

Programação Estruturada

Aula 05 - Entendendo o seu Código: depuração
Videoaula 02: Usando o Console da Ferramenta
de Depuração



Videoaula 02: Usando o Console da Ferramenta de Depuração

Apesar de ter nos ajudado a encontrar os erros, na verdade, o uso das saídas de um programa não é a melhor maneira de encontrá-los. Todos os navegadores modernos têm uma ferramenta de depuração integrada, a qual pode ser ligada e desligada apresentando informações diversas para o programador. Para ativar a ferramenta de depuração, podemos acessá-la diretamente no menu do navegador. Nesta aula, para fins de ilustração, iremos utilizar a ferramenta de depuração do Chrome. Nela, a tecla F12 funciona como um atalho para acessar a ferramenta de depuração. Essa tecla também tem essa mesma função nos principais navegadores.

Para fins de depuração, podemos usar o console da ferramenta de depuração. Além disso, podemos usar outras funções que são mais sofisticadas e eficientes para esse processo. Porém, antes de conhecermos tais funções, vamos ver como podemos escrever saídas do programa no console dessa ferramenta.

Nesse exemplo o que a gente vai fazer é também tentar identificar o erro dessa página HTML, só que dessa vez vamos utilizar o console da ferramenta de depuração. Veja que a nossa página tem o mesmo erro com 5 e 6, e o resultado que aparece é "true". E agora quero verificar novamente o valor das variáveis ao longo da execução do programa.

Então, veja que o meu HTML não tem mais o campo depuração, a minha página permanece intocável, então vou tentar achar um erro que está aqui no meu JavaScript, tá? Para isso, vou usar o método do `console` (`console.log`).

Código 1 - 05_2 Media.js

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>Programação Estruturada - Aula 04</title>
```

```

5 </head>
6 <body>
7   <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
8
9   <h1>Situação do Aluno</h1>
10
11   N1: <input type="number" id="N1" value="">
12   N2: <input type="number" id="N2" value="">
13   <button onclick='situacao()'>OK</button>
14   <p id="resultado"></p>
15
16   <script src="script.js"></script>
17 </body>
18 </html>
19

```

```

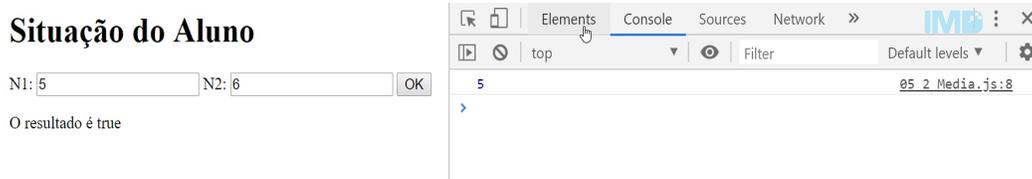
1 function situacao() {
2   var saida = resultado();
3   document.getElementById('resultado').innerHTML = "O resultado é " + saida;
4 }
5
6 function resultado() {
7   var x = Number(document.getElementById('N1').value);
8   var y = Number(document.getElementById('N2').value);
9   var m = media(x,y);
10  console.log(m);
11  var r = m >= 7;
12  console.log(r);
13
14  return r;
15 }
16
17 function media(a, b) {
18   var c = a + b/2;
19   return c;
20 }
21

```

Vamos logar no console o valor de x para ver se eu estou, de fato, lendo o valor de x corretamente (linha 8). Ao executar a minha página, passando o 5 aqui na primeira nota e 6 na segunda, o erro aparece (o resultado é "true"), mas veja que nada apareceu, isso porque esse `log`, essa escrita, é feita no console, na ferramenta de depuração, então é preciso abri-la. Aqui no Google Chrome e em muitos navegadores, a tecla F12 é usada pra isso.

Então, ao apertar o F12 abre-se o console, e em seu menu aqui temos *elements*, *console*, *sources*, *network*, tem outras partes aqui também, mas o que a gente vai usar agora é o console, tá?

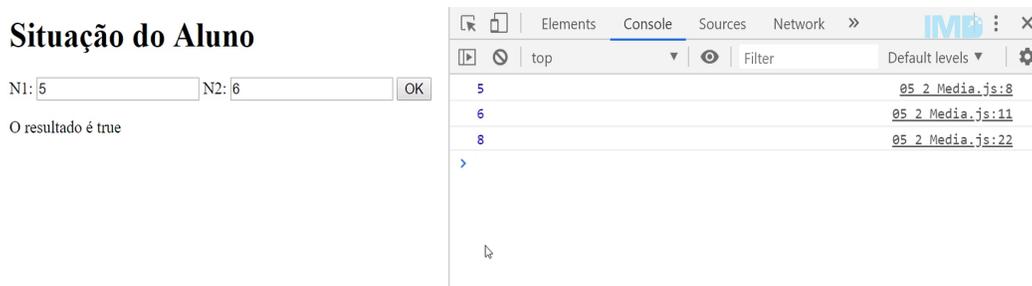
Figura 1 - Console da Ferramenta de Depuração do Chrome



Note que, de fato, foi escrito 5 no nosso console. Vamos recarregar a página e veja que o console é limpo, e vamos executar novamente só para que você veja o momento em que isso acontece: na hora que eu dou OK, ele escreve no console o valor 5.

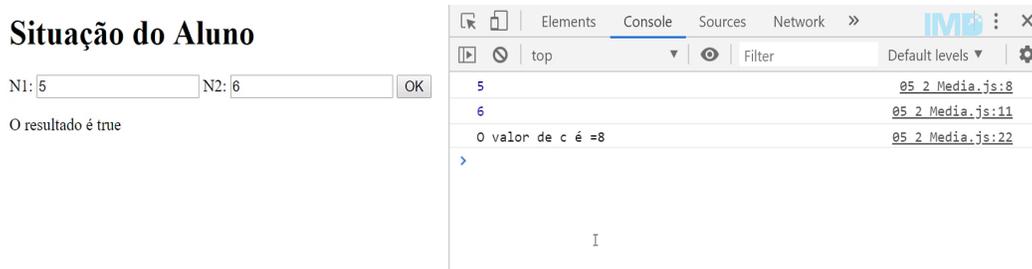
E aí, novamente, eu poderia querer ver o valor de y e o valor da media (linhas 11 e 14), veja que eu posso fazer tudo de uma vez só, ok? Vou recarregar a página, ele vai escrever o valor de x , o valor de y e o valor da *media*, tudo ao mesmo tempo, ele até diz aqui do lado direito qual foi a linha que o termo foi escrito, e podemos ver que a média é 8, então poderíamos escrever também o valor de c (linha 22). Vamos recarregar com os mesmos dados (5 e 6), e veja que o valor de c escrito seria 8, ok?

Figura 2 - Exibindo várias mensagens



Lembre-se que aqui você pode passar qualquer coisa, caso você queira escrever mais, inclusive, uma *string* com o valor de c , por exemplo, aí vou passar o seu valor (linha 22) e vamos executar novamente os valores que a gente está usando, veja que ele escreve "o valor de c é =8".

Figura 3 - Exibindo mensagens mais completas



Você pode passar qualquer coisa para escrever no console e esse valor vai aparecer aqui, e podemos usar esse método para saber até mesmo qual o valor de diversas variáveis, ao mesmo tempo, ou chamadas de funções, para fazer a depuração do programa, tá certo?

Note que, apesar de a escrita de saídas no console ser útil para as depurações simples, assim como as outras formas de escritas de saída que vimos, ela tem a desvantagem de limitar a sua visão sobre o que realmente está acontecendo em seu programa, tanto nas variáveis que você imprimiu quanto no fluxo do programa antes e depois da declaração impressa. Por outro lado, as ferramentas de depuração nos oferecem maneiras bem mais sofisticadas de acompanhar a execução do programa. Todas elas utilizam alguns conceitos básicos comuns que serão apresentados na próxima videoaula. Até lá!