Лабораторная работа № 15. Новый ECMAScript 2018

Цель: изучить новшества ECMAScript 2018: флаг регулярных выражений dotAll, Захват именованных групп в регулярных выражениях, операторы rest и spread.

Теория

Флаг регулярных выражений dotAll

Сейчас, при использовании регулярных выражений, хотя считается, что символ точки соответствует любому одиночному символу, он не соответствует символам перевода строки вроде *\n \r \f* и так далее.

Например, до этого нововведения всё выглядело так:

```
  /first.second/.test('first\nsecond'); //false
```

Благодаря данному улучшению точка теперь соответствует абсолютно любому символу. Для того, чтобы старые регулярные выражения продолжали бы работать так, как раньше, при создании регулярных выражений, следующих новым правилам, нужно использовать флаг *\s*.

```
//ECMAScript 2018
  /first.second/s.test('first\nsecond');
//true - обратите внимание на /s
```

Захват именованных групп в регулярных выражениях

Это улучшение представляет собой полезную возможность регулярных выражений, которая имеется в других языках вроде Python и Java. Речь идёт об именованных группах. Эта возможность позволяет разработчикам писать регулярные выражения с назначением имён (идентификаторов) в формате (?<name>...) для групп. Это имя облегчает работу с группами.

В следующем примере мы используем имена (?<year>), (?<month>) и (?day) для группировки различных частей даты, с которой мы работаем с использованием регулярного выражения. Итоговый объект при таком подходе будет содержать свойство groups, которое будет иметь свойства year, month и day с соответствующими значениями.

```
//До ECMAScript 2018
  let re1 = /(\d{4})-(\d{2})-(\d{2})/;
  let result1 = re1.exec('2015-01-02');
  console.log(result1);
//После
  let re2 = /(?<year>\d{4})-(?<month>\d{2})-(?<day>\d{2})/u;
  let result2 = re2.exec('2015-01-02');
  console.log(result2);
```

Работа со свойствами объектов с использованием оператора rest

Оператор **rest** позволяет извлекать свойства объекта, которые пока из него не извлечены. Оператор представляется как троеточие.

```
  let { firstName, age, …remaining } = {
    firstName: 'Kate',
    lastName: 'Kovaleva',
    age:18,
    height:'5.10',
    race: 'martian',
  };
  firstName;         // Kate
  age;              // 18
  remaining;       // {lastName: 'Kovaleva', height:'5.10', race:
'martian'}
```

Работа со свойствами объектов с использованием оператора spread

Оператор **spread** также представлен как троеточие. Разница между ним и оператором *rest* заключается в том, что он используется для создания новых объектов.

Оператор *spread* используется в правой части выражения со знаком присваивания. Оператор *rest* используется в левой части выражения.

```
  const Person = {fname:'Kate', age:18};
  cosnt account = {name:'Ban', amount:'$500'};
  const personaccount = {…person,…account};
  personaccount;  //  {fname:  'Kate',  age:18,  name:  'Ban',
//amount:'$500'};
```

Задания к лабораторной работе № 15

Задание 1. Сделать так, чтобы точка в регулярных выражениях соответствовала абсолютно всем символам, включая *\n*. (пересмотреть?)

Задание 2. Сделать поиск и замену многострочного вхождения: заменить *[u] … [/u]* на тэг подчеркивания: *<u>*.

Задание 3. Заменить все открывающие тэги *<a>*.

Задание 4. Организовать перестановку слов в строках. Например, изменить строку "firstName, lastName" на "lastName, firstName". (Использовать строковый метод *replace()*). Результат вывести на экран.

Задание 5. Создать объект со свойствами. Удалите несколько свойств с помощью *rest*. Результат вывести на экран.

Задание 6. Создать объект с помощью оператора *spread*. Удалить несколько значений с помощью *rest*. Добавить своё Имя и Фамилию с помощью *spread*. Результат вывести на экран.