



# Основы Программирования

Занятие №7  
Строки, Даты



## IT Education Academy

[WWW.ITEA.UA](http://WWW.ITEA.UA)

# План занятия

- Строки
- Даты

# Повторение материала

---

# Строка

---

Тип данных - String

Для создания значения используются одинарные или двойные кавычки. Если для создания строки вначале использовалась одинарная открывающаяся кавычка, то окончит строку первая на этой линии в коде соответствующая кавычка:

```
'This is string with " symbol inside'
"This is string with ' symbol inside"
'This is string ' and this is no more '
"This is stirng " and this is no more "
```

# Строка

---

Строка может быть описана только на одной линии в коде:

```
var bad = "You cannot do  
it like this";
```

```
var good = "You can do " +  
"it like this";
```

```
console.log(good); // You can do it like this
```

Для переноса текста на следующую строку при выводе на экран используется специальный символ.

# Специальные символы

---

Перенос на другую строчку, как и другие специальный символы начинается со \ за которым следует соответствующий символ.

Перенос	\n
Табуляция	\t
Кавычка ‘	\’
Кавычка “	\”
Слеш \	\\

Когда вы используете специальный символ в строке, интерпретатор по-особому обрабатывает его.

```
var txt = 'Hi\n Vlad \'this\' is inside quotes \\  
console.log(txt) \\Hi  
                \\ Vlad 'this' is inside quotes \
```

# Специальные символы

---

Вывести на экран табличку умножения

1 2 3 4 5 ...

2 4 6 8 10 ...

3 6 9 12 15 ...

...

Используя всего один `console.log`

# Длина строки

---

У строки, также как и у массива есть свойство `length`.

```
var string = 'Hello';  
var len = string.length;  
console.log(len); //5
```



# Обращение к символу

---

У строк, точно так же как и у массивов можно обратиться к отдельному элементу. Синтаксис такой же.

```
var string = 'Hello';
```

```
var h = string[0];  
var e = string[1];  
var l = string[2];  
var o = string[4];  
console.log(h); // h  
console.log(e); // e  
console.log(l); // l  
console.log(o); // o
```

Однако у строк нельзя переписать значение отдельного элемента. Строка неизменяема. Поменять можно значение переменной, но не строки.

# Неизменяемость

---

```
var string = 'hello';
```

```
string[0] = 'H'; // bad
```

```
string = 'H' + string[1] + string[2] + string[3] + string[4]; // better, but not the best
```

```
console.log(string);
```

# Строка

---

Написать функцию, которая принимает строку и выводит на экран каждый символ этой строки на новой строке.

Например:

```
function print(...) {...}
```

```
print('hello'); //h
                //e
                //l
                //l
                //o
```

# Функции над строками

---

<code>.indexOf(searchValue, startIndex = 0)</code>	Поиск позиции в которой находится <code>searchValue</code> , начиная со <code>startIndex</code>
<code>.replace(substring, newValue)</code>	Возвращает НОВУЮ строку в которой заменены все <code>substring</code> на <code>newValue</code> .
<code>.slice(begin, end = length)</code> <code>.substring(begin, end = length)</code>	Возвращает НОВУЮ подстроку, которая включает в себя только символы от <code>begin</code> до <code>end</code>
<code>.split(separator)</code>	Возвращает массив строк, который получен путем разделения исходной строки символами <code>separator</code>
<code>.toLowerCase()</code>	Возвращает НОВУЮ строку в которой все символы с маленькой буквы
<code>.toUpperCase()</code>	<code>==</code> с большой буквы
<code>.match(regex)</code>	Возвращает массив подстрок которые удовлетворяют регулярное выражение

# Регулярное выражение

---

Это специальный язык, который используется при работе со строками и позволяет находить определенные шаблоны в строках.

Например:

`^[2-9]\d{2}-\d{3}-\d{4}$`

Позволяет определять, является ли строка телефонным номером в формате США.

`^` - Начало строки

`[2-9]` - один любой символ в диапазоне от 2 до 9

`\d{2}` - две цифры

`-` - просто дефис

`\d{3}` - три цифры

`$` - конец строки

500-222-3333 - подходит

# Пример

---

```
var text = "Hello my name is Vlad. His name is Andrew.";
text.indexOf('is'); //14
text.indexOf('is', 15); //25
var another = text.replace('Vlad', 'John');
console.log(text); // Hello my name is Vlad. His name is Andrew.
console.log(another); // Hello my name is John. His name is Andrew.
var sub = text.substring(0, 5); //sub = 'Hello', text - the same
var words = text.split(' ');
//words = ['Hello', 'my', 'name', 'is', 'Vlad.', 'His', 'name', 'is',
'Andrew.']
var low = text.toLowerCase(); //low = 'hello my name is vlad. his name
is andrew'
var up = text.toUpperCase(); //up = 'HELLO MY NAME IS VLAD. HIS NAME IS
ANDREW'
```

# Практика

---

Написать функцию, которая принимает строку и возвращает новую строку, в которой первый символ с большой буквы.

```
function capitalize(string) {...}  
  
console.log(capitalize('hello')); //Hello
```

# Replace

---

Эта функция, помимо просто строки также принимает регулярное выражение.

С ее помощью можно удалить какие-то паттерны в тексте.

Например, регулярное выражение `/[\.,:;!?'"]/g` находит знаки препинания в тексте. А функция:

```
function removePunctuation(text) {  
    return text.replace(/[\.,:;!?'"]/g, '');  
}
```

Удалит их из строки.



# Практика

---

Написать набор функций, которые позволят найти все уникальные слова в тексте.

1. Написать функцию `getUnique(array)` которая принимает массив и возвращает новый массив, в котором только уникальные элементы. Подсказка: использовать еще одну переменную с пустым массивом, а также функцию массива `includes(elem)`.
2. Написать функцию удаления символов пунктуации со строки.
3. Написать функцию которая делает все слова в тексте с маленькой буквы и разбивает его на слова.
4. Скомпоновать все вместе, чтобы найти уникальные слова.
5. Проверить работу на тексте: "Hello, how are you? I am fine, and you? I am learning programming."

# Дата

---

В JavaScript нет особого типа данных для Date, поэтому если использовать оператор `typeof` она скажет что тип - `Object`.

Для создания нового значения даты используется синтаксис:

```
var now = new Date();  
console.log(now); //2018-11-03T21:20:00.000Z
```

Таким образом создается новый объект (новое значение) типа `Date`. Более подробно данный синтаксис будет рассмотрен позднее.

# Функции типа Date

---

У переменной, хранящей тип Date можно запросить следующие данные:

getDate()	День месяца (например 11)
getDay()	День недели (0, 1, 2, 3, 4, 5, 6), где 0 - Воскресенье, 1 - Понедельник и тд.
getFullYear()	Год
getHours()	Час
...	

# Функции типа Date

---

Создать массив days = ['Вскр', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб']

Создавать новую дату и выводить на экран какой сегодня день недели.

# 1970 год

---

В компьютере дата храниться в виде числа - количество миллисекунд, прошедших от 1 января 1970 года 00:00:00.000.

Дату можно также создать следующим образом:

```
var date = new Date(10000);
```

т.е. указав сколько миллисекунд прошло от 1970 года.



# 1970 год

---

Узнать какой день был с момента 1 миллиард миллисекунд от 1970 года  
100 миллиардов. 1 триллион.

# Еще один способ создания даты

---

Создать дату можно также:

```
var year = 2000;
var month = 1; //0 == Январь, 1 == Февраль и тд
var day = 2;
var hours = 10;
var minutes = 30;
var seconds = 40;
var date = new Date(year, month, day, hours, minutes, seconds)
```

# GetTime

---

У даты есть функция `getTime()` которая возвращает сколько миллисекнд прошло до указанной даты.

```
var now = new Date();  
console.log(now.getTime()); //1538603656706
```





# GetTime

---

Узнать сколько миллисекунд прошло с 1970 года до рождения:

Ди Каприо (November 11, 1974),

Майли Сайрус (November 23, 1992)

Тома Харди (September 15, 1977)

# Операции

---

Над датой также можно выполнять некоторые арифметические операции.

Запустить node.js

```
var a = new Date();
```

```
//подождать пару секунд
```

```
var b = new Date();
```

```
console.log(b - a); //сколько времени вы подождали
```

```
console.log(b > a); // true
```

# Summary

---

1. Строки неизменяемы.
2. Существуют специальные символы.
3. Строки похожи на массивы (length, доступ к элементу).
4. Существует специальный язык (Regex).
5. Дата считается от 1 января 1970 года.

# Домашнее задание

---

1. Попросить пользователя ввести любой текст. Сказать ему сколько слов было в тексте.
2. Написать функцию, которая проверяет на спам сообщение. `function checkSpam(message)`. Сообщение считается спамом, если в нем присутствует слово 'купить' либо 'выгодное предложение'. Проверить работу на сообщении 'КуПитЕ у нас премиум аккаунт' и 'выгодное: предложение только сейчас'.
3. Написать функцию, которая просит пользователя ввести дату его рождения в формате год месяц день (1991 август 24) и говорит ему сколько миллисекунд прошло с 1970 года.
4. Написать функцию, которая усекает строку если она слишком длинная (больше 10 символов). Например: `trunc('hello, how are you? I am fine')` возвращает 'hello, how...'.

**Спасибо за внимание!**