



JS

JavaScript

лекція 5

Регулярні вирази та дати

Шаблони для пошуку в строках, тип даних RegExr

- RegExr - Об'єкт, що містить шаблон регулярного виразу разом з прапорами, що визначають спосіб застосування шаблону.

Синтаксис регулярних виразів

- `function RegExp(pattern : String [,flags : String])`
- `function RegExp(regexObj)`
- `/pattern/[flags]`

Спец символы, неалфавітні символы

Символ	Значение
\	Для обычных символов - делает их специальными. Например, выражение /s/ ищет просто символ 's'. А если поставить \ перед s, то /\s/ уже обозначает пробельный символ. И наоборот, если символ специальный, например *, то \ сделает его просто обычным символом "звездочка". Например, /a*/ ищет 0 или больше подряд идущих символов 'a'. Чтобы найти a со звездочкой 'a*' - поставим \ перед спец. символом: /a*/.
^	Обозначает начало входных данных. Если установлен флаг многострочного поиска ("m"), то также сработает при начале новой строки. Например, /^A/ не найдет 'A' в "an A", но найдет первое 'A' в "An A."
\$	Обозначает конец входных данных. Если установлен флаг многострочного поиска, то также сработает в конце строки. Например, /t\$/ не найдет 't' в "eater", но найдет - в "eat".
*	Обозначает повторение 0 или более раз. Например, /bo*/ найдет 'boooo' в "A ghost boooooed" и 'b' в "A bird warbled", но ничего не найдет в "A goat grunted".
+	Обозначает повторение 1 или более раз. Эквивалентно {1,}. Например, /a+/ найдет 'a' в "candy" и все 'a' в "caaaaaaandy".
?	Обозначает, что элемент может как присутствовать, так и отсутствовать. Например, /e?le?/ найдет 'el' в "angel" и 'le' в "angle." Если используется сразу после одного из квантификаторов *, +, ?, или {}, то задает "нежадный" поиск (повторение минимально возможное количество раз, до ближайшего следующего элемента паттерна), в противоположность "жадному" режиму по умолчанию, при котором количество повторений максимально, даже если следующий элемент паттерна тоже подходит. Кроме того, ? используется в предпросмотре, который описан в таблице под (?=), (?!), и (? :).

Повтори

- Набори [...]
- Діапазони [a-z] [0-9]
- \d - [0-9]
- \w - [a-zA-Z0-9_]
- Діапазо [^...]
- Шаблон (...)

Альтернатива, групування і посилення

- Зворотні посилення \n и \$n

```
var name = "Александр Пушкин";  
name = name.replace(/([а-яё]+) ([а-яё]+)/i, "$2, $1");
```


Флаги

- g - глобальний пошук
- i - регістр
- m - багатостроковий пошук

Властивості екземплярів RegExr

- `global` - чи використовувався глобальний флаг
- `lastIndex` - вказує індекс, з якого починати наступний пошук `match`.
- `source` - незмінна властивість, яка містить текст пошукового виразу
- `multiline` - чи слід використовувати пошук по всіх рядках.
- `ignoreCase` - чи використовувався флаг `i`

Методи екземплярів RegExr та String для пошуку з використанням шаблонів

- `str.search(reg)` - знаходить індекс першого входження
- `str.match(reg)` без флага `g` - повертає масив з властивостями `index` та `input`
- `str.match(reg)` з флагом `g` - повертає звичайний масив з усіма входженнями
- `str.split(reg|substr, limit)` - розбиває строку на масив по роздільнику
- `str.replace(reg, str|func)` - замінить перше входження, якщо передати регулярний вираз з флагом `g` - замінить все
- `regex.test(str)` - повертає `true/false` що значить чи було входження
- `regex.exec(str)` - без флага `g` повертає перше входження (аналог `str.match(reg)`)
- `regex.exec(str)` - повертає перше входження і запам'ятовує позицію в `lastIndex`

Створення дат, тип даних Date

- Створення - `new Date()`
- `new Date(milliseconds)` - час з 1 січня 1970
- `new Date(datestring)`
- `new Date(year, month, date, hours, minutes, seconds, ms)`

Властивості та методи екземплярів Date

- `getFullYear()`
- `getMonth()`
- `getDate()`
- `getHours()`, `getMinutes()`, `getSeconds()`, `getMilliseconds()`
- `getDay()` - день тижня
- `getTime()` - кількість мілісекунд
- `setFullYear(year [, month, date])`
- `setMonth(month [, date])`
- `setDate(date)`
- `setHours(hour [, min, sec, ms])`
- `setMinutes(min [, sec, ms])`
- `setSeconds(sec [, ms])`
- `setMilliseconds(ms)`
- `setTime(milliseconds)` (установлює всю дату по мілісекундам з 01.01.1970 UTC)

Завдання

- Створити функцію для валідації поля повного імені користувача
- Створить функцію для валідації поля e-mail