**Вопросы 💎**

1. Сколько есть видов условных операторов?

There are 5 types of conditional operators:

- conditional operator \*\*if\*\* (with one branch)

- conditional operator \*\*if...else\*\* (two branches)

- conditional operator \*\*else if...\*\* (with several branches)

- ternary operator (\*\*?:\*\*)

- selection operator \*\*switch\*\*

1. Перепишите пример про определение времени суток через тернарный оператор

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

let date = new Date();

let time = date.getHours();

function checkTime (time) =

(time < 10) ? alert( "Доброе утро!" ) : alert( "Добрый день!" );

1. Выведется ли alert?

if ("0") {

alert( 'Привет' );

}

Yes, it will. Any string other than an empty string (and string "0" is not empty) becomes true in the logical context.

1. Чему будет равно условие (правда или ложь) в этих случаях, если x = 6 and y = 3?
   * (x < 10 && y > 1) *логическое И* **true**
   * (x == 5 || y == 5) *логическое ИЛИ* **false**
   * !(x == y) *логическое НЕ* **false**
2. Назовите три способа назначения обработчиков событий. Какой из них самый универсальный?
3. The handler can be assigned directly in the markup, in an attribute called `on<event>`.

For example, to assign a `click` event handler to the `input` element, the `onclick` attribute can be used. <input value="Нажми меня" *onclick="alert('Клик!')"* type="button">

When you click on the button, the code specified in the onclick attribute will be executed.

1. You can assign a handler using the `on<event>` property of the DOM element.

For example, `elem.onclick`:

<input id="elem" type="button" value="Click me!">

elem.onclick = function() {

alert('Thank you');

};

1. Inside the event handler, `this` refers to the current element, i.e. the one the handler is said to be `hanging' (i.e. assigned to).

In the code below, `button` outputs its content using `this.innerHTML`:

<button onclick="alert(this.innerHTML)">Click me</button>

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

Not, it’s not correct, because we don’t need brackets here.

1. Какие есть события у клавиатуры?

keydown and keyup - when the user presses / releases a key

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

To process an event well, you may need details of what happened. Not just a 'click' or a 'keystroke', but which coordinates has the mouse pointer, which key was pressed and so on.

When an event occurs, the browser creates an \*event object\*, writes the details into it and passes it as an argument to a handler function.

Some properties of the `event` object:

\*\*event.type\*\* Event type, in this case ``click``.

\*\*event.currentTarget\*\* Element on which the handler was triggered. The value is usually the same as `this`, but if the handler is an arrow function or with `bind` tied to another object as `this`, we can get the item from `event.currentTarget`.

\*\*event.clientX / event.clientY\*\*Cursor coordinates at the time of click relative to the window, for mouse events.

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

Comparison statements allow you to compare two values with each other and return true if the condition is met, and false if not.

Comparison statements always return logical values, which are commonly used in **if, while**, or **for** instructions to control program execution.

A complete list of comparison statements in JavaScript is given in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Оператор** | **Описание** | **Пример** | **Результат** |
| == | Равно | 10 == 20 | false |
| === | Строго равно (идентичность) | "20" === 20 | false |
| != | Не равно | 10 != 20 | true |
| !== | Строго не равно | 20! == 20 | false |
| > | Больше чем | 20 > 10 | true |
| >= | Больше или равно | 20 >= 10 | true |
| < | Меньше чем | 20 < 10 | false |
| <= | Меньше или равно | 20 <= 10 | false |

"равно" ==

"не равно" !=

"больше чем" >