**1. JS-GENERAL-T-BASICS-1. Explain the following JavaScript code clips in your own words. Tell what is the value of variable currentPort in part e. What does the part f print on the console? (2 pts):**

a.      “intomieli”                                                                     (0,25 points)

This is a string variable.

b.      ==                                                                                 (0,25 points)

This means same data but not same type.

c.      ===                                                                               (0,25 points)

This means data and type are same.

d.      “Se” + “ “ + “on” + “ “ + “totta” + “.”;                           (0,25 points)

All these strings are concatenated with + arithmetic operator.

e.      var port = 3001; var currentPort = port || 3000;           (0,5 points)

The currentPort variable is assigned the value of port if port has a true value. If port is falsy (e.g., undefined, null, 0, false, etc.), then currentPort is assigned the value of 3000.

In this case, since port is set to 3001 (which is a truthful value), the currentPort will take the value of port. Therefore, the value of currentPort will be 3001.

f.       var saldo = 0;  
           if (saldo) {  
               console.log(“Rahaa on”);  
          } else {  
               console.log(“Rahaa ei ole”);  
          }                                                                                     (0,5 points)

 The condition in the if statement is checking the truth of the balance. In JavaScript, a number 0 is considered false. Therefore, the code inside the else block will be executed.

So, the output on the Console will be:

**2. JS-GENERAL-T-BASICS-2. Look at the following code clip. Does it work? If not, why, and how could you fix it? (2 pts):**

let number = 5;  
const cNumber = 5;  
console.log(number \* cNumber);  
cNumber = 6;  
console.log(number \* cNumber);

 const variable is unchangeable. If we use let instead of const then it will work.

**3. JS-GENERAL-P-BASICS-1. Write the following programs. Use loops. Include source code and screenshots from** **results (2 pts):**

**a. Output even numbers from 20 to 200 to console. (1 point)**

let number = 20;

while (number <= 198) {

    console.log(number);

    number = number + 2;

}

A screenshot of a computer

Description automatically generated

**b. Sum items from array deposits while the sum is less than 100. (Add the following deposit to sum if the sum of deposits is before addition less than 100.) Here on possible example of an array: deposits = [ 12.11, 34.22, 31.95. 50.00, 22.50, 200.00 ] (1 point)**

let deposits = [12.11, 34.22, 31.95, 50.00, 200.00];

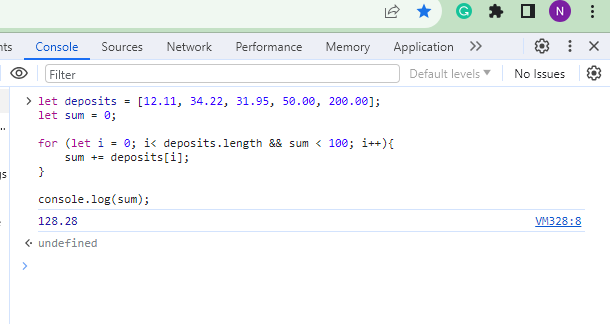
let sum = 0;

for (let i = 0; i< deposits.length && sum < 100; i++){

    sum += deposits[i];

}

console.log(sum);



**4. JS-GENERAL-P-BASICS-2. Make exercise “FizzBuzz” from book (Exercises, chapter 2).  (2 pts)**

for (let i = 0; i <= 100; i++) {

    if (i % 3 === 0 && i % 5 === 0) {

        console.log("FizzBuzz");

    }  else if (i % 3 === 0){

        console.log("Fizz");

    } else if (i % 5 === 0){

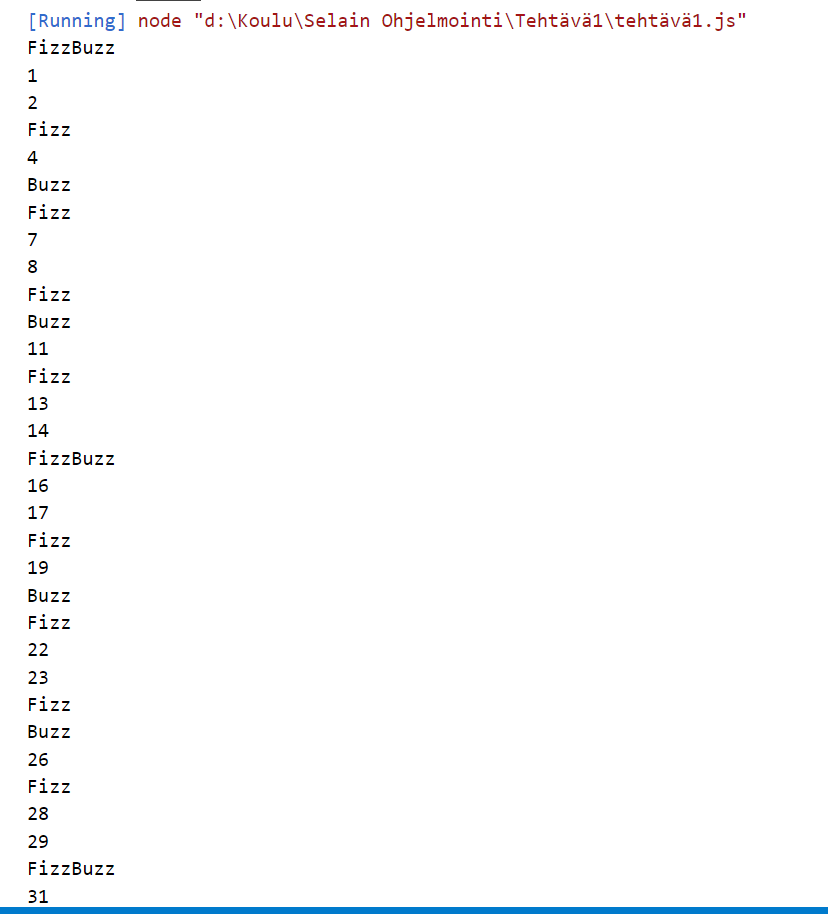
        console.log("Buzz");

    } else {

        console.log(i);

    }

}



**5. JS-ARRAY-T-BASICS-2 (1 point)**

**a. Can the same array contain numbers and objects at the same time?**

Yes, the same array in JavaScript can contain a mix of numbers, objects, or any other data types. JavaScript arrays are dynamic and can hold elements of different types.

let mixedArray = [1, "apple", { key: "value" }, true, 3.14];

**b. Give one example of a method of an Array object that modifies the array in place, and one example of a method that modifies a copy (returns a modified copy of the array).**

let array1 = [1, 2, 3];

array1.push(4);

console.log(array1);

A screenshot of a computer code

Description automatically generated

**c. When checking the length of an array, are you utilizing an attribute of an array or a method of an array?**

When checking the length of an array, you are using an attribute of an array. The length property is not a method but an attribute that provides the number of elements in an array.

let myArray = [1, 2, 3, 4, 5];

console.log(myArray.length);

A screenshot of a computer code

Description automatically generated

**6. JS-ARRAY-P-ARRAY-PROPERTIES-1 (2 points)**

Lets have an array distances = [ 165, 626, 148, 12, 81, 181 ].

1. **Write a code clip that returns the length of the array. (0,5 points)**

let myArray = [1, 2, 3, 4, 5];

let arrayLength = myArray.length;

console.log("Array length:", arrayLength);

A screenshot of a computer code

Description automatically generated

1. **Write code clip that adds the distances 8, 533 and 76 into the end of the array and in this order. One distance at a time. Use one the array methods. (0,5 points)**

let myArray = [1, 2, 3, 4, 5];

myArray.push(8, 533, 76);

console.log("Updated array:", myArray);

A screenshot of a computer code

Description automatically generated

**c. Write a code clip that removes the number 148 from the array. Use array methods. A tip: One of the methods to use could be splice. (1 points)**

let myArray = [1, 2, 3, 4, 148, 5];

let indexOf148 = myArray.indexOf(148);

if (indexOf148 !== -1) {

myArray.splice(indexOf148, 1);

console.log("Array after removing 148:", myArray);

} else {

console.log("Number 148 not found in the array.");

}.                   
  
A screenshot of a computer code

Description automatically generated