



Desenvolvimento Web Completo 2022


Por Jorge Sant Anna, Programador
Jamilton Damasceno, Analista de Sistema e Professor




Seção 4: JavaScript




Aula 7: Variáveis Null e Undefined

- 
- Agora iremos entender as diferenças que existem entre dois valores que podem estar presentes na lógica dos nossos scripts, que são os valores null e undefined.
 - Esses valores possuem uma diferença muito sutil entre si e é importante compreender essa diferença para evitar possíveis bugs que, geralmente, são bastante complexos de serem resolvidos.

- 
- As variáveis null e undefined são considerados valores primitivos, ou seja, assim como number, string e boolean, são valores nativamente suportados pelo JavaScript.
 - A diferença prática entre null e undefined, é que o null representa a ausência intencional de um valor, ou seja, podemos criar uma variável, por exemplo, a variável “teste” para indicarmos que não tenhamos, nesse momento, um valor para ser atribuído a essa variável, mas, queremos, intencionalmente, indicar que ela, apesar de declarada, faremos uma referência para uma posição sem nenhum tipo de informação, ou seja, intencionalmente apontaremos a ausência de um valor dentro da variável.


Exemplo


```
//Variável Null  
var teste = null;
```


- 
- Já o undefined, assim como o null, pode ser um valor atribuído, porém, o valor undefined indica que a variável, apesar de declarada, de fato, não possui nenhum valor.
 - Por exemplo, se criarmos a variável “testar”, podemos atribuir o valor undefined, indicando que aquela variável está indefinida, ou seja, sem nenhum valor ou, podemos omitir essa atribuição.

Exemplo

```
//Variável Undefined  
var testar = undefined;  
var testar2;
```


- 
- Então, se apenas declarar uma variável, automaticamente ela será undefined. Se declararmos mais uma variável, como string, como number ou como boolean, ela não será undefined, ela será do respectivo tipo.
 - E se declarar uma variável com valor de null, estamos indicando de forma intencional que aquela variável representa, naquele momento, a ausência intencional de algum valor.

- 
- Em qualquer uma dessas situações, null ou undefined, essas variáveis podem ser trabalhadas, ou seja, podemos, posteriormente, atribuir valores a elas.
 - O que muda é que é importante ter em mente que esses valores são diferentes. Por isso, em testes condicionais é importante considerar se a variável é intencionalmente vazia, portanto, null, ou se ela é undefined, ou seja, aponta para um valor inexistente.

- 
- De forma geral, as funções podem retornar o valor null, desde que essas funções sejam encontradas pela interpretação do JavaScript e sua respectiva lógica retorna esse tipo de informação.
 - Já o undefined, está mais caracterizado para as situações em que tentamos acessar um recurso que está indisponível. Nesse caso, por exemplo, se tentasse selecionar o navegador, um elemento que não existe, teríamos um retorno indefinido. Isso porque, naturalmente, o elemento não existe.