Project: Calculate total size of files in a directory

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculate Folder Size</title>
</head>
```

(HTML Head)

```html
<body>
    <h2>Select a Folder</h2>
    <!-- choose file -->
    <input type="file" id="folderInput" webkitdirectory multiple> <br><br>
    <!-- output -->
    <textarea name="" id="output" rows="15" cols="80" readonly></textarea>
```

In the body has some tages:

- <input>: Users can choose folder from it
  - Type = file: This type of input tag is using for file
  - id=folderInput: this id is unique and using it in JavaScript
  - webkitdirectory: This is an attribute that allow user to select folders
  - multiple: This attribute is allowed user to select multiple time at once
- <textarea>: It will show the result after calculated and it has read only. So, the users cannot access it.

```javascript
<script>
        function formatBytes(bytes){
            const sizes = ['Bytes', 'KB', 'MB', 'GB'];
            if (bytes ===0 )
                return "0 Byte";
            //find index of size by loggarith
            const i = Math.floor(Math.log(bytes) / Math.log(1024));
            //div bytes and end it with expression
            return (bytes / Math.pow(1024,i)).toFixed(2) + ' ' + sizes[i];
        }
```

HTML is not a programming language. So, I use JavaScript.

- Function formatBytes: Is a function is using in the calculate of bytes such as bytes, kilobytes, megabytes, gigabytes. It has a parameter that store the value of folder's size.
- Array sizes: It will show the name of memory's scales.
- Variable i: It contain the value of logarithm of folder's size and 1024 (1 kilobyte = 1024 bytes). The result of it will be an index in array sizes to choose which value's name that will show.

- Finally, here's the result of it, if folder's size equal to Zero, it will show 0 byte, but if folder's size different from Zero it will return a value.
- The value that will return is from the division between folder's size and scale's size.
    - Method Math.pow(base, exponent): This method stands for self-multiplication
    - .toFixed(): This method limits numbers of decimal place

```
document.getElementById("folderInput").addEventListener("change",function(event){
    const files = event.target.files;
    let totalSize = 0;
    let counts = {};
```

When the textbox "folderInput" got change, it will start an anonymous function with an parameter. That function has stored:

- Variable files: It stores the selected file
- Variable totalSize: It will use for total size of all files in the folder
- variable counts: It will use for count which file has stored how many file.

```
for(let file of files){
    //sum all size of file
    totalSize += file.size;

    //add type of files to textarea
    let ext = file.name.split('.').pop().toLowerCase();

    //count type of files
    if(counts[ext]){
        counts[ext]++;
    }else{
        counts[ext]=1;
    }
}
```

For loop allows to calculate size again and again as variable files is condition. And then the variable totalSize will calculate again as many files has access.

- Variable ext: stands for extension of file with dot in front and it must be lower case that done it by the method .toLowerCase()
- The control statement of counts: if the first file access in this statement, it will not find in the first statement, so it will go to else statement and count it 1. If that type of file access for the second time, it will true and count itself plus one again and it will two.

```
let result = `Total files selected: ${files.length} files \n`;
result += `Total size: ${formatBytes(totalSize)}\n\n `;
result += `File count by Extensions: `+"\n";
```

Output of Total Size:

- Variable result: It stores the total of files, the result that return from the function formatBytes()

```
for(let ext in counts){
    result += `.${ext}: ${counts[ext]} files\n`
}
```

Output of extension:

Variable ext that contain the name of extension process in for loop as variable counts as a condition. In this loop the variable result will store the value of how many times which type of extension access and it will show the name of extension with dot in front.

```
    document.getElementById("output").value = result;
});
```

Finally, The values that stored in variable result will show up and change the data in textarea that has an id output.