

# ⚡ Types of JavaScript Errors

---

Errors in JavaScript are mainly of three types:

❶ Syntax Error | ❷ Reference Error | ❸ Type Error

---

## ❶ Syntax Errors

These occur when the code structure is invalid, and the program won't even start executing.

### 1.1 ⛔ Uncaught SyntaxError: Identifier 'a' has already been declared

- This happens when you **redeclare a variable** that's already been declared with `let`.
- **!** No execution will happen beyond this point.

```
let a = 5;  
let a = 10; // ✖ SyntaxError
```

### 1.2 ⛔ Uncaught SyntaxError: Missing initializer in const declaration

- This means you **declared a `const` variable without assigning it a value**.
- `const` **must** be initialized at the time of declaration.

```
const b; // ✖ SyntaxError
```

## ❷ Reference Errors

These occur when trying to access variables that aren't properly declared or are in the Temporal Dead Zone.

### 2.1 ? Uncaught ReferenceError: x is not defined

- You tried to **use a variable that was never declared** anywhere in your program.

```
console.log(x); // ✖ ReferenceError
```

### 2.2 ⚠ Uncaught ReferenceError: Cannot access 'a' before initialization

- You tried to **access a `let` or `const` variable before it was initialized**.
- This happens because of the **Temporal Dead Zone (TDZ)** — the phase between hoisting and initialization.

```
console.log(a); // ✖ ReferenceError  
let a = 5;
```

---

## 3 Type Errors

These happen when a value is not of the expected type.

### 3.1 💧 Uncaught TypeError: Assignment to constant variable

- You **tried to reassign a value to a `const` variable**, which is not allowed.

```
const c = 10;  
c = 20; // ✖ TypeError
```

---

## 🌟 Good Practices to Avoid Such Errors

- ☒ Prefer `const` whenever possible.
  - ☒ Use `let` only when reassignment is needed.
  - ☐ Avoid using `var` (it can lead to weird bugs due to hoisting issues).
  - ☒ Declare and initialize variables at the top of their scope to **minimize the Temporal Dead Zone**.
-