Кафедра комп’ютерної та програмної інженерії

УДУ імені М.П. Драгоманова

Факультет математики, інформатики та фізики

Кафедра комп’ютерної та програмної інженерії

ЛАБОРАТОРНА РОБОТА № 8

ТЕМА:: Локалізація та інтернаціоналізація

з курсу

«Програмування в інтернеті»

Студент Яворський Нікіта

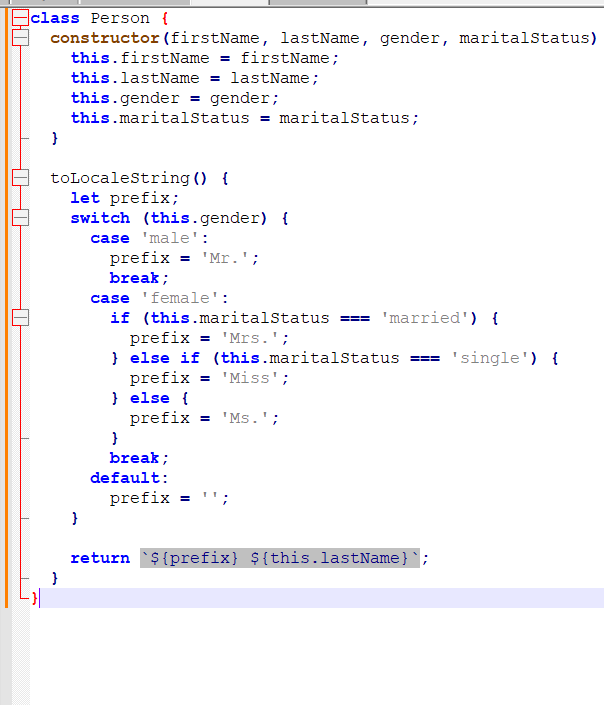
Група 31ІПЗ

Факультет МІФ

Викладач: Кархут В.Я.

Київ 2023

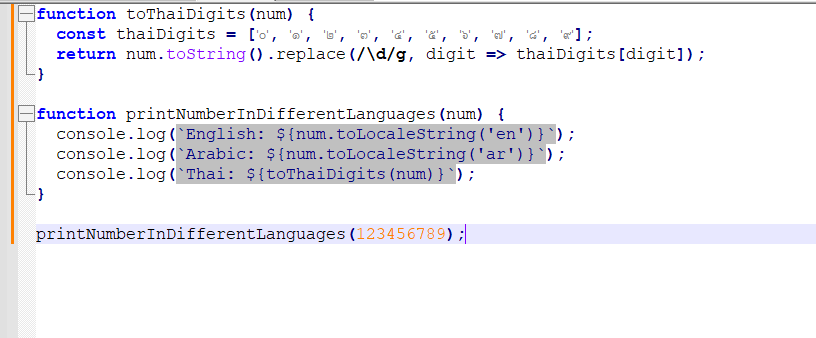
1. Реалізуйте клас Person з полями для імені, прізвища, статі та сімейного становища. Реалізуйте метод toLocaleString для форматування імені, наприклад, 'Ms. Smith', 'Frau Smith', 'Mme Smith'. Дізнайтесь, які форми ввічливості прийняті у різних мовах, і реалізуйти такі варіанти як Ms. або Mrs./Miss.



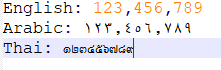
Цей клас містить чотири поля - ім'я, прізвище, стать та сімейний стан. Метод toLocaleString визначає, який префікс (заголовок) повинен бути використаний для даної особи на основі її статі та сімейного стану. За замовчуванням, якщо стать не вказана або невідома, префікс буде порожнім рядком.

1. Рeалізуйте програму яка приймає число та друкує його у трьох версіях - англійських, арабських та тайських цифр.

Ось програма, яка приймає число та друкує його у трьох версіях - англійських, арабських та тайських цифр. Для англійської та арабської версій використовується функція toLocaleString(), а для тайської - функція toThaiDigits():



При виклику printNumberInDifferentLanguages(123456789) результат буде наступним:



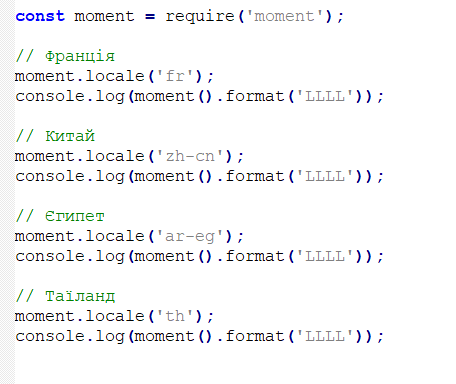
1. Напишіть програму для демонстрації стилів форматування дати та часу у Франції, Китаї, Єгипті та Таїланді (з використанням тайських цифр).

Для демонстрації стилів форматування дати та часу у різних країнах, використаємо бібліотеку moment.js, яка має широкі можливості для форматування дати та часу.

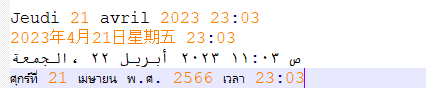
Встановимо бібліотеку за допомогою npm команди:

a3

Після встановлення бібліотеки, створимо дату now та виконаємо форматування дати та часу за допомогою методу format() для кожної країни. Для кожної країни використаємо різні формати дати та часу.

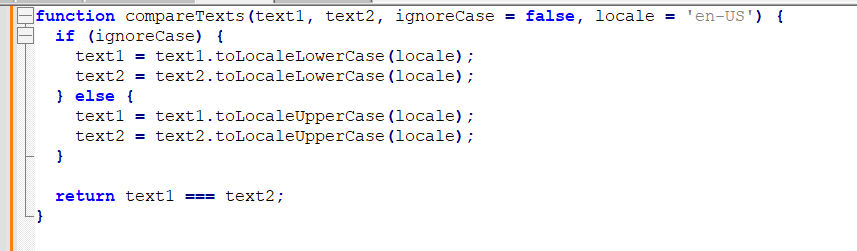


Результат:



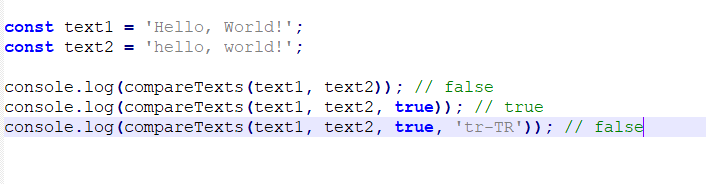
1. Напишіть функцію порівняння двох текстових фрагментів відповідно до локалі. Вона повинна працювати в режимах ігнорування та врахування регістру.

Для порівняння текстових фрагментів відповідно до локалі та режимів ігнорування та врахування регістру, можна використати вбудовані функції toLocaleLowerCase та toLocaleUpperCase. Нижче наведено приклад функції, яка порівнює два текстових фрагменти з урахуванням локалі та режимів ігнорування та врахування регістру:



Дана функція приймає два текстових фрагменти text1 та text2, а також два необов'язкових параметри ignoreCase та locale. Параметр ignoreCase вказує на те, чи необхідно ігнорувати регістр символів при порівнянні (за замовчуванням false). Параметр locale вказує на локаль, у якій буде проводитись порівняння (за замовчуванням 'en-US').

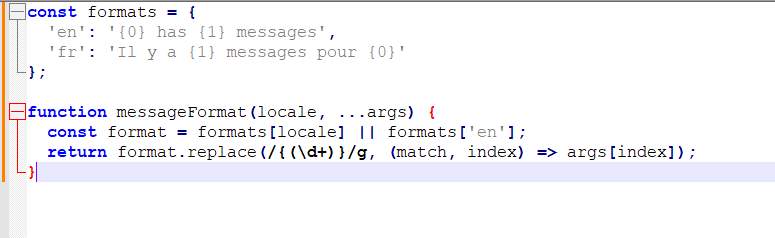
Приклад використання функції:



1. Розглянемо шаблон '{0} has {1} messages'.Його французька версія повинна мати вигляд 'Il y a {1} messages pour {0}'. При форматуванні повідомлення аргументи передаються у фіксованому порядку, що не залежить від мови. Напишіть функцію messageFormat, яка приймає шаблонний рядок та змінну кількість аргументів. Придумайте механізм який виставлятиме аргументи в шаблон відповідно до локалі.

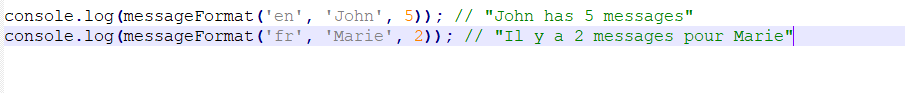
Одним з підходів до реалізації функції messageFormat є використання об'єкту-словника для зберігання локалей та відповідних форматів шаблонів. Кожен ключ в об'єкті-словнику відповідає мові, а значення - формату шаблону для цієї мови.

Наприклад, для англійської мови формат шаблону може виглядати так: "{0} has {1} messages", а для французької - "Il y a {1} messages pour {0}". Тоді функція messageFormat може виглядати таким чином:

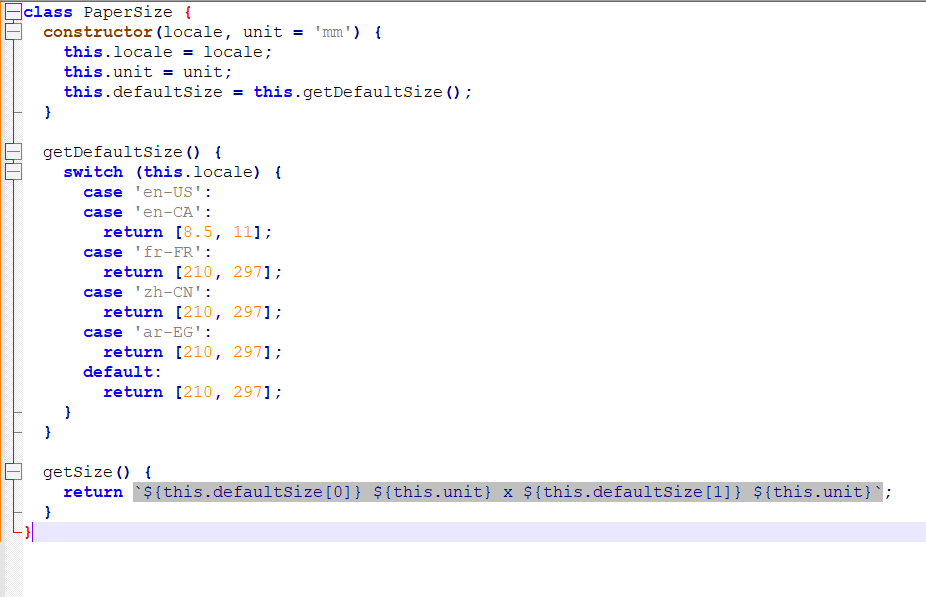


Перший аргумент функції - це локаль, яку потрібно використати для форматування шаблону. Решта аргументів - це дані, які потрібно вставити в шаблон. Функція messageFormat спочатку вибирає формат шаблону для поточної локалі, або використовує англійський формат, якщо локаль не знайдена. Потім функція використовує метод replace для заміни плейсхолдерів у форматі шаблону на відповідні аргументи.

Приклад використання функції messageFormat:



1. Запропонуйте клас для відображення розмірів аркуша паперу, що залежить від локалі, з використанням бажаної одиниці вимірювання та розміру за умовчанням для даної локалі. У всіх країнах, окрім США та Канади, розміри аркушів паперу визначаються стандартом ISO 216. Лише три країни ще не перейшли офіційно на метричну систему: Ліберія, М'янма (Бірма) та США.



Цей клас містить два поля: locale (локаль, для якої відображатимуться розміри) та unit (одиниця вимірювання за умовчанням, яка буде використовуватися для відображення розмірів).

Конструктор класу приймає два аргументи: locale та unit. Якщо одиниця вимірювання не вказана, за умовчанням використовується міліметр (mm).

Метод getDefaultSize повертає розмір за умовчанням для заданої локалі. У випадку локалей, для яких немає стандартів ISO 216, повертається розмір за умовчанням 210 мм на 297 мм. У випадку США та Канади повертається розмір 8,5 на 11 дюймів.

Метод getSize повертає розмір аркуша паперу у форматі 'ширина x висота', де ширина та висота відображаються в одиницях вимірювання, переданих у конструктор. Наприклад, якщо одиниця вимірювання задана як 'mm', а розмір аркуша за умовчанням становить 210 на 297 мм, метод getSize поверне рядок '210 mm x 297 mm'.