Splitting Binary String into Substrings with Equal 0's and 1's

```
let str = "0100110101"
let str1 = "0111100010"
let str2 = "0000000000"
let str3 = "01111100011010"
let str4 = "001110010"

function checkConsecutivePairs(str) {
    let arr = str.split("");
    let maxSubArray=0;
    let count=0;
    let flag=arr[0];
    for(let i = 0; i < arr.length; i++) {
        if(arr[i] === flag ){
            count++
        }else{
            --count;
            if(count<=0){
                maxSubArray++;
                count = 0;
                flag=arr[i+1];
            }else if(count>0  && arr[i] !==arr[i+1]){
                maxSubArray++;
                count = 0;
                flag=arr[i+1];
            }

        }
    }
    // let count0 = 0;
    // let count1 = 0;
    // for(let i = 0; i < arr.length; i++) {
    //     if(arr[i] === '0'){
    //         count0++
    //     }else{
    //         count1++
    //     }
    //     if(count0 === count1){
    //         maxSubArray++
    //     }
```

Making Binary String Alternate with Minimum Flips

```
let s = "001"
let s1 = "0001010111"
let s2 = "100"
let s3 = '0101'
let s4 = '000001111111100000000000'
function c(str){
    let arr = str.split("")
    let count = 0
    let i = 1
    while(i <arr.length ){
        if(arr[i] === arr[i - 1]){
            console.log(arr[i])
            console.log(arr[i + 1])
            console.log('<===>')
            count++
            i = i + 2
        }
        else{
            i = i + 1
        }

    }
    return count
}

// console.log(c(s))
//console.log(c(s1))
//console.log(c(s2))
//console.log(c(s3))
console.log(c(s4))
```