МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Циклова комісія програмування***

**Практична робота №2**

**Робочий план**

# з курсу «Front-end-програмування на React JS»

Воробйова Валентина Юріївна

*ПІБ студента*

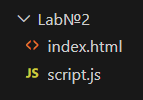
студента групи **1П-20**

|  |  |  |
| --- | --- | --- |
| **Вид роботи** | **Дата** | **Оцінка** |
| *Пром. контроль* |  |  |
| *Залік* |  |  |

Викладач   Сухенко А. С.

# Черкаси-2023

1. Створити папку для виконання лабораторної роботи. В папці створити файл index.html зі стандартним html документом. Створити файл script.js та підключити його до файлу index.html



1. Визначте як константу теперішній рік. Напишіть стрілкову функцію як буде приймаи 2 аргументи, імя та рік народження. Та повертати рядок «Я name, мені age років», де name та age наші змінні. Для виконання цього завдання використовуйте шаблонні рядки. Крім того має бути змога викликликати функцію предаючи в неї як обидва параметри, так і тільки name.

const year\_now = new Date().getFullYear();

let getAgeInfo = (name, birthYear=year\_now) => {

    if (birthYear === null) {

        return `Я ${name}.`;

    } else {

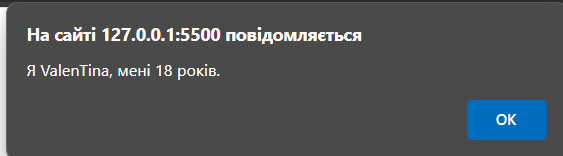
        let age = year\_now - birthYear;

        return `Я ${name}, мені ${age} років.`;

    }

};

alert(getAgeInfo("ValenTina", 2005));



1. Створіть класс Human який приймє, та зберігає такі праметри: name, age, weight, gender. Де name – ім’я, age – вік, weight – вага, gender – стать (man/woman)

class Human {

        constructor(name, age, weight, gender) {

            this.name = name;

            this.age = age;

            this.weight = weight;

            this.gender = gender;

        }

    }

1. Додайте, до класу Human метод getInfo(), який приймає параметр prop. Якщо prop передано зі значенням null, то повертаємо обєкт зі всією інформацією про людину, якщо назва конкретного параметру, значення тільки цього параметру. Для спрощення будемо вважати що передаватись будуть тільки назви існуючих параметрів або Null. Також додайте метод greeting який повертає рядко “Hi!”

class Human {

        constructor(name, age, weight, gender) {

            this.name = name;

            this.age = age;

            this.weight = weight;

            this.gender = gender;

        }

        getInfo(prop) {

            if (prop === null) {

                let info = `Name: ${this.name}\nAge: ${this.age}\nWeight: ${this.weight}\nGender: ${this.gender}`;

                return info;

            } else if (Object.keys(this).includes(prop)) {

                return this[prop];

            } else {

                return `Invalid property: ${prop}`;

            }

        }

        greeting() {

            return "Hi!";

        }

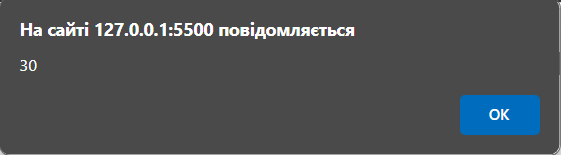
    }

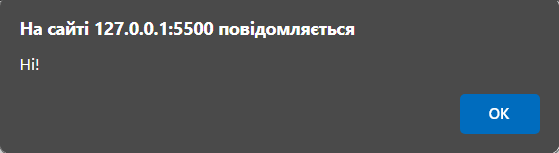
    const person = new Human("John", 30, 70, "Male");

    alert(person.getInfo('age'));

    alert(person.greeting());

;





1. Створіть Класи Man та Woman які наслідують клас Human, проте приймають лише 3 аргументи name, age та weight. Кожен клас має реалізовувати метод greeting. У чоловіка він повертає рядок, де він вітаєтся, вказує своє імя та вік. Жінка лише вітаєтся та вкзує імя

class Man extends Human {

    constructor(name, age, weight) {

        super(name, age, weight, "Male");

    }

    greeting() {

        return `Hi! My name is ${this.name}, and I am ${this.age} years old.`;

    }

}

class Woman extends Human {

    constructor(name, age, weight) {

        super(name, age, weight, "Female");

    }

    greeting() {

        return `Hi! My name is ${this.name}.`;

    }

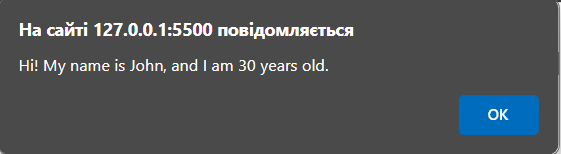
}

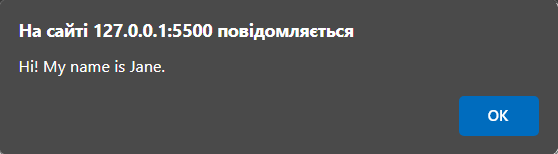
const john = new Man("John", 30, 70);

const jane = new Woman("Jane", 25, 60);

alert(john.greeting(), john.getInfo());

alert(jane.greeting(), jane.getInfo('name'));





1. Створіть масив довжиною 15 елементів, кожен з яких це екземпляр класу Man або Woman.

function getRandomInt(min, max) {

    return Math.floor(Math.random() \* (max - min + 1)) + min;

}

const people = [];

for (let i = 0; i < 15; i++) {

    const randomName = `Person${i + 1}`;

    const randomAge = getRandomInt(20, 40);

    const randomWeight = getRandomInt(50, 90);

    const randomGender = getRandomInt(0, 1) === 0 ? "Male" : "Female";

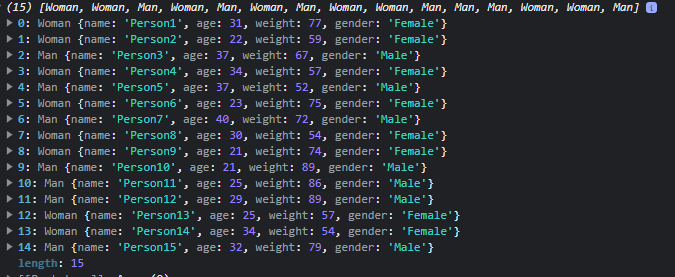
    const person = randomGender === "Male"

        ? new Man(randomName, randomAge, randomWeight)

        : new Woman(randomName, randomAge, randomWeight);

    people.push(person);

}



7. Створіть функцію пошуку яка повертає Promise. Функція приймає аргумент list – масив з попереднього завдання. Aргумент searchField – назва параметру за яким буде відбуватись пошук. Аргумент searchValue – значення параметру за яким буде відбуватись пошук. Затримку на роботу функції емулюйте через setTimeout, час роботи якого визначаєтся окремо заданою константою. Сам пошук можна реалізувати через метод масиву [filter](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter), проте Ви можете за бажанням реалізувати власний алгоритм пошуку. В результаті роботи функції в then маєть приходити усі результати пошуку, якшо таких немає в catch повинно повертатись повідомлення, що за пошуком по searchField:searchValue не знайдено.

const searchPeople = (list, searchField, searchValue) => {

const delay = 1000;

return new Promise((resolve, reject) => {

setTimeout(() => {

const results = list.filter(person => person[searchField] === searchValue);

if (results.length > 0) {

resolve(results);

} else {

reject(`За пошуком по ${searchField}:${searchValue} не знайдено.`);

}

}, delay);

});

};

searchPeople(people, "name", "Person7")

.then(results => {

console.log("Результати пошуку:");

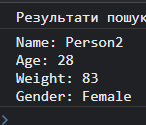
for (const person of results) {

console.log(person.getInfo(null));

}

})

.catch(errorMessage => console.log(errorMessage));



8. Розділіть код на модулі. Кожен клас повине бути окремим модулем. Функція пошуку повинна бути реалізована в окремому модулі. Створення масиву елементів – окремий модуль. Створіть точку входу main.js, в яку імпортуйте функію пошуку та Ваш масив. Додайте кілька викликів Вашої функції. Перед кожним викликом виведить в консоль з якими параметрами пошуку вона викликаєтся. Якщо є результати пошуку, то для кожного такого елементу виведіть в консоль результати роботи методів greeting та та getInfo(‘gender’). Обовязково має бути приклад успішного та не успішного пошуку.