1. Write a function that capitalizes the first letter of each string argument it receives.

**Function arguments:** ['hello', 'there', 'ES', 6]

**Output:** ['Hello', 'There', 'ES']

1. Write a function that calculates sale tax that should be paid for the product of the given price. Use a constant to set a fixed value of the tax rate (i.e. 20% in Serbia).

**Input:** [{ name: “Banana”, price: 120 }, {name: “Orange”, price: 100}]

**Output:** [{ name: “Banana”, **price: 144** }, { name: “Orange”, **price: 120** }] *// product full price*

**Output2:** [ **24**, **20** ] *// tax only*

const products = [{ name: "Banana", price: 120.23, date: new Date() }, { name: "Orange", price: 100, date: new Date() }]

const productsWithTax = products.map(product => {

const { price: prodPrice } = product;

const newProd = {

...product,

price: Number.parseFloat(prodPrice),

priceWithTax: prodPrice \* 1.2,

tax: prodPrice \* 0.2

}

return newProd

})

console.log(products);

console.log(productsWithTax);

1. Write a function that increases each element of the given array by the given value. If the value is omitted, increase each element of the array by 1.

**Input:** [4, 6, 11, -9, 2.1], 2

**Output:** [6, 8, 13, -7, 4.1]

1. Write a function that filters all even elements of the array.

**Input:** [6, 11, 9, 0, 3]

**Output:** [6, 0]

1. Write a function that filters all the strings from the given array that contain substring *JS* or *js*.

**Input:** ['compiler', 'transpiler', 'babel.js', 'JS standard', 'linter']

**Output:** ['babel.js, 'JS standard']

1. Write a function that filters all integer numbers from the given array.

**Input:** [1.6, 11.34, 9.23, 7, 3.11, 8]

**Output:** [7, 8]

1. Write a function that filters all integer arguments that contain digit 5.

**Function arguments:** 23, 11.5, 9, 'abc', 45, 28, 553

**Output:** [45, 553]

1. Write a function that returns indexes of the elements greater than 10.

**Input:** [1.6, 11.34, 29.23, 7, 3.11, 18]

**Output:** 1, 2, 5

* 1. Create an array of persons. A person should be an object with *name* and *age* properties. Experiment with enhanced object literals.
  2. Write a function that prints out the names of persons older than 25.
  3. Write a function that check if there is a person older than 40.
  4. Write a function that checks if all persons are older than 20.

1. Write a function that checks if the given array is an array of positive integer values.

**Input:** [3, 11, 9, 5, 6]

**Output:** yes

**Input:** [3, -12, 4, 11]

**Output:** no

1. Write a function that calculates the product of the elements of the array.

**Input:** [2, 8, 3]

**Output:** 48

1. Write a function that calculates the maximum of the given array.

**Input:** [2, 7, 3, 8, 5.4]

**Output:** 8