

ntander Coders 2024.1 | Front-End | #1178

Isaias Soares

Id: 1178008



Constantes e Variáveis (valores únicos) e sua relação com o conceito matemático

Constantes e Variáveis (valores únicos) e sua relação com o conceito matemático

Declaração e atribuição

Usamos o `var` para indicar a declaração de uma variável (ainda veremos opções melhores) e usamos o `=` (igual) para atribuir um valor à ela. Vale observar que o operador `=` é um operador de identidade da matemática, ou seja, quando dizemos `a = 10` isso quer dizer que, não importa se utilizamos o `10` ou o `a`, o valor de ambos será o mesmo.

Sendo assim, variáveis podem ser substituídas em qualquer linha de um programa, **contanto que a linha não faça atribuição na variável**, pelo valor que ela carrega sem prejuízos.

```
// Boolean
var a = true;
var b = false;

// Number
var c = 10;
var d = 11.5;
```

```
// String
var e = "teste";

// Undefined
var f;
var g = undefined;

// Function
var h = function(){};
var i = () => {};

// Null
var l = null;

// Object
var j = {};
var k = []; //array
```

Let e const

Acima usamos `var`, mas podemos ter alguns problemas com isso. No JavaScript, temos um comportamento diferente das variáveis, elas podem ser declaradas depois de utilizadas, pois elas são "levantadas" para o início do escopo. Isso se chama **hoisting**.

Variáveis também não respeitam seu escopo, uma variável declarada em uma função estará disponível depois do escopo da função, por exemplo. Esses dois comportamentos são peculiares e diferentes de muitas linguagens, só se aplicando às variáveis declaradas com `var`.

Como esses comportamentos são desnecessários, na imensa maioria das vezes, e causam mais problemas do que ajudam, a versão ES6 do JavaScript adicionou duas alternativas que se comportam como as variáveis de outras linguagens.

Vamos padronizar, a partir de agora, que usaremos essas alternativas, sendo assim usaremos a palavra **let** para declaração de variáveis e a palavra **const** para declaração de constantes. Isso fará com que os escopos sejam respeitados e não haja hoisting.

```
if (true) {  
  var titulo = "Professor"  
}
```

```
console.log(titulo) // Resultado: Professor
```

```
if (true) {  
  let titulo2 = "Professor"  
}
```

```
console.log(titulo2) // Resultado: Uncaught ReferenceError: titulo2 is not defined
```

```
if (true) {  
  let titulo3 = "Professor"  
}
```

```
console.log(titulo3) // Resultado: Uncaught ReferenceError: titulo3 is not defined
```

Let

Uma **let**, tem o mesmo comportamento de um **var** só que prevenindo o hosting.

Constantes

Uma constante tem o valor único e imutável, pode parecer estranho à primeira vista, mas muitas coisas na programação são constantes e não variáveis.

Veja o exemplo abaixo:

```
const valor1 = prompt('digite um número'); //solicita um número
const valor2 = prompt('digite um número'); //solicita outro número

const media = (valor1 + valor2) / 2.0; //calcula a média

console.log(media); //mostra no console
```

Usamos apenas constantes. Os números digitados pelo usuário não podem mudar durante o programa, portanto, são constantes. A média de duas constantes também é constante. Sendo assim, o programa pode ser escrito sem nenhuma variável.

Referências e Materiais Complementares

- [ECMAScript® Language Specification](#)
- [Sintaxe e tipos](#)
- [Type Variables](#)

Próximo Tópico