JavaScript

Урок №6: Строки

Автор Бахшиллоев Бехзод
Ментор по Frontend в PRO UNITY

Строки

В JavaScript любые текстовые данные являются строками. Не существует отдельного типа «символ», который есть в ряде других языков.

Внутренний формат для строк — всегда UTF-16, вне зависимости от кодировки страницы.

В JavaScript есть разные типы кавычек.

Строку можно создать с помощью одинарных, двойных либо обратных кавычек:

```
let single = 'single-quoted';

let double = "double-quoted";

let backticks = `backticks`;
```

Одинарные и двойные кавычки работают, по сути, одинаково, а если использовать обратные кавычки, то в такую строку мы сможем вставлять произвольные выражения, обернув их в ${...}:

```
alert(`1 + 2 = ${1 + 2}.`); // 1 + 2 = 3.
```

Backticks

Ещё одно преимущество обратных кавычек — они могут занимать более одной строки, вот так:

```
let guestList = `Guests:
 * John
 * Pete
 * Mary
`;
alert(guestList); // список гостей, состоящий из нескольких строк
```

Спецсимволы

Многострочные строки также можно создавать с помощью одинарных и двойных кавычек, используя так называемый «символ перевода строки», который записывается как \n:

```
let guestList = "Guests:\n * John\n * Pete\n * Mary";

alert(guestList); // список гостей, состоящий из нескольких строк
```

В частности, эти две строки эквивалентны, просто записаны по-разному:

```
// перевод строки добавлен с помощью символа перевода строки

let str1 = "Hello\nWorld";

// многострочная строка, созданная с использованием обратных кавычек

let str2 = `Hello

World`;

alert(str1 == str2); // true
```

Список спецсимволов

Есть и другие, реже используемые спецсимволы. Вот список:

Символ	Описание
`\n`	Перевод строки
`\r`	В текстовых файлах Windows для перевода строки используется комбинация символов `\r\n`, а на других ОС это просто `\n`. Это так по историческим причинам, ПО под Windows обычно понимает и просто `\n`.
`\'`, `\"`, `\``	Кавычки
`\\`	Обратный слеш
`\t`	Знак табуляции
`\b`, `\f`, `\v`	Backspace, Form Feed и Vertical Tab — оставлены для обратной совместимости, сейчас не используются.

Как вы можете видеть, все спецсимволы начинаются с обратного слеша, \ — так называемого «символа экранирования».

Он также используется, если необходимо вставить в строку кавычку. К примеру:

```
alert( 'I\'m the Walrus!' ); // I'm the Walrus!
```

Строки. Методы.

```
let str = “Hello world”;

str.length;
str.charAt(1); (устаревший)
str[1];
str.toLowerCase();
str.toUpperCase();
str.indexOf(‘is’); // lastIndexOf
str.substr(5 [, 2])
str.substring(5 [, 10]);
str.slice(5 [, 10]);
str.includes(substring);
str + ‘another string’
```

Полезные ссылки: Методы строк

Строки. Шаблонные строки.

```
let name = “Denis”;
let age = 29;
let str = `Hello! my name is ${name} I’m ${age} years old`; // Hello! my name is Denis I’m 29 years old
```

Полезные ссылки: Шаблонные строки ,

Задачи:

let string = ‘some test string’;
ВРУЧНУЮ НИЧЕГО НЕ СЧИТАТЬ

1. Получить первую и последнюю буквы строки
2. Сделать первую и последнюю буквы в верхнем регистре
3. Найти положение слова ‘string’ в строке
4. Найти положение второго пробела (“вручную” ничего не считать)
5. Получить строку с 5-го символа длиной 4 буквы
6. Получить строку с 5-го по 9-й символы
7. Получить новую строку из исходной путем удаления последних 6-и символов (то есть исходная строка без последних 6и символов)
8. Из двух переменных a=20 и b=16 получить переменную string, в которой будет содержаться текст “2016”