

Existe a forma padrão de declarar if/else em JS que eu conheço óbvio:

```
function numeroPositivo(num) {  
  let resultado;  
  
  if(num < 0) {  
    resultado = false;  
  } else {  
    resultado = true;  
  }  
  return resultado;  
}
```

```
numeroPositivo(2);
```

Mas existe uma forma diferente de escrever o mesmo código, você pode colocar sua condicional dentro de uma const e essa é uma boa prática de programação em JS, de código limpo porque você vai:

- Estar deixando legível para qualquer pessoa que for ler o if
- A pessoa vai entender que você está checando se é aquela condição
- Armazenar condicionais em variáveis

Olhe o exemplo:

```
function numeroPositivo(num) {  
  let resultado;  
  
  const ehNegativo = num < 0;  
  
  if (ehNegativo) {  
    resultado = false;  
  } else {  
    resultado = true;  
  }  
  
  return resultado;  
}
```

A pessoa vai entender neste exemplo que você está checando se o número é negativo e você vai executar todo o resto normalmente.

Nós temos uma Terceira opção mais curta ainda:

```
function numeroPositivo(num) {
  const ehNegativo = num < 0;

  if(ehNegativo) {
    return false;
  }

  return true;
}
```

Nós encurtamos o código mais ainda para mostrar que a gente não precisa necessariamente todas as vezes do else. Então eu poderia fazer essa mesma função, essa mesma coisa sem usar a variável resultado e aí eu vou poupar a memória do meu computador porque eu não vou estar precisando de mais espaço, então você pode apenas retornar. O return vai sair da função, não importa se ele está dentro de um if ele vai sair da função inteira.

Outra coisa muito interessante que nós podemos fazer é **aninhar** ifs um dentro do outro:

```
function numeroPositivo(num) {
  const ehNegativo = num < 0;
  const maiorQueDez = num > 10;

  if (ehNegativo) {
    return "Esse número é Negativo";
  } else if (!ehNegativo && maiorQueDez) { // Se ele não é negativo e ele é maior
que 10
    return "Esse número é positivo e maior que 10!";
  }

  return "Esse número é positivo";
}
```

! é diferente;

JavaScript não tem **elseif**, as palavras sempre estão espaçadas.

### Switch

Ele serve para quando você tem muitas verificações que você precisa fazer já que em um if ficaria muito grande e você pode fazer de uma forma que você faz o switch case mas é bom lembrar que:

- Ele equivale a uma comparação de tipo e valor (===)

No if você pode ter apenas 2 == fazendo a comparação e já o Switch compara o tipo e o valor sempre, é uma comparação bem mais estrita.

- Ele sempre precisa ter um valor **default**  
Ele acontece quando nenhuma das opções funciona

Exemplo:

```
function getAnimal(id) {  
  switch(id) {  
    case 1:  
      return "Cão";  
    case 2:  
      return "Gato";  
    case 3:  
      return "Pássaro";  
    default:  
      return "Peixe";  
  }  
}
```

```
getAnimal(1);  
getAnimal(2);  
getAnimal("1");
```

- Ideal para quando se precisa comparar muitos valores