1. += Оператор сложения с присваиванием (+=) добавляет значение правого операнда

к переменной и присваивает переменной результат.

Пример:

let a = 2;

let b = 'hello';

console.log(a += 3);

будет 5 т.к a прибавляет и присваивает 3!!!!

console.log(b += ' world');

2. Просто поместите внутрь шаблона с обратными кавычками ещё одни, обернув их в подстановку ${ }.`` бэктики не забывать

3. Оператор typeof возвращает строку, указывающую тип операнда.Он показывается что это.Число,строка и тд.

Пример:

console.log(typeof 42);

// expected output: "number"

console.log(typeof 'blubber');

// expected output: "string"

console.log(typeof true);

// expected output: "boolean"

console.log(typeof undeclaredVariable);

// expected output: "undefined"

if (typeof(times) !== 'number' || times <= 0) {

return num;

}

4. Присваение:

Пример 1:

const a = prompt('Один из последних просмотренных фильмов?', ''),

b = prompt('На сколько оцените его?', ''),

c = prompt('Один из последних просмотренных фильмов?', ''),

d = prompt('На сколько оцените его?', '');

personalMovieDB.movies[a] = b;

personalMovieDB.movies[c] = d;

Пример 2:

for (let i = 1; i <= 3; i++) {

personalMovieDB.genres[i - 1] = prompt(`Ваш любимый жанр под номером ${i}`);

}

5. Метод Number.isInteger()определяет переданное значение большого числа.

определяет, является ли передаваемое значение целым числом и возвращает логическое значение.

Описание

Если целевое значение составляет числовое значение, возвращается true. Если значение NaNили Infinity, то возвращается

Метод также подтверждается true, если это произошло числом с целым числом, которое может быть представлено в исчисленном виде.

6. Метод Math.ceil() - округление вверх. Округляет аргумент до следующего большего присутствия.

7. Метод Math.floor() - округление вниз. Округляет аргумент до ближайшего меньшего целого.

8. Метод Math.max() - находит большее число

9. join()Метод включения всех элементов массива (или массивоподобного объекта ) в структуру.

**1.Объекты**

const personalPlanPeter = {

    name: "Peter",

    age: "29",

    skills: {

        languages: ['ru', 'eng'],

    }

};

*Разбираем на этом примере*

***Как обратиться к свойству объекта* (age) и другим св-вам, самый простой способ: console.log(personalPlanPeter.age);**

**Перебрать все свойства объекта Будем перебирать ключи в personalPlanPeter**

for(let key in personalPlanPeter) {

    console.log(`Свойство ${key} имеет значение ${personalPlanPeter[key]}`);

}

**Key – ключи которые перебераем (name,age,skills), мы обращаемся именно к ключу объета**

**personalPlanPeter[key] – это значение ключа, а тут мы обращаемся именно к значению ключа**

**Ремарка: когда одно из св-в объекта является тоже объектом пример skills тогда: мы внутри проверяем при помощи условия, если значение ключа будет объектом, мы можем запустить перебор внутри перебора**

for(let key in personalPlanPeter) {

    if(typeof(personalPlanPeter[key]) === 'object') {

        for (let i in personalPlanPeter[key]) {

            console.log(`Свойство ${i} имеет значение ${personalPlanPeter[key][i]}`);

        }

        }

}

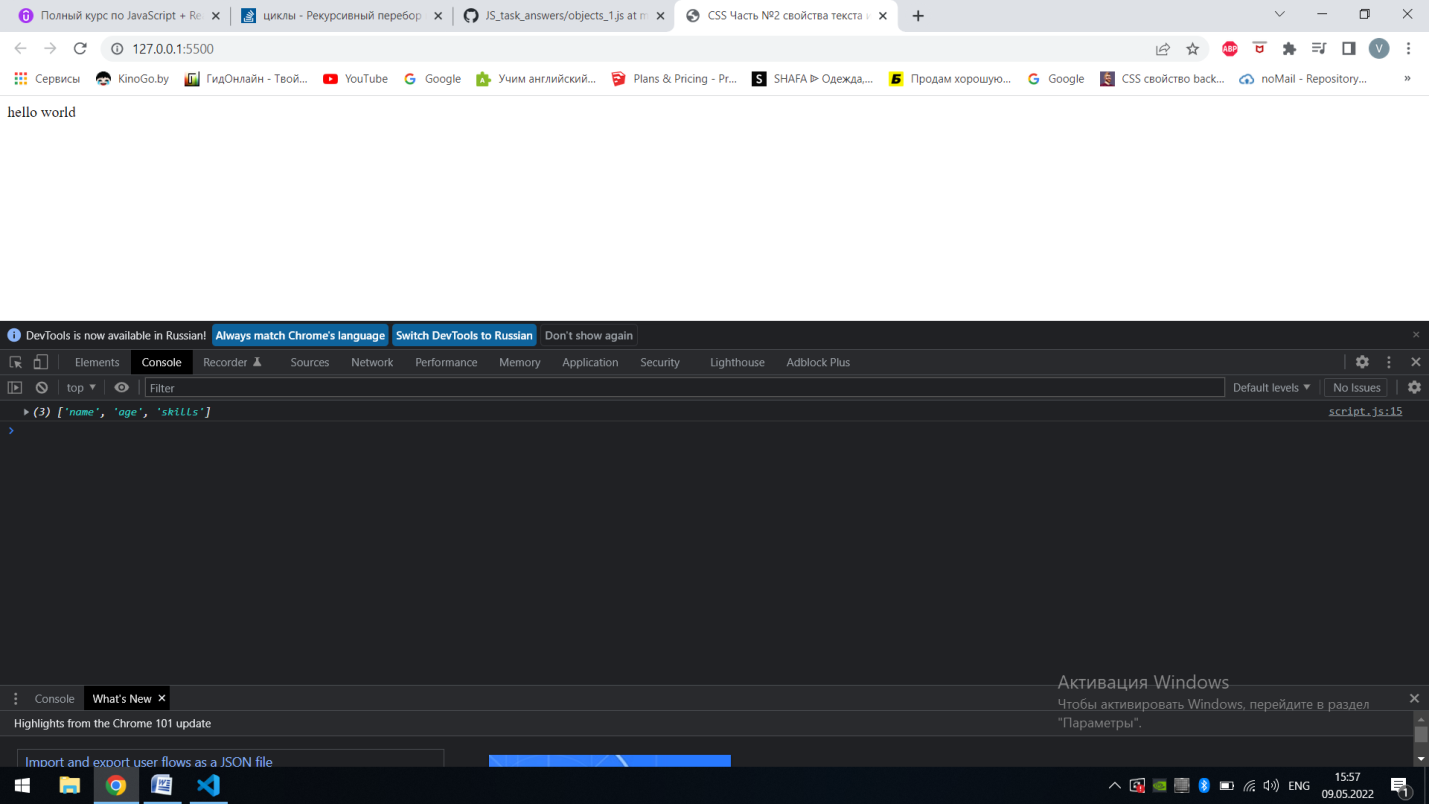
**тут мы видим условие, что если в key есть объект( в данном случае это langueges) то мы перебираем его**

**Фун-ции и методы которые есть в объекте**

Количество св-в в объекте Object.keys()

console.log(Object.keys(personalPlanPeter));

и мы получаем массив со всеми ключами



**Есть свойство length у массивов и мы можем узнать какое количество элементов там содержится**

console.log(Object.keys(personalPlanPeter).length);

**Диструктиризация объекта**

**Мы будем вытаскивать переменные в качестве отдельных структур, а потом использовать**

const {exp} = personalPlanPeter.skills;

**и теперь это отдельная переменная**