# Solutions - Exercise: JavaScript Syntax and Operators

## 1.I like JavaScript!

function greeting(name){

console.log(`Hello ${name}, do you like JavaScript?`);

}

## 2.Even Numbers 1 to N

function evenNumbers(n){

let num = Number(n);

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

if (i % 2 === 0){

console.log(i)

}

}

}

## 3.Fruit

function buyFruits(fruit, weightInGramsString, pricePerKilogramString){

let weightInKilograms = Number(weightInGramsString) / 1000;

let pricePerKilogram = Number(pricePerKilogramString);

let price = weightInKilograms \* pricePerKilogram;

console.log(`I need ${price.toFixed(2)} leva to buy ${weightInKilograms.toFixed(2)} kilograms ${fruit}.`);

}

## 4.Fitness Rates

function showServicePrice(weekDay, service, hourString){

let hour = Number(hourString);

if (hour < 8.0 || hour > 22.0){

console.log('The gym does not work in this time!')

return;

}

let price = 0;

switch(weekDay){

case 'Monday':

case 'Tuesday':

case 'Wednesday':

case 'Thursday':

case 'Friday': if (hour > 15.0) {

price += 2.5;

} switch(service){

case 'Fitness': price += 5; break;

case 'Sauna': price += 4; break;

case 'Instructor': price += 10; break;

} break;

case 'Saturday':

case 'Sunday': switch(service){

case 'Fitness': price = 8; break;

case 'Sauna': price = 7; break;

case 'Instructor': price = 15; break;

} break;

}

console.log(price);

}

showServicePrice('Monday', 'Sauna', 15.30)

showServicePrice('Sunday', 'Fitness', 22.00)

## 5.Greatest Common Divisor – GCD

function findGreatestCommonDivisor(firstNumber, secondNumber){

let lowerNumber = Math.min(firstNumber, secondNumber);

for(let i = lowerNumber; i > 0; i--){

if (firstNumber % i === 0 && secondNumber % i === 0){

console.log(i);

break;

}

}

}

findGreatestCommonDivisor(15, 5)

findGreatestCommonDivisor(2154, 458)

## 6.Same Numbers

function checkEqualDigits(inputNumber){

let numberString = inputNumber.toString();

let digitsSum = 0;

let areEqualDigits = true;

for (let i = 0; i < numberString.length; i++){

if (numberString[0] !== numberString[i]){

areEqualDigits = false;

}

digitsSum += Number(numberString[i]);

}

console.log(areEqualDigits);

console.log(digitsSum);

}

checkEqualDigits(2222222)

checkEqualDigits(1234)

## 7.Time to Walk

function calculateTimeToWalk(stepsNumber, footprintLengthInMeters, speedInKilometersPerHour){

let pathInMeters = stepsNumber \* footprintLengthInMeters;

let pauseTimeInSeconds = Math.floor(pathInMeters / 500) \* 60;

let pathInKilometers = pathInMeters / 1000;

let goTimeInHours = pathInKilometers / speedInKilometersPerHour;

let goTimeInSeconds = goTimeInHours \* 3600;

let walkTimeInSeconds = goTimeInSeconds + pauseTimeInSeconds;

let walkHours = Math.floor(walkTimeInSeconds / 3600);

let walkMinutes = Math.floor((walkTimeInSeconds % 3600) / 60);

let walkSeconds = Math.round((walkTimeInSeconds % 3600) % 60);

//let walkSeconds = (walkTimeInSeconds % 3600) % 60;

//let walkSeconds = Math.floor((walkTimeInSeconds % 3600) % 60);

let walkHoursString;

if (walkHours < 10){

walkHoursString = `0${walkHours}`;

}

else{

walkHoursString = `${walkHours}`;

}

let walkMinutesString;

if (walkMinutes < 10){

walkMinutesString = `0${walkMinutes}`;

}

else{

walkMinutesString = `${walkMinutes}`;

}

let walkSecondsString;

if (walkSeconds < 10){

walkSecondsString = `0${walkSeconds}`;

}

else{

walkSecondsString = `${walkSeconds}`;

}

console.log(`${walkHoursString}:${walkMinutesString}:${walkSecondsString}`);

}

calculateTimeToWalk(4000, 0.60, 5)

calculateTimeToWalk(2564, 0.70, 5.5)

## 8.Flight Timetable

function showFlightData(arr){

let type = arr[0];

let townName = arr[1];

let time = arr[2];

let flightNumber = arr[3];

let gateNumber = arr[4];

console.log(`${type}: Destination - ${townName}, Flight - ${flightNumber}, Time - ${time}, Gate - ${gateNumber}`);

}

showFlightData(['Departures', 'London', '22:45', 'BR117', '42'])

showFlightData(['Arrivals', 'Paris', '02:22', 'VD17', '3'])

## 9. Calorie Object

function showCaloriesFromFood(arr){

let caloryObject = {};

for (let i = 0; i < arr.length; i += 2){

let caloryObjectKey = arr[i];

let caloryObjectValue = arr[i + 1];

caloryObject[caloryObjectKey] = Number(caloryObjectValue);

}

console.log(caloryObject);

}

showCaloriesFromFood(['Yoghurt', 48, 'Rise', 138, 'Apple', 52])

showCaloriesFromFood(['Potato', 93, 'Skyr', 63, 'Cucumber', 18, 'Milk', 42])

## 10.\*Coffee Machine

function calculateDrinkPrices(orders){

let income = 0;

for(let i = 0; i < orders.length; i++){

let price = 0;

let order = orders[i].split(", ");

let money = Number(order[0]);

let drink = order[1];

if (drink === "tea"){

price = 0.80;

}

else if (drink === "coffee"){

if (order[2] === "caffeine"){

price = 0.80;

}

else if (order[2] === "decaf"){

price = 0.90;

}

}

if (order.includes("milk")){

price += 0.10;

//price += Math.round((price \* 0.1) \* 10) / 10;

}

let sugarCount = Number(order[order.length - 1]);

if (sugarCount > 0){

price += 0.10;

}

if (money >= price){

console.log(`You ordered ${drink}. Price: ${price.toFixed(2)}$ Change: ${(money - price).toFixed(2)}$`);

income += price;

}

else{

console.log(`Not enough money for ${drink}. Need ${(price - money).toFixed(2)}$ more.`);

}

}

console.log(`Income Report: ${income.toFixed(2)}$`);

}

calculateDrinkPrices(['1.00, coffee, caffeine, milk, 4', '0.40, tea, milk, 2', '1.00, coffee, decaf, 0'])

calculateDrinkPrices(['8.00, coffee, decaf, 4', '1.00, tea, 2'])