**String Methods:**

**Методи зміни регістру:**

1. str**.toUpperCase()** – піднімає всі символи до верхнього регістру. **Повертає модифіковану поточну строку.**
2. str**.toLowerCase()** – піднімає всі символи до нижнього регістру. **Повертає модифіковану поточну строку.**

**Методи пошуку підстроки:**

**Повертають індекс:**

1. str**.indexOf(**sub-str, no-mandatory-pos**)** – **повертає індекс позиції / -1**. Другий параметр дозволяє стартувати пошук із будь-якої точки.
2. str**.lastIndexOf(**sub-str, no-mandatory-pos**)** – **повертає індекс позиції / -1** Другий параметр дозволяє стартувати пошук із будь-якої точки.

**Повертають true/false:**

1. str**.includes(**sub-str, pos**) – Повертає true/false**, як результат пошуку підстроки з певної позиції або з початку
2. str**.startWith(**sub-str**)** – **Повертають true/false** перевіряючи чи підстрока є початком строки.
3. str**.endWith(**sub-str**)** - **Повертають true/false** перевіряючи чи підстрока є кінцем строки.

**String Methods:**

**Методи отримання підстроки:**

Повертає строку з start по, не включно end. Якщо не вказати останній параметр, то метод виріже все з start і до кінця

1. str**.slice(**start, <end**)** – Можна задавати від’ємні значення – рахунок починатиметься з кінця. Являється найбільш ефективним методом.
2. str**.substring(**start, <end**)** – Можна задавати end > start і це дасть ту ж строку
3. str**.substr(**start, length**)** - start може бути і від’ємним, length – довжина строки, яку потрібно вирізати.

**Методи взаємодії з кодом символа:**

1. str**.codePointAt(**pos**)** – **повертає код символа** в заданій позиції
2. str**.fromCodePoint(**kod**) – повертає символ** по його коду.
3. str.**localCompare(**str2**)** – повертає результат порівняння строк в числах ><= 0

**Методи** **модифікації строки**:

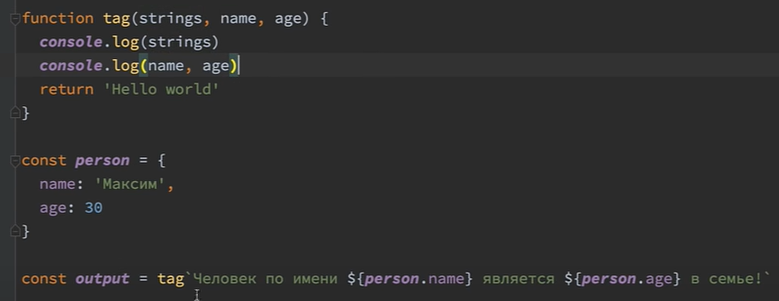
1. str**.repeat**(num) – повертає строку **повторену без пробілів** вказану к-сть разів
2. str**.trim()** – **повертає строку без пробілів**
3. str**.trimLeft() ||** str**.trimStart()** / str**.trimRight() ||** str**.trimEnd()** – **повертає строку без пробілів** по вказаній стороні
4. str**.padStart(**num, str**)** – вказуємо мінімальний розмір строки (якщо строка менша, то спереду/позаду добавляються пробіли, щоб заповнити мінімальне значення): **num** вказує мінімальний розмір строки, **str** – необов’язковий параметр вказуючий на те, як заповнити недостающі символи, наприкад іншою срокою
5. str**.padEnd(**num, str**) –** добавляє недостатні символи вкінець
6. str.**replaceAll**(substr, newstr) – міняє всі співпадіння в строці на вказану нову строку

**Шаблони і патерни:**

* Можна написати метод, який прийматиме в себе шаблонну строку і розбиватиме її на аргументи (1- параметр це вся шаблонна строка розбита на масив по динамічним змінним ${}, решта параметрів залежать від к-сті цих самих динамічних змінних ‘${}’)

**logPerson`Ім’я:** ${personName}**, Вік:** ${personAge}**!`**

**logPerson(**s, name, age**) –** s[‘**Ім’я:** ’, ‘**, Вік:** ’, ‘!’], name = personName, age = personage



В string змінну потрапляє масив, зі строки розбитої по параметрам, а в самі параметри функції, отримуємо динамічні параметри строки (person.name && person.age)

