1. Make a Rectangle class that stores a width and a height. Make a few instances and print out the properties. Modify a few of the properties and print out the results again.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 1</title>

</head>

<body>

</body>

<script>

    class Rectangle {

        constructor(height, width) {

            this.height = height;

            this.width = width;

        }

    }

    const obj = new Rectangle(13, 10);

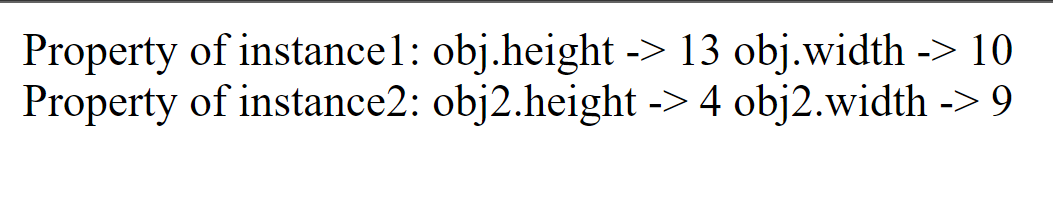
    const obj2 = new Rectangle(4, 9);

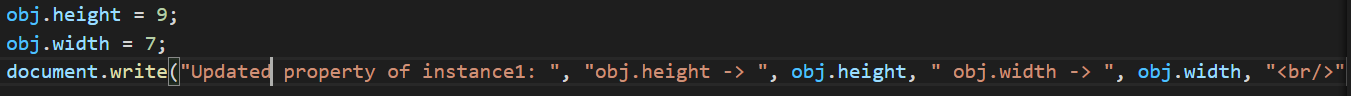
    document.write("Property of instance1: ", "obj.height -> ", obj.height, " obj.width -> ", obj.width, "<br/>");

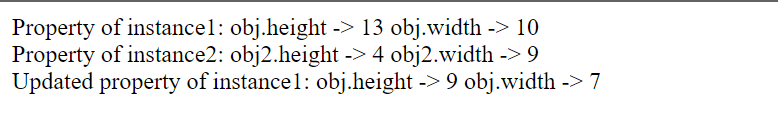
    document.write("Property of instance2: ", "obj2.height -> ", obj2.height, " obj2.width -> ", obj2.width);

</script>

</html>







2. Add a getArea method. Use the prototype property

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

<script>

    class Rectangle {

        constructor(height, width) {

            this.height = height;

            this.width = width;

        }

    }

    Rectangle.prototype.area = function area() {

        return this.height \* this.width;

    }

    const obj = new Rectangle(13, 10);

    const obj2 = new Rectangle(4, 9);

    document.write("Property of instance1: ", "obj.height -> ", obj.height, " obj.width -> ", obj.width, "<br/>");

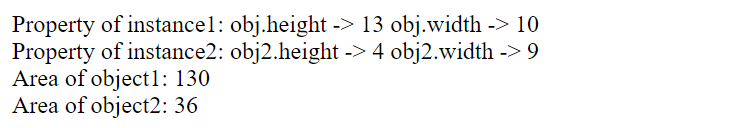
    document.write("Property of instance2: ", "obj2.height -> ", obj2.height, " obj2.width -> ", obj2.width, "<br/>");

    document.write("Area of object1: ", obj.area(), "<br/>");

    document.write("Area of object2: ", obj2.area());

</script>

</html>



3. Assuming that the Rectangle constructor takes a width and a height, why does the following output 20 instead of 200? (Hint: if you see an answer that seems too obvious to be what I am looking for, it probably is the answer I am looking for.)

Rectangle r = new Rectangle(4, 5);

r.hieght = 50;

r.getArea(); --> 20 // Not 200

4. Make a variable whose value is an object with firstName and lastName properties, but don’t define a Person class first. Try looking up the first and last names. Try changing the last name. It seems very odd to Java programmers to make an object without first defining a class, but JavaScript programmers do this sort of thing all the time.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 4</title>

</head>

<body>

</body>

<script>

    const person = {

        firstname: "Pradumnya",

        lastname: "Ghadole"

    }

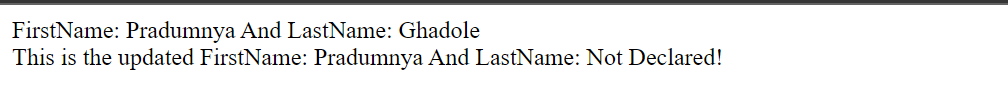
    document.write("FirstName: ", person.firstname, " And LastName: ", person.lastname, "<br/>")

    person.lastname = "Not Declared!";

    document.write("This is the updated FirstName: ", person.firstname, " And LastName: ", person.lastname, "<br/>")

</script>

</html>



5. Try reading the middleName property from your variable above. Try assigning to the middleName property. Try reading the property again after you assign to it. Is this behavior a good thing or a bad thing?

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 4</title>

</head>

<body>

</body>

<script>

    const person = {

        firstname: "Steve",

        lastname: "Smith",

        middlename: "Unstopable"

    }

    document.write("FirstName: ", person.firstname, "<br/>MiddleName: ", person.middlename, "<br/> And LastName: ", person.lastname, "<br/>")

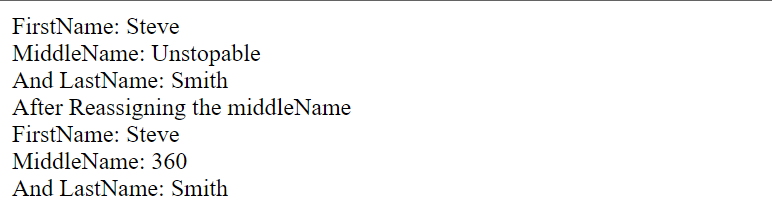
    person.middlename = "360"

    document.write("After Reassigning the middleName<br/>")

    document.write("FirstName: ", person.firstname, "<br/>MiddleName: ", person.middlename, "<br/> And LastName: ", person.lastname, "<br/>")

</script>

</html>



6. Create a string that contains what looks like an object with firstName and lastName properties. Use “eval” to turn it into a real object, and test it the same way you did with the previous object that you created directly.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 6</title>

</head>

<body>

    <p id="p1"></p>

</body>

<script>

    var str = '({"firstName":"Bill","lastName":"Gates"})';

    var obj = eval(str);

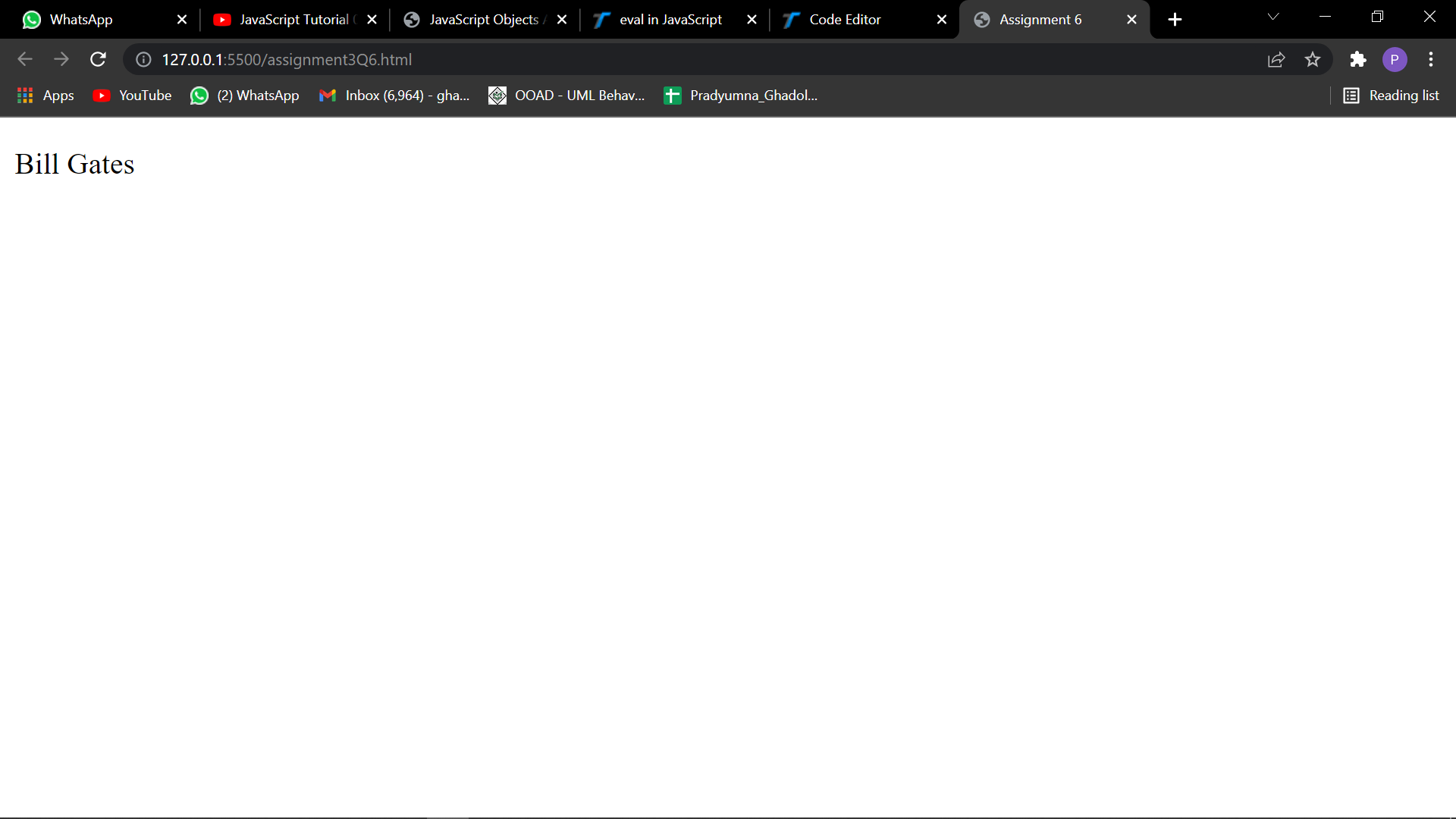
    document.getElementById("p1").innerHTML = obj.firstName + " " + obj.lastName;

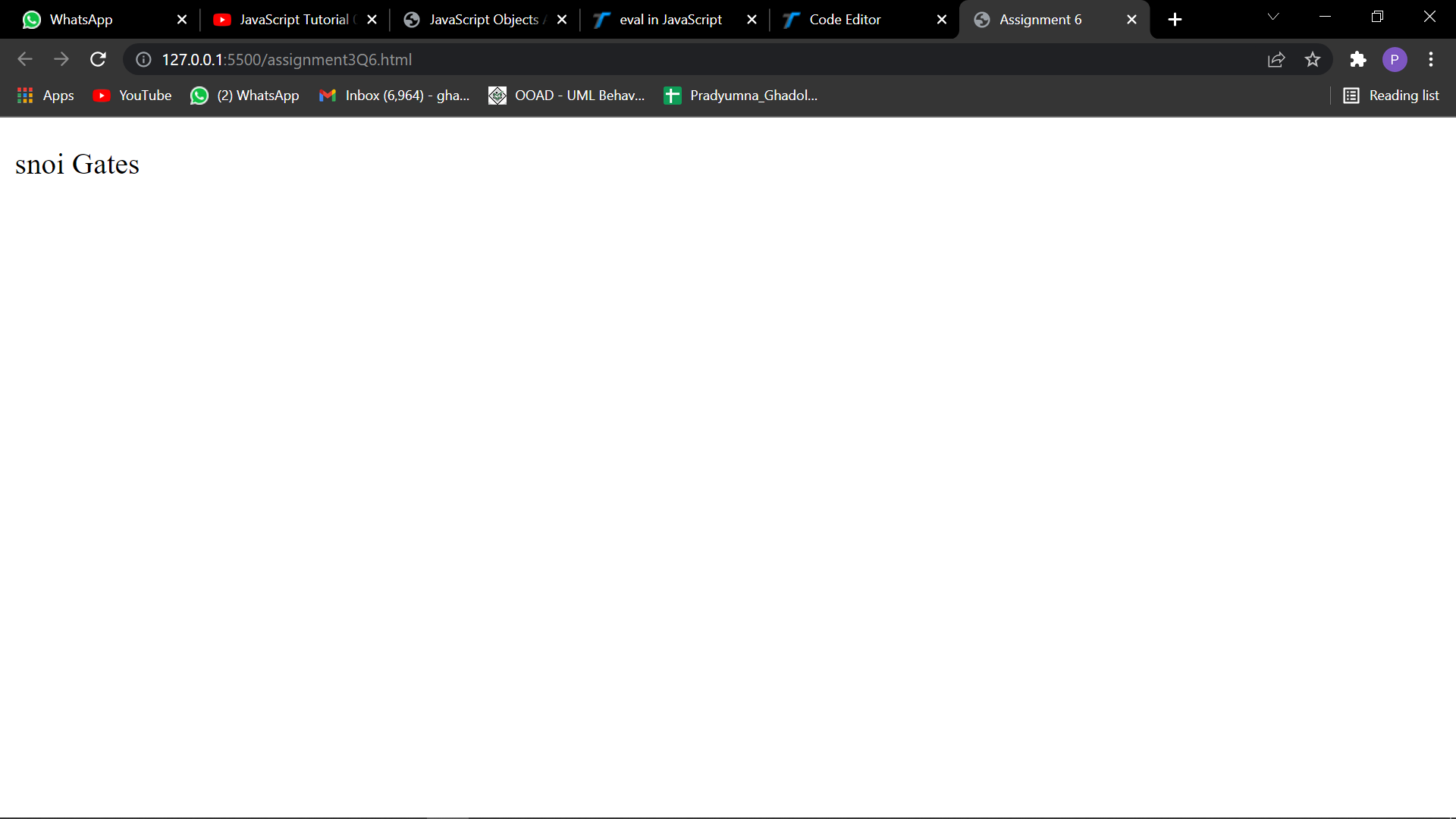
    obj.firstName = "snoi";

    document.getElementById("p1").innerHTML = obj.firstName + " " + obj.lastName;

</script>

</html>





7. Do the same with JSON.parse. You have to follow strict JSON rules in this case.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 7</title>

</head>

<body>

    <p id="p1"></p>

</body>

<script>

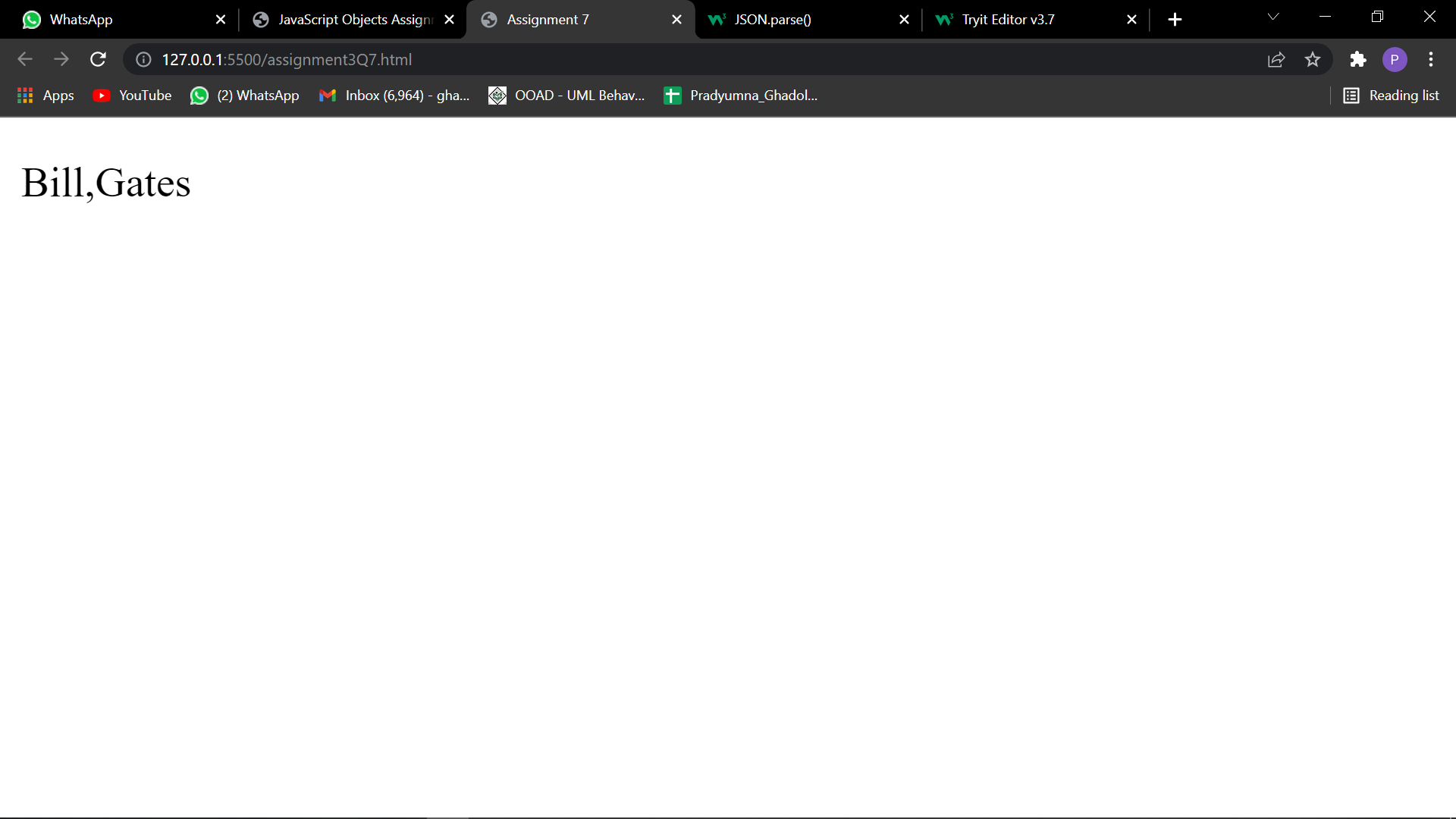
    var str = '{"firstName":"Bill","lastName":"Gates"}';

    var obj = JSON.parse(str);

    document.getElementById("p1").innerHTML = obj.firstName + "," + obj.lastName;

</script>

</html>



1. Write a JS program to create object of person with fields as follows:- fname - string lname - string age - int skills - array address - object city - string pincode - int dateOfBirth - Date married - Boolean profession – string

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 8</title>

</head>

<body>

</body>

<script>

    function person(fname, lname, age, skills, dateofbirth, address, married, profession) {

        this.fname = fname;

        this.lname = lname;

        this.age = age;

        this.skills = skills;

        this.dateofbirth = dateofbirth;

        this.address = address;

        this.married = married;

        this.profession = profession;

    }

    person1 = new

        person("nikhil", "goud", 22, ["c"], "24/10/1996", { city: "hyderabad", pincode: "521185" }, "false", "sr analyst")

    person2 = new

        person("harish", "chinna", 21, "HTML", "08/06/1997", { city: "Ameerpet", pincode: "500038" }, "false", "jr analyst")

    print = function () {

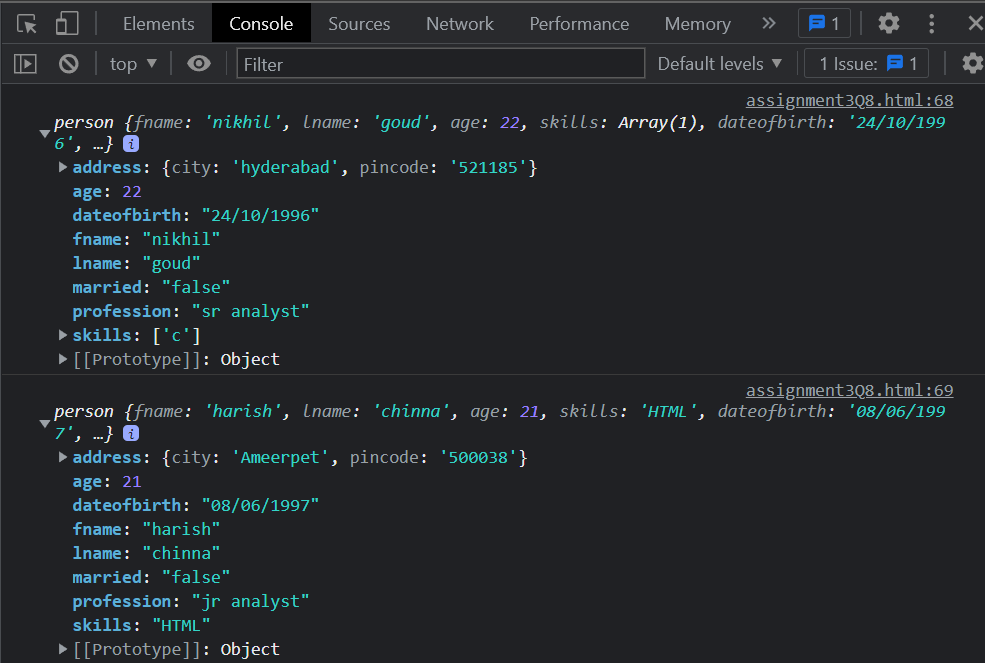
        console.log(person1);

        console.log(person2);

    }();

</script>

</html>



2. Modify the above program to create 2 objects, amitabh and abhishek, here abhishek has some common properties from amitabh, try to use it such common properties from amitabh instead of creating it in abhishek.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Assignment 8</title>

</head>

<body>

</body>

<script>

    function person(fname, lname, age, skills, dateofbirth, address, married, profession) {

        this.fname = fname;

        this.lname = lname;

        this.age = age;

        this.skills = skills;

        this.dateofbirth = dateofbirth;

        this.address = address;

        this.married = married;

        this.profession = profession;

    }

    person1 = new

        person("nikhil", "goud", 22, ["c"], "24/10/1996", { city: "hyderabad", pincode: "521185" }, "false", "sr analyst")

    person2 = new

        person("harish", "chinna", 21, "HTML", "08/06/1997", { city: "Ameerpet", pincode: "500038" }, "false", "jr analyst")

    amitabh = new

        person("amitabh", "bachan", 76, "acting", "24/10/1996", { city: "Mumbai", pincode: "897342" }, "true", "Actor");

    abhishekh = new

        person("abhishekh", 52, ["nothing"], "24/09/78", "false", "actor");

    var abhishekh = Object.create(amitabh);

    print = function () {

        console.log(person1);

        console.log(person2);

        console.log(amitabh);

        console.log(abhishekh);

    }();

</script>

</html>

3. Modify the above code to create third object as "Aaradhya", this object shares the common properties from amitabh as well as abhishek accordingly demostrate on your own.