

## Phần 2: Toán tử, câu lệnh điều kiện, vòng lặp, function, HOF, arrow function, call(), apply(), bind() trong Javascript

---

### 1. Dự đoán kết quả

#### 1.1. Dự đoán kết quả liên quan đến toán tử

```
let a = 1
let b = 2
let c = a + b++
console.log(a, b, c)
```

```
let a = 2
console.log(--a)
console.log(a)
```

```
let a = 1
let b = '2'
let c = a + b
console.log(c)
```

```
let a = 1
a += 2
console.log(a)
```

```
let a = 1
let b = 2
a && b = 3
console.log(b)
```

#### 1.2. Kiểm tra truthy và falsy

```
true && false
true || false
!true && false
!''
```

```
!!0  
!!{ }  
!![ ]  
!null || false
```

## 2. Bài tập luyện thuật toán

1. Viết hàm nhận vào số nguyên dương  $n$  và return lại một mảng các ước số của nó
2. Viết hàm nhận vào số nguyên dương  $n$  và return lại tổng tất cả ước số nguyên dương của nó
3. Viết hàm kiểm tra số nguyên dương  $n$  có phải số nguyên tố hay không
4. Viết hàm tính tổng các chữ số của số nguyên dương  $n$
5. Viết chương trình nhập tháng, năm. Hãy cho biết tháng đó có bao nhiêu ngày. Biết rằng năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100.
6. Viết hàm in bảng cửu chương ra màn hình
7. Tìm số fibonacci thứ  $n$ . Cụ thể, các số đầu tiên của dãy Fibonacci là 1, 1, 2, 3, 5, 8, 13...
8. Tính tổng  $n$  số fibonacci đầu tiên

## 3. Đáp án bài tập luyện thuật toán

1. Viết hàm nhận vào số nguyên dương  $n$  và return lại một mảng các ước số của nó

```
const uocso = (n) => {  
  const result = []  
  let i = 1  
  while (i <= n) {  
    if (n % i === 0) {  
      result.push(i)  
    }  
    i++  
  }  
  return result  
}
```

2. Viết hàm nhận vào số nguyên dương  $n$  và return lại tổng tất cả ước số nguyên dương của nó

```
const tongUocSo = (n) => {  
  let result = 0  
  let i = 1  
  while (i <= n) {  
    if (n % i === 0) {  
      result += i  
    }  
    i++  
  }  
  return result  
}
```

3. Viết hàm kiểm tra số nguyên dương n có phải số nguyên tố hay không

```
const kiemTraSoNguyenTo = (n) => {  
  if (n < 2) {  
    return false  
  }  
  for (let i = 2; i <= n / 2; i++) {  
    if (n % i === 0) {  
      return false  
    }  
  }  
  return true  
}
```

4. Viết hàm tính tổng các chữ số của số nguyên dương n

```
const tinhTong = (n) => {  
  let result = 0  
  const string = String(n)  
  for (const char of string) {  
    result += Number(char)  
  }  
  return result  
}
```

5. Viết chương trình nhập tháng, năm. Hãy cho biết tháng đó có bao nhiêu ngày. Biết rằng năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100.

```
const demNgay = (thang, nam) => {  
  const namNhuan = (nam) => {  
    if (nam % 400 === 0) {  
      return true  
    }  
    if (nam % 4 === 0 && nam % 100 !== 0) {  
      return true  
    }  
    return false  
  }  
  if ([1, 3, 5, 7, 8, 10, 12].includes(thang)) {  
    return 31  
  } else if ([4, 6, 9, 11].includes(thang)) {  
    return 30  
  } else {  
    if (namNhuan(nam)) {  
      return 29  
    }  
    return 28  
  }  
}
```

```
}  
}
```

## 6. Viết hàm in bảng cửu chương ra màn hình

```
const inBanCuuChuong = () => {  
  let result = ''  
  for (let i = 1; i <= 9; i++) {  
    if (i === 1) {  
      result += `Bảng số ${i}`  
    } else {  
      result += `\nBảng số ${i}`  
    }  
    for (j = 1; j <= 10; j++) {  
      result += `\n${i} x ${j} = ${i * j}`  
    }  
  }  
  return result  
}
```

## 7. Tìm số fibonacci thứ n

```
const fibonacci = (n) => {  
  if (n === 1 || n === 2) {  
    return 1  
  }  
  return fibonacci(n - 1) + fibonacci(n - 2)  
}
```

## 8. Tính tổng n số fibonacci đầu tiên

```
const fibonacci = (n) => {  
  if (n === 1 || n === 2) {  
    return 1  
  }  
  return fibonacci(n - 1) + fibonacci(n - 2)  
}  
const sumFibo = (n) => {  
  let sum = 0  
  for (let i = 1; i <= n; i++) {  
    sum += fibonacci(i)  
  }  
  return sum  
}
```