен прямо в разметке, в атрибуте, который называется **on<событие>**.
- Использование свойства DOM-объекта. Можно назначать обработчик, используя свойство DOM-элемента **on<событие>**.
- Доступ к элементу через **this**. Внутри обработчика события **this** ссылается на текущий элемент, то есть на тот, на котором, как говорят, «висит» (т.е. назначен) обработчик.

6. Корректна ли такая запись? **button.onclick = hello();**

Ответ: нет

7. Какие есть события у клавиатуры?

Ответ:

**keydown** и **keyup** – когда пользователь нажимает / отпускает клавишу

8. Что такое объект события и какие у него могут быть свойства?

Ответ:

Чтобы хорошо обработать событие, могут понадобиться детали того, что произошло. Не просто «клик» или «нажатие клавиши», а также – какие координаты указателя мыши, какая клавиша нажата и так далее.

Когда происходит событие, браузер создаёт *объект события*, записывает в него детали и передаёт его в качестве аргумента функции-обработчику.

Получить детальную информацию о событии в обработчике можно посредством **объекта события** (Event). Данный объект создаёт браузер, когда это событие происходит. В него он помещает много различной информации. Свойства объекта **Event**:

- `bubbles` – логическое значение, указывающее на то является ли данное событие всплывающим;
- `cancelable` – определяет можно ли событие отменить;
- `cancelBubble` – при установке `true` предотвращает всплытие события, т.е. оно всплывать не будет (является псевдонимом метода `stopPropagation`);
- `composed` – указывает может ли событие всплывать через из теневого DOM (внутреннего DOM конкретного элемента) в обычный DOM документа;
- `currentTarget` – элемент, привязанный к обработчику события;
- `defaultPrevented` – показывает был ли для события вызван метод `preventDefault`;
- `eventPhase` – число, указывающее фазу процесса распространения события (0 – не обрабатывается, 1 – погружение, 2 – целевой элемент, 3 – всплытие);
- `isTrusted` – указывает вызвано ли событие действием пользователя или программно (посредством использования метода `dispatchEvent`);
- `returnValue` – альтернатива для `preventDefault`;
- `target` – элемент, который создал событие;
- `timestamp` – время, когда произошло событие;
- `type` – тип (имя) события.

9. Самостоятельно разберитесь, какие бывают операторы сравнения? Напишите сюда как выглядят сравнение "равно", "не равно", "больше чем".

Ответ:

- Больше/меньше:  $a > b$ ,  $a < b$ .
- Больше/меньше или равно:  $a \geq b$ ,  $a \leq b$ .
- Равно:  $a == b$ . Обратите внимание, для сравнения используется двойной знак равенства `==`. Один знак равенства  $a = b$  означал бы присваивание.
- Не равно. В математике обозначается символом  $\neq$ , но в JavaScript записывается как  $a \neq b$ .