**Задание №1 на стажировке**  
По теме   
«Собственный JSON.stringify»

Выполнил: Ноздряков Богдан Валериевич

Проверил: Пармузин Александр Игоревич

Санкт-Петербург

2024

**Условие:**

-Написать свой собственный JSON.stringify() без использования стандартных методов JSON, поддержав в том числе:

* все параметры функции во всех вариантах их возможных значений
* все типы, которые обрабатывает JSON.stringify() (например Date, Symbol)
* генерировать ошибки всех тех же типов, в т.ч. отсекать ошибку циклических ссылок

В результате собственная функция при любом содержимом параметров должна выдавать строку полностью идентичную результату выполнения JSON.stringify(). Это можно проверять прямыми тестами на сравнение результатов.

**Решение:**

-Анализ метода stringify объекта JSON:

Функция stringify преобразует значение в строку JSON. Принимает три параметра:

1. value – обязательный; представляет собой сам элемент, который должен преобразоваться в строку JSON
2. replacer – необязательный; представляет собой функцию, которая определяет как именно преобразовать значение value. Также может представлять из себя массив – в таком случае он будет интерпретирован как список свойств, которые должны быть включены в результирующую строку JSON
3. space – необязательный; представляет собой отступ (количество пробелов), который должен использоваться для форматирования вывода

Данная функция:

1. Возвращает строку в формате JSON
2. Выдает исключения в следующих случаях:
3. Параметр value содержит циклическую ссылку
4. Встречается BigInt значение
5. Если передается в качестве value – функция, undefined, symbol, то возвращается undefined

**-Для написание собственного JSON.stringify – создадим класс CUSTOM\_JSON, в котором определим статический метод stringify, а после проведем тесты с реальным JSON.stringify и с собственным CUSTOM\_JSON на каждом типе данных, который есть в JavaScript.**

**-Всего проведем тестов 5 серий тестов:**

1) Тесты № a – для тестирования работы функции при одном параметре value

2) Тесты № b – для тестирования работы функции при параметрах value и replacer

3) Тесты № c – для тестирования работы функции при параметрах value и space

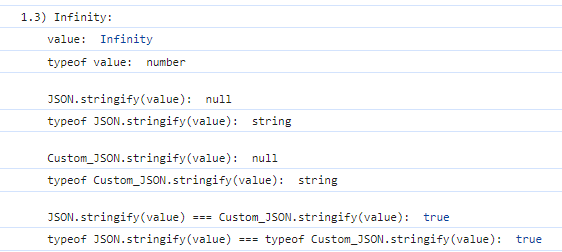
4) Тесты № d – для тестирования работы функции при параметрах value, replacer, space

5) Тесты № e – для тестирования работы функции при наличии циклической ссылки

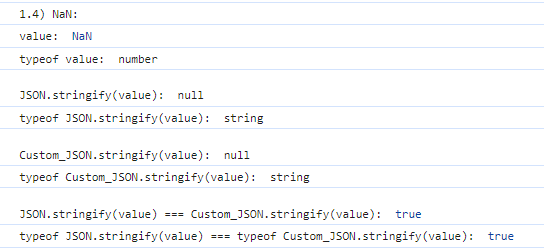
**-Тесты № a (без replacer и без space – файлы ./index.html и ./main.js):**

1. Тип данных – number:

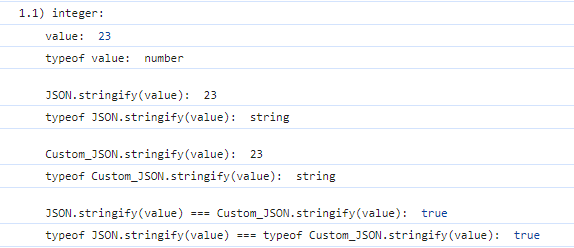
a. Подтип Infinity:



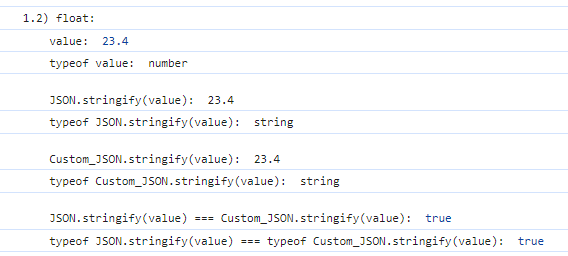
b. Подтип NaN:



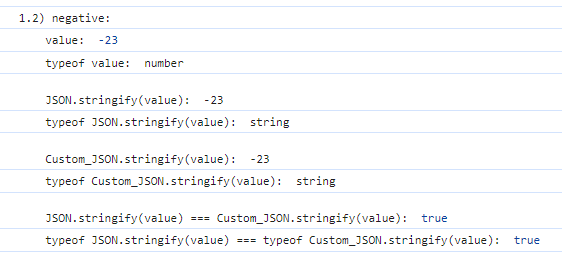
c. Целое число:



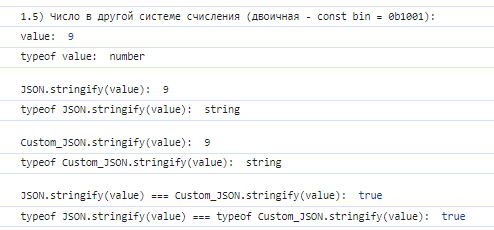
d. Вещественное число:



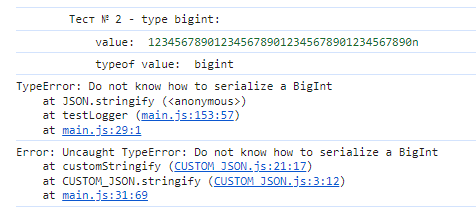
e. Отрицательное число:



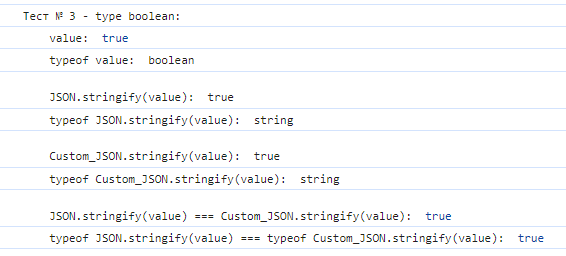
f. Число в другой системе счисления (двоичная) – ob1001:



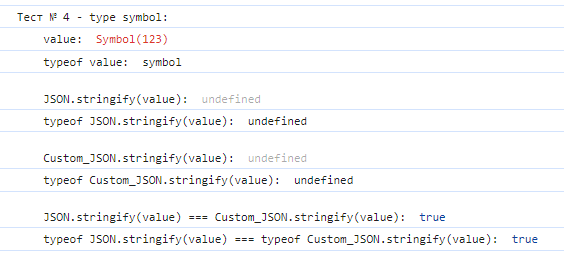
2. Тип данных – bigint:



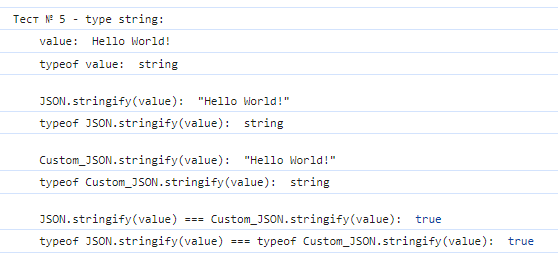
3. Тип данных – boolean:



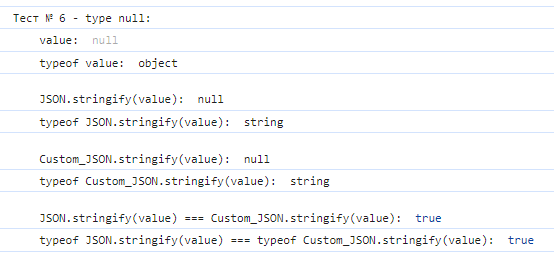
4. Тип данных – symbol:



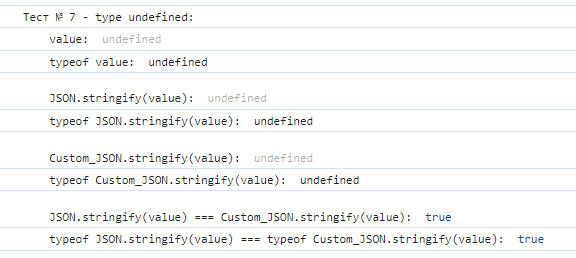
5. Тип данных – string:



6. Тип данных – null:



7. Тип данных – undefined:



8. Тип данных – object:

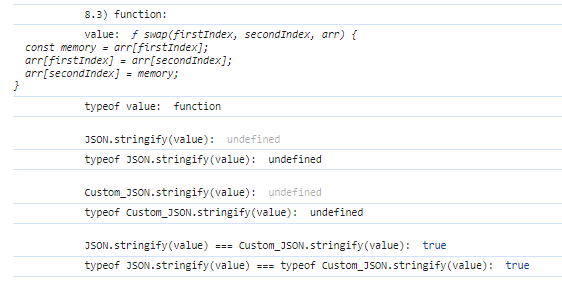
a. Подтип – object:



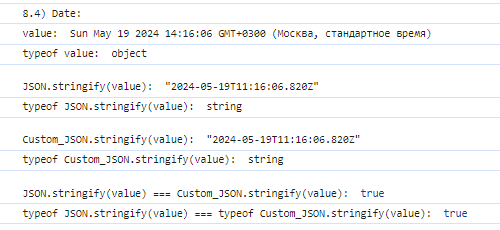
b. Подтип – array:



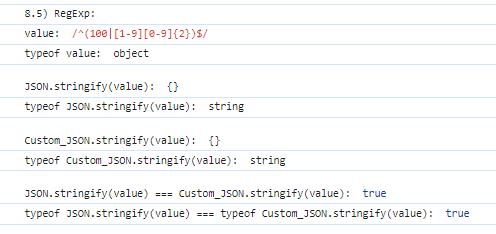
c. Подтип – function:



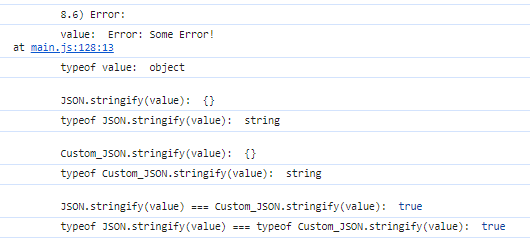
d. Подтип - Date:



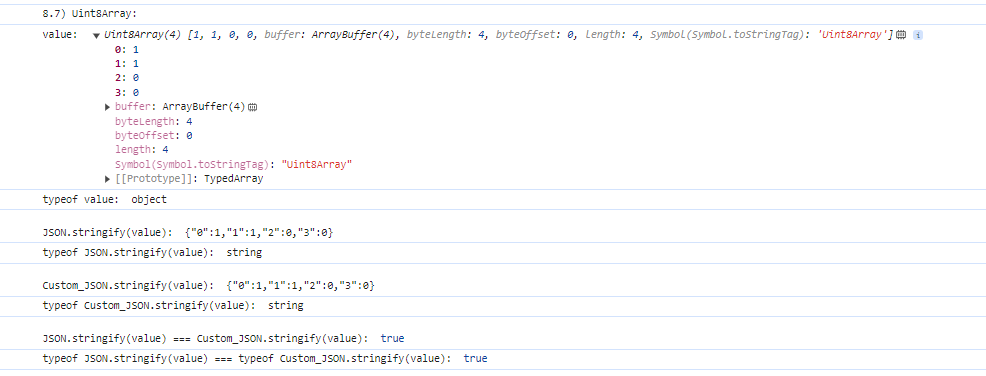
e. Подтип - RegExp:



f. Подтип – Error:

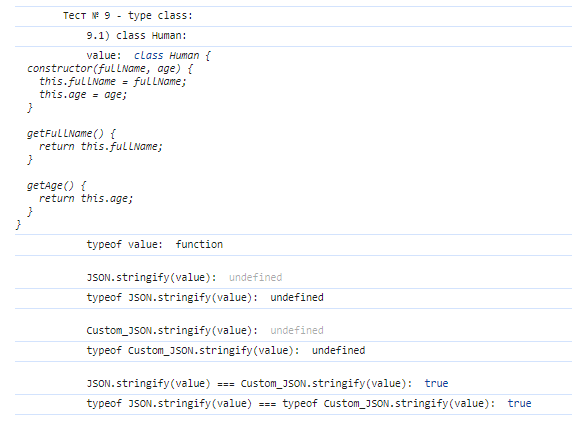


g. Подтип - Uint8Array:

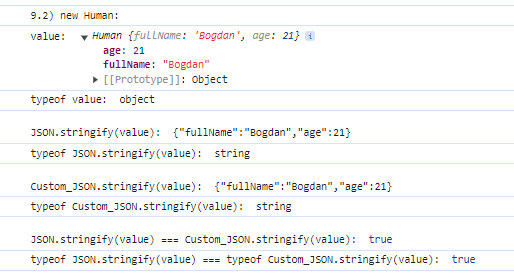


9. Тип данных - class:

a. Сам класс:



b. Экземпляр класса:



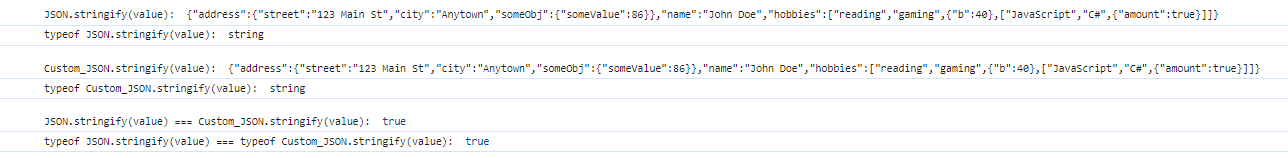
-Также стоит отметить, что CUSTOM\_JSON будет также правильно работать и для оставшихся подтипов объекта – Map, Set, Promise, Math, JSON, Intl, URLSearchParams, Proxy, Reflect.

**-Тесты № b (с replacer и без space – файлы ./fragment/replacer.html и ./js/replacer.js):**

**1. value – это объект:**

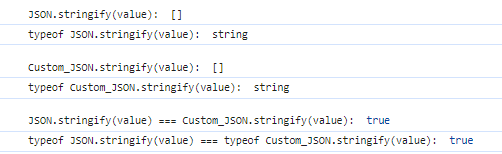
-Тест № 1:





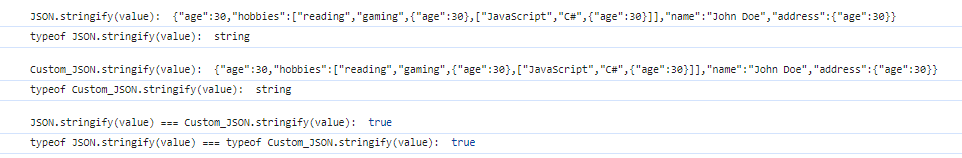
-Тест № 2:





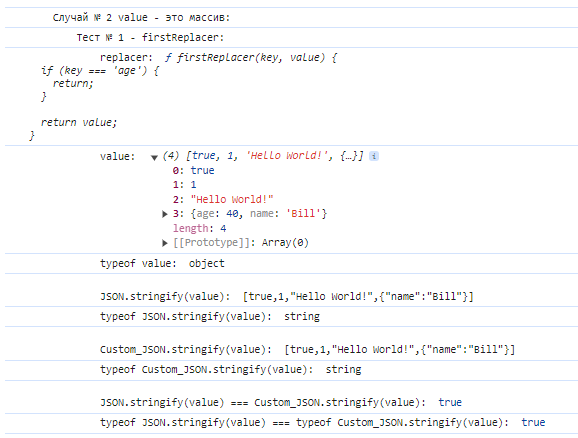
-Тест № 3:



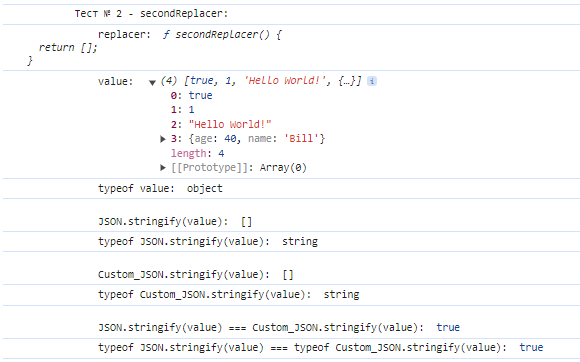


**2. value – это массив:**

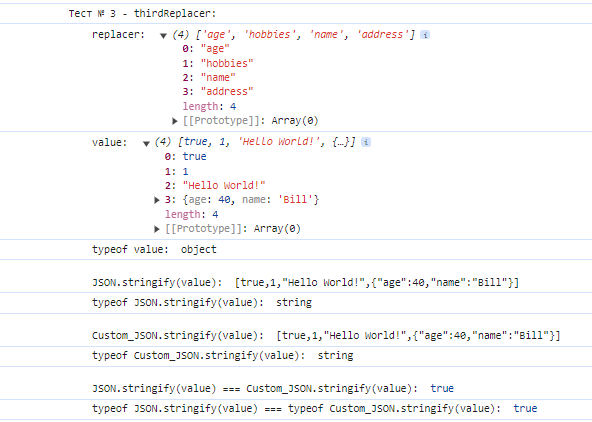
-Тест № 1:



-Тест № 2:

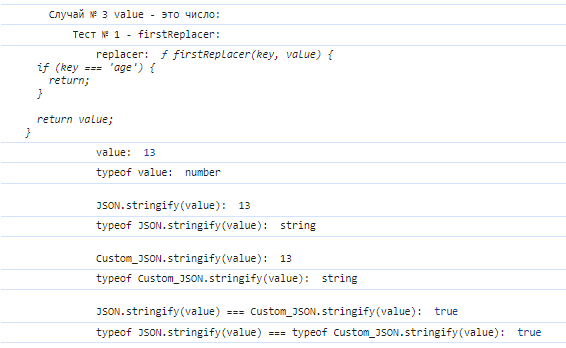


-Тест № 3:

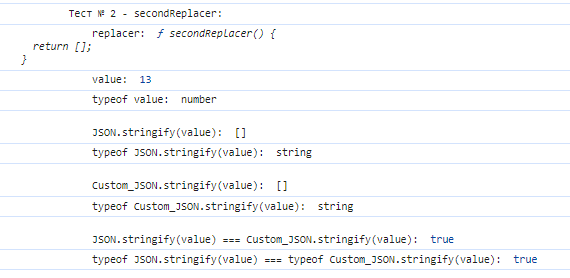


**3. value – это число:**

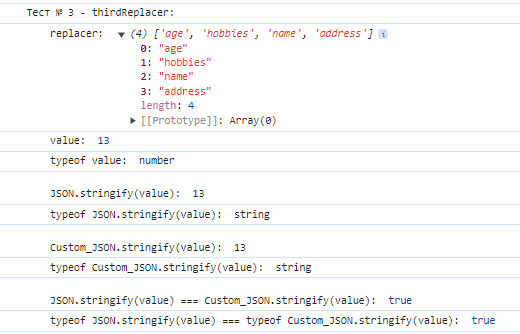
-Тест № 1:



-Тест № 2:

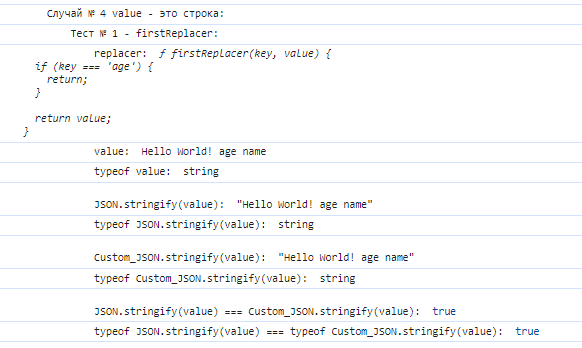


-Тест № 3:

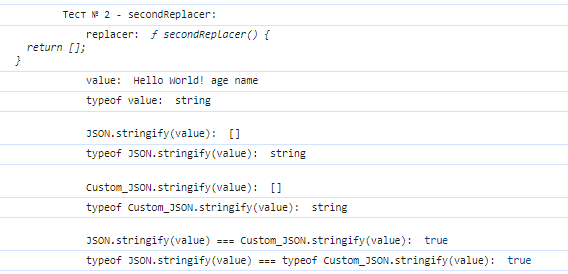


**4. value – это строка:**

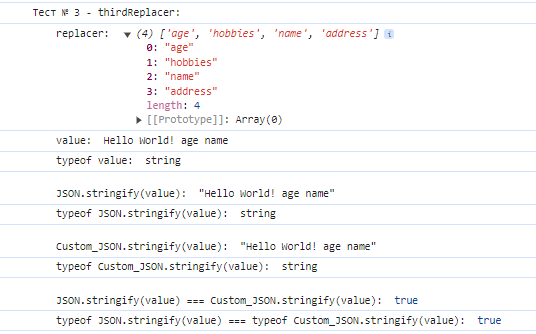
-Тест № 1:



-Тест № 2:



-Тест № 3:



**-Тесты № c (со space и без replacer – файлы ./fragment/space.html и ./js/space.js):**

1. Тип данных – number:

a. Подтип Infinity:

-Результат такой же, как и в тесте № a

b. Подтип NaN:

-Результат такой же, как и в тесте № a

c. Целое число:

-Результат такой же, как и в тесте № a

d. Вещественное число:

-Результат такой же, как и в тесте № a

e. Отрицательное число:

-Результат такой же, как и в тесте № a

f. Число в другой системе счисления (двоичная) – ob1001:

-Результат такой же, как и в тесте № a

2. Тип данных – bigint:

-Результат такой же, как и в тесте № a

3. Тип данных – boolean:

-Результат такой же, как и в тесте № a

4. Тип данных – symbol:

-Результат такой же, как и в тесте № a

5. Тип данных – string:

-Результат такой же, как и в тесте № a

6. Тип данных – null:

-Результат такой же, как и в тесте № a

7. Тип данных – undefined:

-Результат такой же, как и в тесте № a

8. Тип данных – object:

a. Подтип – object:



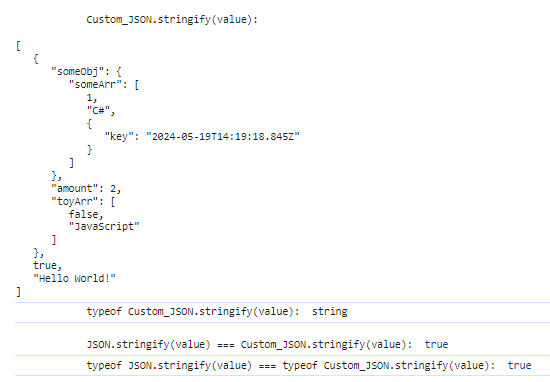




b. Подтип – array:







c. Подтип – function:

-Результат такой же, как и в тесте № a

d. Подтип - Date:

-Результат такой же, как и в тесте № a

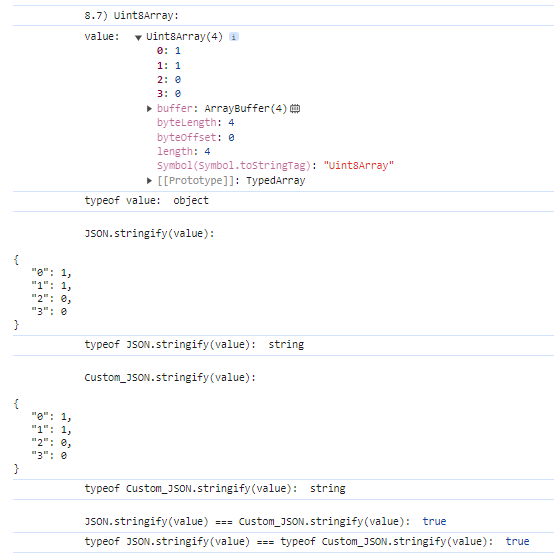
e. Подтип - RegExp:

-Результат такой же, как и в тесте № a

f. Подтип – Error:

-Результат такой же, как и в тесте № a

g. Подтип - Uint8Array:

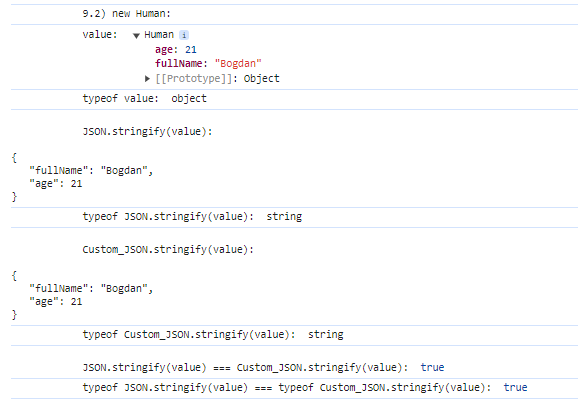


9. Тип данных - class:

a. Сам класс:

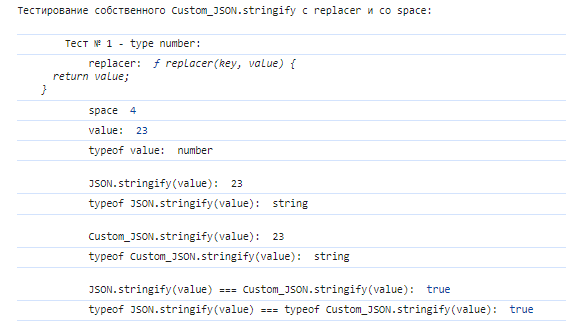
-Результат такой же, как и в тесте № a

b. Экземпляр класса:

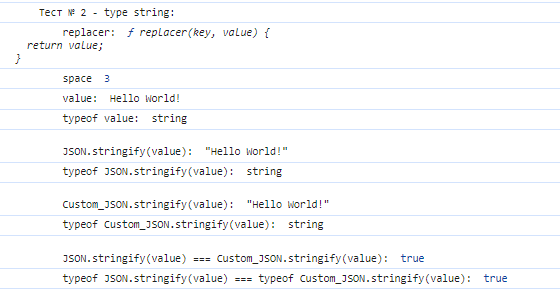


**-Тесты № d (и с replacer, и со space - файлы ./fragment/all.html и ./js/all.js):**

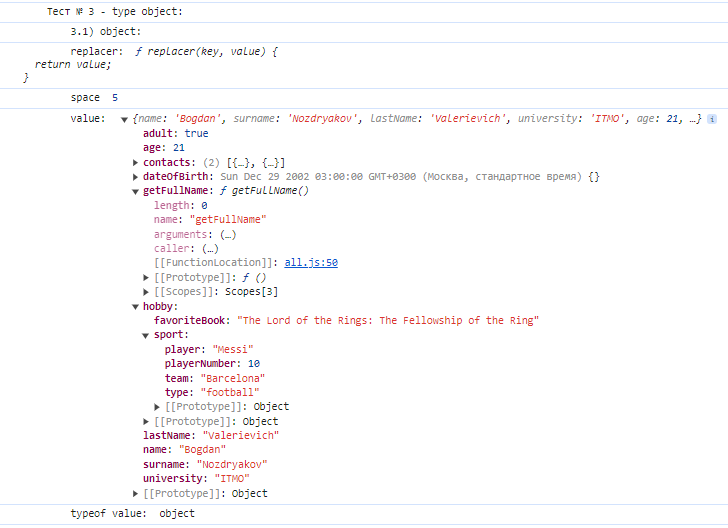
**1. value – это число:**



**2. value – это строка:**



**3. value – это объект:**

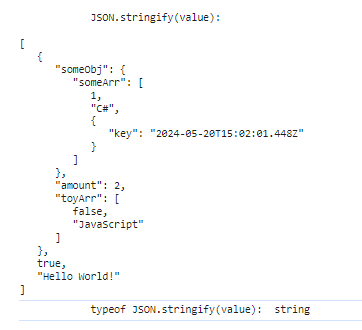


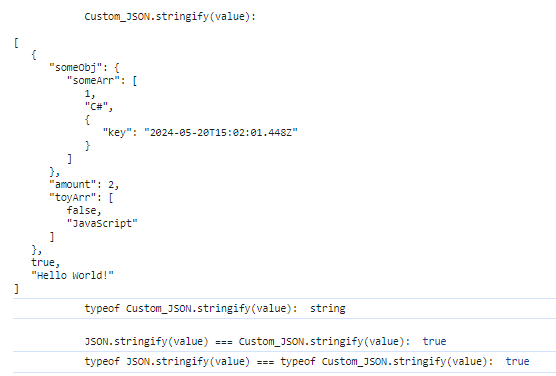




**4. value – это массив:**

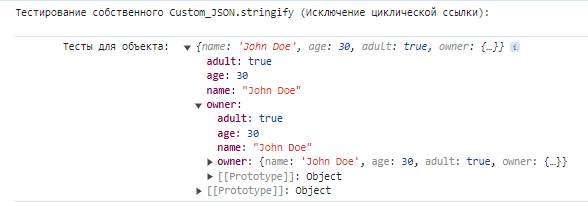






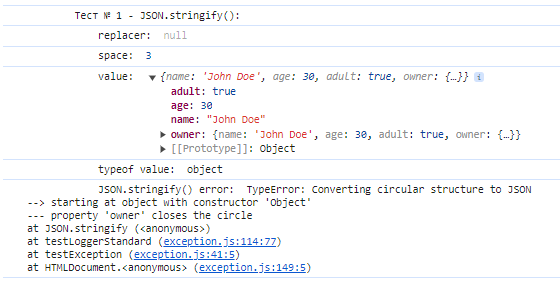
**-Тесты № e (Обработка исключительной ситуации циклической ссылки – файлы ./fragment/exception.html и ./js/exception.js):**

**1) Тесты на объекте (value – объект):**

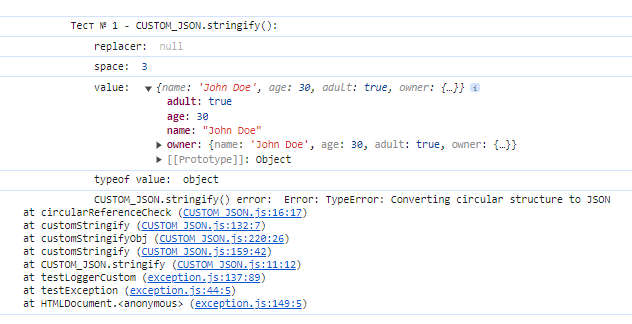


-Тест № 1:

1. JSON.stringify():

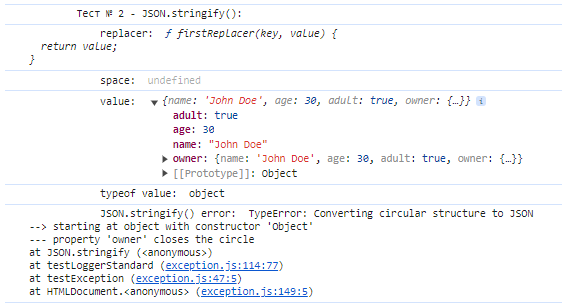


1. CUSTOM\_JSON.stringify():

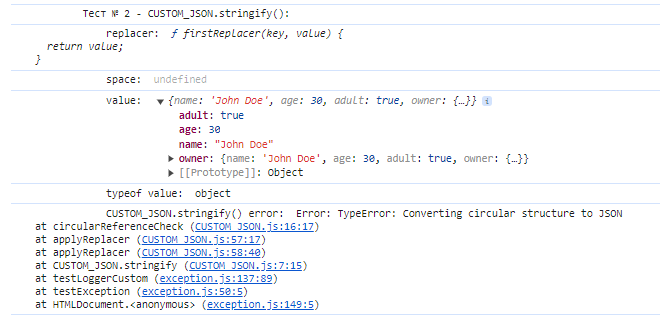


-Тест № 2:

a. JSON.stringify:

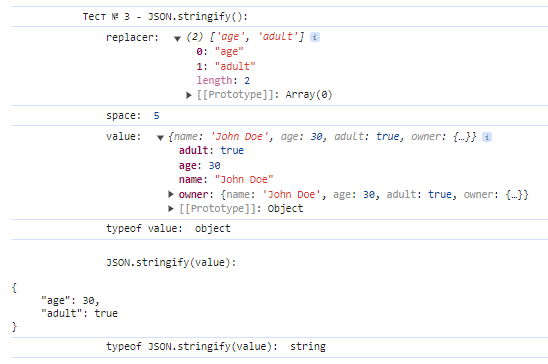


1. CUSTOM\_JSON.stringify:

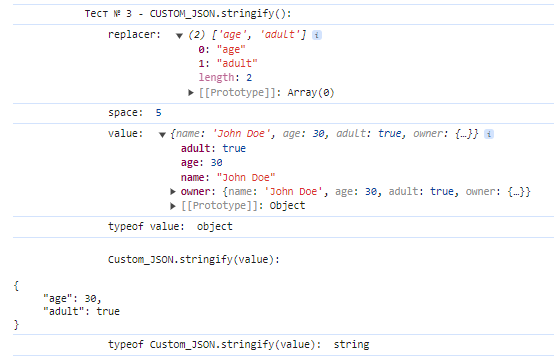


-Тест № 3:

a. JSON.stringify:

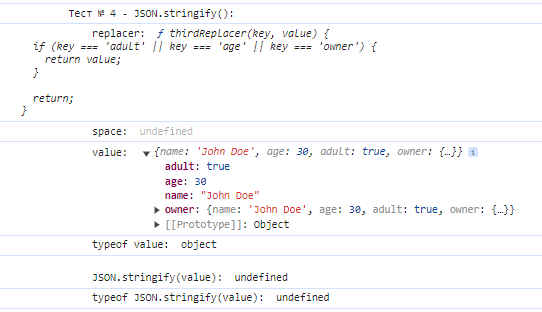


b. CUSTOM\_JSON.stringify:

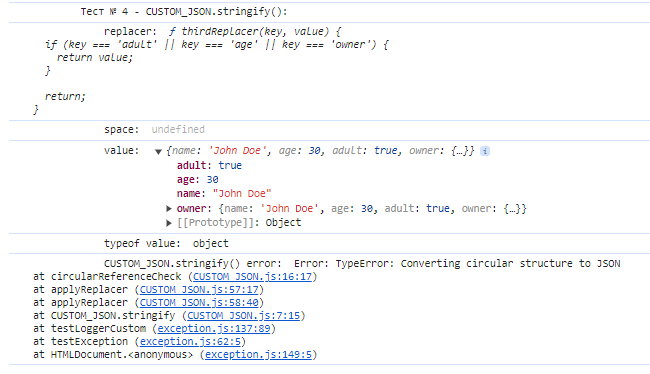


-Тест № 4:

a. JSON.stringify:

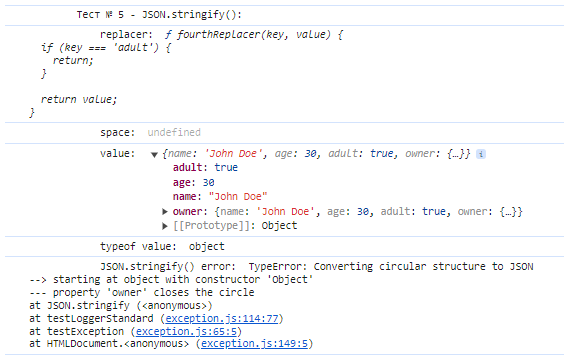


b. CUSTOM\_JSON.stringify:



-Тест № 5:

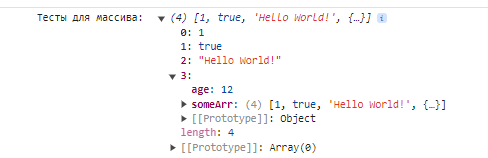
a. JSON.stringify:



b. CUSTOM\_JSON.stringify:

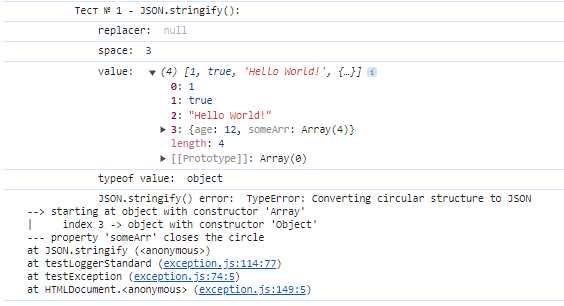


**2) Тесты на массиве (value – массив):**

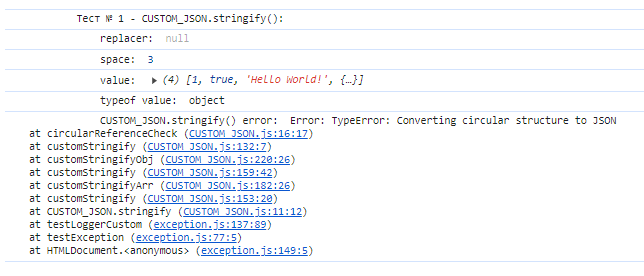


-Тест № 1:

1. JSON.stringify():

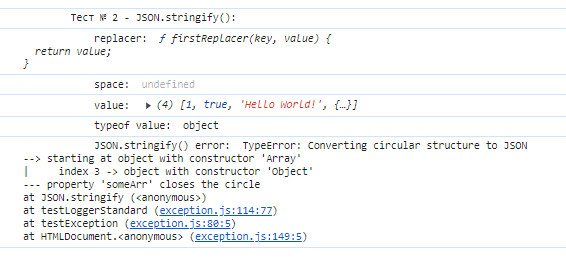


1. CUSTOM\_JSON.stringify():

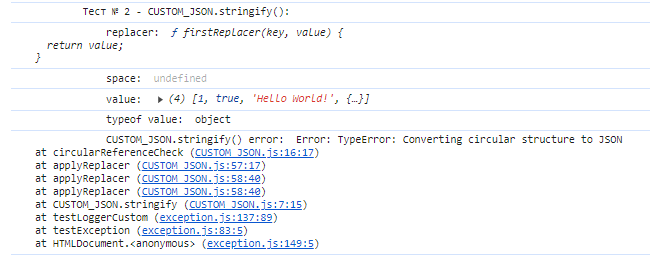


-Тест № 2:

a. JSON.stringify:

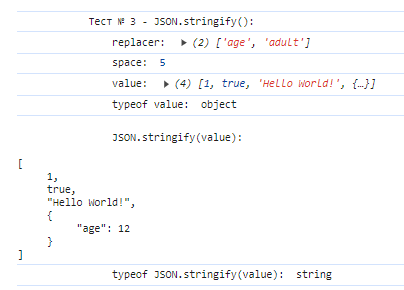


1. CUSTOM\_JSON.stringify:

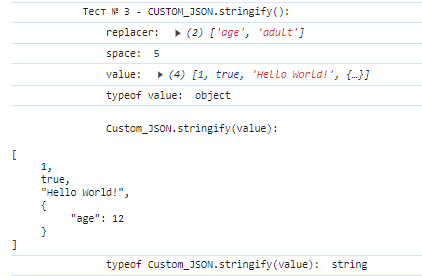


-Тест № 3:

a. JSON.stringify:

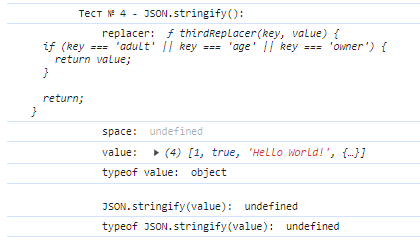


b. CUSTOM\_JSON.stringify:

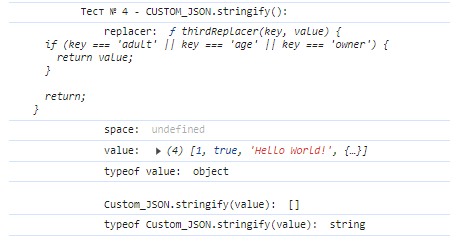


-Тест № 4:

a. JSON.stringify:

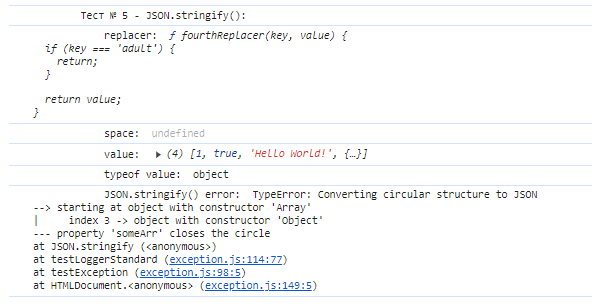


b. CUSTOM\_JSON.stringify:



-Тест № 5:

a. JSON.stringify:



b. CUSTOM\_JSON.stringify:

