

# Algorithms

---

## ===== Arrays =====

- нужно перебрать массив потом вложенный массив
1. отбить основной цикл и внутренний разделителями

```
function uniteUnique(arr) {
  arrOfArg.map((arr, id) => {
    console.log("++++++")
    console.log("-----arr-----", arr)

    arr.map((el) => {
      console.log("=====el=====", el)
    });
  });
}
uniteUnique([1, 3, 2, 2, 2], [5, 2, 1, 4]);
```

## ===== Strings =====

### Compare strings

1. find the missing letter from a string and return it

```
// fiend uniq literals form alphabet
function fearNotLetter(str) {
  let currCharCode = str.charCodeAt(0);
  let missing = undefined;

  str.split("").forEach((letter) => {
    if (letter.charCodeAt(0) === currCharCode) {
      currCharCode++;
    } else {
      missing = String.fromCharCode(currCharCode);
    }
  });

  return missing;
}
```

2. convert the characters &, <, "", " to name code Алгоритм замены ряда символов на другие

1. создать хранилище ключ значение
2. найти символ в строке
3. создать новую строку с новым значением

Решение один:

1. разбить строку на массив элементов `str.split("")`
2. создать массив с characters `&, <, "", "`
3. перебрать массив
4. найти элемент и заменить на нужный
5. собрать массив в кучу

Решение два:

1. создать объект ключ characters значение на что надо заменить
2. создать регулярное выражение и заменить по ключу

```
function convertHTML(str) {
  const characterEntries = {
    "&": "&amp;",
    "<": "&lt;",
    ">": "&gt;",
    "'": "&quot;",
    "'": "&apos;"
  }
  return str.replace(/([<>\'"])/g, match => characterEntries[match]);
}
```

## ===== Numbers =====

### 1. Sum All Odd Fibonacci Numbers

- нужно посчитать сумму фибоначчи чисел 0,1,1,2,3,5,8,13,21,34,55 Алгоритм чисел фибоначчи:
  1. получить предыдущее значение;
  2. получить текущее значение;
  3.  $(0) + (1) = (1) \Rightarrow 1 + 1 = (2) \Rightarrow 1 + 2 = (3)$
  4. сложить предыдущее с текущее  $\rightarrow (2)+(3)=5$

```
function fibonacciShort(num){
  let a = 1,
      b = 1;

  for (let i = 3; i <= num; i++) {
    [a,b] = [b, a + b];
  }
  return b;
}

function sumFibs(num) {
  let prev = 0;
  let current = 1;
```

```

    let fibonacci = 0;

    while(current <= num){
        if(current % 2 !== 0 ) {
            fibonacci += current;
        }

        current += prev
        prev = current - prev;
    }

    return fibonacci;
}

function fibonacci(num){
    const result = [0,1];
    for (let i = 2; i < num; i++) {
        const prevNum1 = result[i-1];
        const prevNum2 = result[i-2];
        result.push(prevNum1 + prevNum2);
    }

    return result[num]
}

```

## 2. Sum all Primes Short info:

- простое число это натуральное число, единственным делителями которого являются только оно само и единица
- ▪ 2 это простое число которое делиться на 1 и 2
- ▪ 4 это не простое число которое делиться на 1,2,4
- ▪ натуральное число 1 не являются ни простым ни составным
- разложением на простые множители - это если в натуральном числе все множители простые числа Теорема Каждое натуральное число, отличное от 1, может быть разложено на простые множители, и притом единственным образом (если отождествлять разложения  $pq$  и  $qp$ , где  $p$  и  $q$  простые числа)

Решето Эратосфена применимо к решению этой задачи.

Условие: Нужно что бы возвращать сумму всех простых чисел который меньше либо равны входному числу:

Алгоритм решения:

1. разложить число начиная с 2 ...
2. исключение составных чисел (которые имеют делители отличные от 1 и самого числа)
3. суммируем все оставшиеся простые числа

```

function sumPrimes(num) {
    let primes = [];

```

```
    for (let i = 2; i <= num; i++) {  
      if (primes.every((prime) => i % prime !== 0)) {  
        primes.push(i);  
      }  
    }  
  
    return primes.reduce((sum, prime) => sum + prime, 0)  
  }  
}
```