МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка

JavaScript
Работа со строками

МШП

Цели урока:

1. Потренируемся работать со строками
2. Напишем свой собственный чат

МШП

Разбор ДЗ

Что получилось?

Давайте посмотрим :)

МШП

Теория

Метод и свойство

МШП

К этому моменту вы знаете:

1. Как изменять содержимое страницы
2. Как управлять стилем через JS
3. Как работать с событием клик

Настало время поговорить правильных терминах!

МШП

Значение

Значение – это строка, число или элемент на странице.

Существуют другие типы значений, вы познакомитесь с ними позже.

- `Злой гном`
- 45
- node

МШП

Переменная

Значения записывают в **переменные,**
чтобы у них появилось **название**
и их можно было использовать много раз.

```
let string = `Злой гном`;

let number = 45;

let node = document.querySelector(`#node`);
```

МШП

Свойство

Свойство – это **значение**, которое принадлежит другому значению.

Иногда это значение можно изменить, а иногда нет.

- `node.innerHTML`
- `input.value`
- `node.classList`

МШП

str.length

У строк есть специальное свойство, в котором хранится **число** – **длина строки**.

Это число показывает, сколько символов сейчас в строке.

```
let str = `Привет!`;

console.log(str.length);

// 7
```

МШП

str.length

Длина строки очень полезное свойство, с помощью него можно проверять значения в поле ввода.

```
let str = input.value;
if (str.length == 0) {
    console.log(`Пустая строка!`);
}
```

МШП

Метод

Метод – это **действие**, которое можно выполнить.

Иногда это действие возвращает **результат**.

```
let node = document.querySelector(`#node`);

let number = Number(input.value);

node.classList.add(`green`);
```

МШП

Метод

У методов есть **параметры** –
это значения, которые
необходимы для работы их работы.

```
let node = document.querySelector(`#node`);

let number = Number(input.value);

node.classList.add(`green`);
```

МШП

Теория

Методы строк

МШП

toUpperCase()

```
let str1 = `шКоЛЬная ПиЦца`;

let str2 = str1.toUpperCase();
// `ШКОЛЬНАЯ ПИЦЦА`
```

Метод toUpperCase() делает из строки новую строку с заглавными буквами.

МШП

toLowerCase()

```
let str1 = `шКоЛЬная ПиЦца`;

let str2 = str1.toLowerCase();
// `школьная пицца`
```

Метод toLowerCase() делает из строки новую строку со строчными буквами.

МШП

toLowerCase()

```
let str = `шКоЛЬная ПиЦца`;

str = str.toLowerCase();
// `школьная пицца`
```

Не обязательно объявлять новые переменные! Можно сохранять **новое значение в старую переменную**.

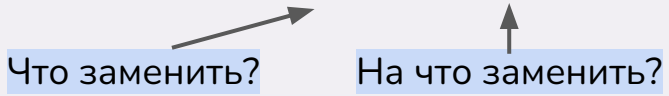
МШП

replaceAll()

```
let str = `Как так!!!!!!`;

str = str.replaceAll(`!`, `?`);

// `Как так??????`
```



Метод replaceAll заменяет
одну подстроку на другую подстроку

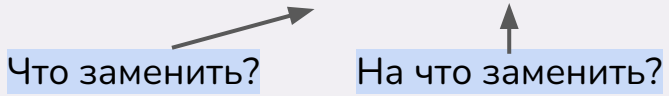
МШП

replaceAll()

```
let str = `кот кот кот`;

str = str.replaceAll(`кот`, `пёс`);

// `пёс пёс пёс`
```



Метод replaceAll заменяет
одну подстроку на другую подстроку

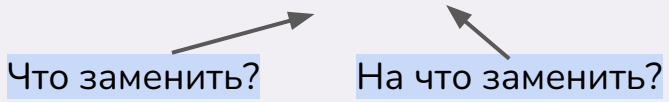
МШП

replaceAll()

```
let str = `7-999-999-99-99`;

str = str.replaceAll(`-`, ``);

// `79999999999`
```



С помощью replaceAll можно даже удалить "лишние" символы

МШП

Тест

Методы строк

МШП

Практика

Чат

МШП

Теория

Составные условия

МШП

Представим типичную и очень жизненную ситуацию: нам нужно проверить пароль на выполнение нескольких условий.

МШП

Проверка пароля

```
if (password.length < 6) {
    // Короткий
} else if (password.length > 50) {
    // Длинный
} else if (password == '123456') {
    // Лёгкий
} else {
    // ОК
}
```

Чем больше условий, тем длиннее и страшнее будет код :(

МШП

С помощью логических операторов можно объединять условия!

Знакомьтесь:

1. **&&** – И
2. **||** – ИЛИ

МШП

|| – ИЛИ

```
if (
    password.length < 0 ||
    password.length > 50 ||
    password == '123456'
) {
   // Ошибка
} else {
   // OK
}
```

`книга` подойдёт?

Условие выполняется, если хотя бы одно из условий ИЛИ выполняется.

МШП

Давайте перепишем эту программу так, чтобы она работала через условие И

МШП

&& – И

```
if (
    password.length > 6 &&
    password.length < 50 &&
    password != '123456'
) {
   // OK
} else {
   // Ошибка
}
```

`пирожок` подойдёт?

`1234` подойдёт?

Условие выполняется, если выполняются все условия И.

МШП

Для супер сложности пароля в нём обязательно должны быть специальные символы.

Как проверить, есть ли они?

МШП

str.includes()

Метод **includes** проверяет, есть ли в строке символ или даже целое слово.

```
let answer = password.includes(`?`);

answer = true, если ответ ДА
answer = false, если ответ НЕТ
```

МШП

str.includes()

```
if ( password.includes(`?`) ) {
    // OK
}
```

Этот метод можно использовать сразу внутри условий.

Если **includes** ответит ДА, то условие выполнится.

МШП

str.includes()

```
if (
    password.includes(`?`) ||
    password.includes(`!`) ||
    password.includes(`*`)
) {
    // OK
}
```

С помощью **includes** можно устроить целую серию проверок :)

Дальше всё зависит от твоей любви к копипасту!

МШП

! – НЕ

```
// если `a` нет...
if ( !(password.includes(`a`)) ) {
    // OK
}
```

Логический оператор НЕ позволяет проверить, что какого-то символа в строке **нет**.

МШП

Практика

Составные условия

МШП

Практика

Анимация

Animate.css

Just-add-water CSS animations

МШП

Цели урока:

- ✓ Потренируемся работать со строками
- ✓ Напишем свой собственный чат

МШП

Домашнее задание