

# Phần 1: Kiểu dữ liệu, tham trị & tham chiếu, khai báo biến, scope và hoisting trong Javascript

---

## 1. Dự đoán

### 1.1 Các khai báo biến nào dưới đây là hợp lệ

```
let a = 1
let 1b = 2
let class = 3
let $c = 1
let _d = 1
let +f = 1
let K = 1
```

### 1.2 Dự đoán kết quả về khai báo, tham chiếu/tham trị, falsy/truthy

```
console.log(null === undefined)
```

```
console.log(null == undefined)
```

```
let a
console.log(a)
```

```
function sum(a, b) {
  let result = a + b
}
console.log(sum(1, 2))
```

```
let car = { name: "BMW" }
console.log(car.color)
```

```
let student = "dev"
console.log(student.name)
```

```
let x = 1
let y = x
y = 2
console.log(x)
```

```
let song1 = { name: "Em cua ngay hom qua" }
let song2 = song1
song2 = { name: "Con mua ngang qua" }
console.log(song1)
```

```
const list = ["MTP", "Jack", "KICM"]
list[1] = "Duc Phuc"
console.log(list)
```

```
let obj1 = { profile: { name: "Duoc", age: 24 } }
let obj2 = obj1
obj2.profile.age = 25
console.log(obj1)
```

```
let obj3 = { profile: { name: "Duoc", age: 24 } }
function handle(value) {
  value.profile.name = "xdev"
}
handle(obj3)
console.log(obj3)
```

```
if (1) {
  // có chạy không
}
if (2021) {
  // có chạy không
}
if (0) {
  // có chạy không
}
if ("") {
  // có chạy không
}
if (" ") {
  // có chạy không
}
```

```
if ("xdev") {  
  // có chạy không  
}  
if (true) {  
  // có chạy không  
}  
if (null) {  
  // có chạy không  
}  
if ({} ) {  
  // có chạy không  
}  
if ([]) {  
  // có chạy không  
}  
if ([0]) {  
  // có chạy không  
}
```

```
const keyboard = "a"  
keyboard = "b" // có lỗi hay không
```

```
let computer = "Samsung"  
let computer = "LG" // có lỗi hay không (strict mode và non strict mode)
```

### 1.3 Tại sao nên dùng let và const thay cho var

Để tránh tính trạng **hoisting** làm code không chặt chẽ.

### 1.4 Sự khác nhau giữa const và let

- **const** dùng để khai báo hằng số
- **let** dùng để khai báo biến có thể gán lại được

### 1.5 Dự đoán kết quả về hoisting

```
console.log(click())  
function click() {  
  return 1  
}
```

```
console.log(handle())  
const handle = function handle() {
```

```
    return 2
  }
```

```
console.log(handle())
const handle = function handle() {
  return 2
}
```

```
console.log(handle())
var handle = function handle() {
  return 2
}
```

```
console.log(a)
var a = 2
```

```
console.log(b)
let b = 2
```

## 1.6 Dự đoán kết quả về scope

```
if (true) {
  var age = 24
}
console.log(age)
```

```
if (true) {
  let a = 1
}
console.log(a)
```

```
function handle() {
  var a = 1
  let b = 2
  const c = 3
}
handle()
console.log(a)
```

```
console.log(b)
console.log(c)
```

```
let a = 1
if (true) {
  let a = 2
}
console.log(a)
```

```
let a = 1
if (true) {
  a = 2
}
console.log(a)
```

## 2. Đáp án

Phần này không có đáp án, các bạn tự chạy code sẽ ra nhé