

# FRONT-END PRO

## LESSON 2

---

# ШАБЛОННЫЕ СТРОКИ

Заключены в обратные кавычки (``` ```) вместо двойных или одинарных. Можно использовать выражения внутри `${variable}` и писать в несколько строк.

```
`строка текста`
```

```
`строка текста 1  
строка текста 2`
```

```
`строка текста ${выражение} строка текста`
```

```
tag `строка текста ${выражение} строка текста`
```

---

## DOCUMENT.WRITE()

Метод, который пишет текст прямо в HTML.

В современных скриптах он редко встречается из-за следующего важного ограничения:

Вызов `document.write` работает только во время загрузки страницы.

Если вызвать его позже, то существующее содержимое документа затрётся.

---

## ALERT, PROMPT, CONFIRM

`alert(str)` - показывает сообщение.

`prompt(str)` - показывает сообщение и запрашивает ввод текста от пользователя. Возвращает напечатанный в поле ввода текст или `null`, если была нажата кнопка «Отмена» или Esc с клавиатуры.

`confirm(str)` - показывает сообщение и ждёт, пока пользователь нажмёт ОК или Отмена. Возвращает `true`, если нажата ОК, и `false`, если нажата кнопка «Отмена» или Esc с клавиатуры.

---

# МЕТОДЫ СТРОКИ

`trim()` - метод удаляет пробельные символы с начала и конца строки.

`replace(str, newValue)` / `replaceAll(str, newValue)` - методы поиска и замены.  
Возвращают новую строку с заменой всех вхождений.

`toLowerCase()` / `toUpperCase()` - перевести строку в нижний или верхний регистр.

`slice(start, end)` / `substring(start, end)` - получения подстроки.

`indexOf(str, start)` / `includes(str, start)` / `startsWith(str)` / `endsWith(str)` - поиск подстроки, когда надо только проверить, есть ли вхождение.

---

## ТИП ДАННЫХ BOOLEAN

Булевый тип может принимать только два значения: `true` (истина) и `false` (ложь).

Такой тип, как правило, используется для хранения значений да/нет: `true` значит «да, правильно», а `false` значит «нет, не правильно».

# ОПЕРАТОРЫ СРАВНЕНИЯ

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

# IF/ELSE

Инструкция `if(...)` вычисляет условие в скобках и, если результат `true`, то выполняет блок кода.

Инструкция `if(...)` может содержать необязательный блок «`else`» («иначе»). Он выполняется, когда условие ложно.

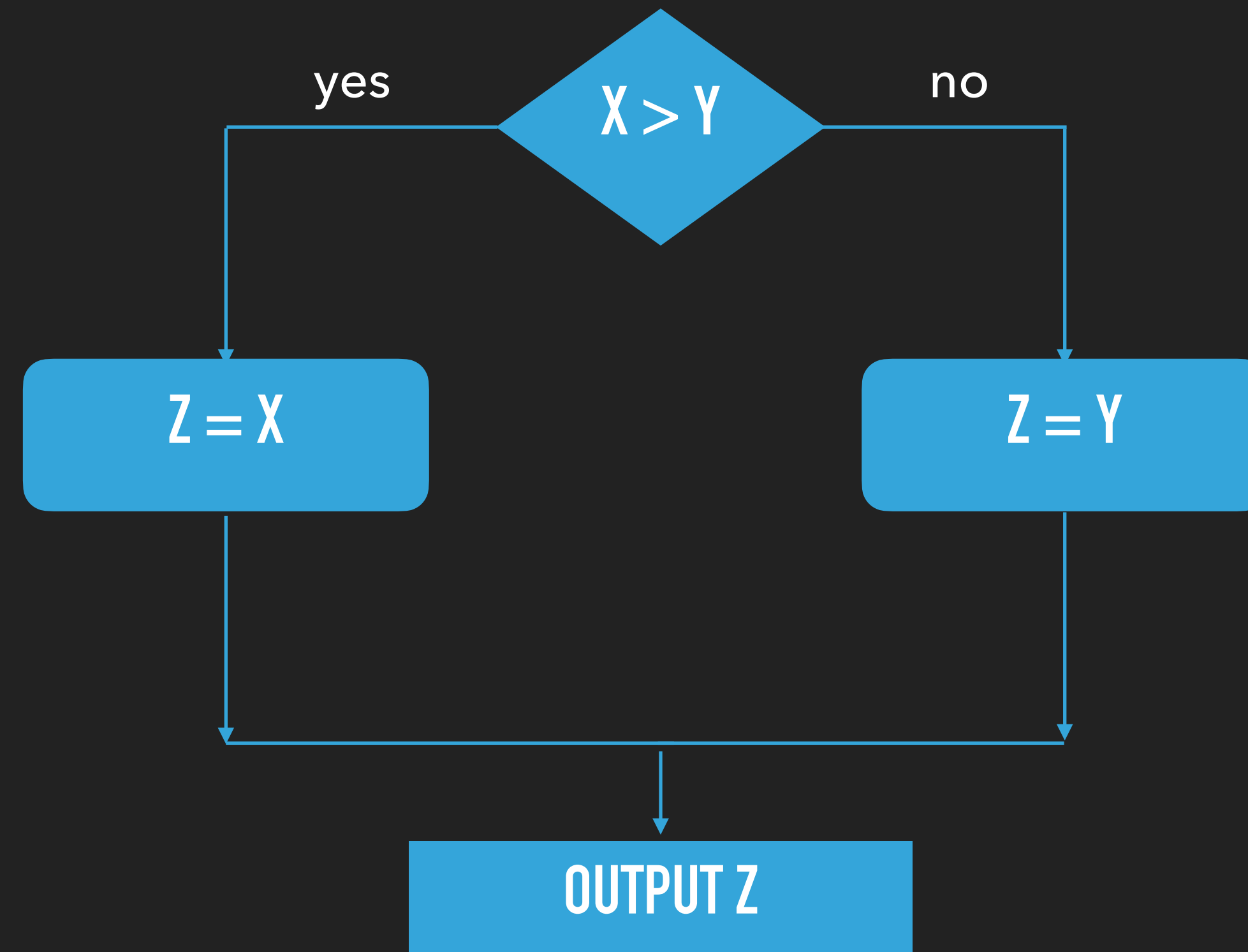
Если нужно проверить несколько вариантов условий. Для этого используется блок `else if`

```
let year = prompt('В каком году была опубликована спецификация ECMAScript-2015?', '');

if (year < 2015) {
  alert( 'Это слишком рано...' );
} else if (year > 2015) {
  alert( 'Это поздновато' );
} else {
  alert( 'Верно!' );
}
```



# IF/ELSE



# МЕТОДЫ ЧИСЕЛ

`Math.floor(num)` - Округление в меньшую сторону: 3.1 становится 3, а -1.1 – -2.

`Math.ceil(num)` - Округление в большую сторону: 3.1 становится 4, а -1.1 – -1.

`Math.round(num)` - Округление до ближайшего целого: 3.1 становится 3, 3.6 – 4, а -1.1 – -1.

`parseFloat(str)` - преобразования строки в число, по порядку считывают число из строки до тех пор пока не возникнет ошибка. Возвращает число с плавающей точкой.

`toFixed(n)` - округляет число до `n` знаков после запятой и возвращает строковое представление результата.

```
let num = 12.34;  
num.toFixed(1); // "12.3"
```