

Comandos do Console JavaScript

Visual Studio 2015

Publicado: agosto de 2016

Para obter a documentação mais recente do Visual Studio 2017 RC, consulte a documentação do Visual Studio 2017 RC (<http://docs.microsoft.com/visualstudio>).



Você pode usar comandos para enviar mensagens e executar outras tarefas na janela Console do JavaScript do Visual Studio. Para obter exemplos que mostram como usar essa janela, consulte início rápido: depurar JavaScript (<https://msdn.microsoft.com/pt-br/library/dn688631.aspx>). As informações neste tópico se aplica a aplicativos da Windows Store, aplicativos da Windows Store Phone e aplicativos criados usando o Visual Studio Tools for Apache Cordova. Para obter informações sobre comandos do console com suporte em aplicativos Cordova, consulte depurar seu aplicativo (<https://msdn.microsoft.com/pt-br/library/dn757061.aspx>). Para obter informações sobre como usar o console nas ferramentas F12 do Internet Explorer, consulte neste tópico (<https://msdn.microsoft.com/library/ie/dn255006.aspx>).

Se a janela do Console do JavaScript é fechada, você pode abri-lo enquanto você estiver depurando no Visual Studio escolhendo **Depurar** > **Windows** > **JavaScript Console**.

Observação

Se a janela não estiver disponível durante uma sessão de depuração, certifique-se de que o tipo de depurador é definido como **Script** nas propriedades de depuração para o projeto.

Comandos do objeto console

Esta tabela mostra a sintaxe dos comandos de objeto do `console` que você pode usar na janela do Console do JavaScript ou para enviar mensagens de seu código para o console. Esse objeto oferece diversos formatos para que você possa distinguir entre mensagens informativas e mensagens de erro, se desejar.

Você pode usar o formato de comando mais longo, `window.console.[command]`, se precisar evitar uma possível confusão com objetos locais chamados `console`.

Dica

Versões anteriores do Visual Studio não suportam o conjunto completo de comandos. Use o IntelliSense no objeto de console para obter informações rápidas sobre comandos com suporte.

Comando	Descrição	Exemplo
<code>assert(expression, message)</code>	Envia uma mensagem se <code>expression</code> for avaliada como false .	<pre>console.assert((x == 1), "assert message: x != 1");</pre>

Comando	Descrição	Exemplo
<code>clear()</code>	Limpa as mensagens da janela do console, incluindo mensagens de erro de script, e limpa também o script exibido na janela do console. Não limpa o script inserido no prompt de entrada do console.	<code>console.clear();</code>
<code>count(title)</code>	Envia o número de vezes que o comando <code>count</code> foi chamado para a janela do console. Cada chamada do comando <code>count</code> é identificada exclusivamente pelo <code>title</code> opcional. A entrada existente na janela do console é identificada pelo parâmetro <code>title</code> (se presente) e atualizada pelo comando <code>count</code> . Não é criada uma nova entrada.	<code>console.count();</code> <code>console.count("inner loop");</code>
<code>debug(message)</code>	Envia <code>message</code> para a janela do console. Esse comando é idêntico a <code>console.log</code> . Os objetos transmitidos usando o comando são convertidos em um valor de cadeia de caracteres.	<code>console.debug("logging message");</code>
<code>dir(object)</code>	Envia o objeto especificado para a janela do console e o exibe em um visualizador de objetos. Você pode usar o visualizador para inspecionar propriedades na janela do console.	<code>console.dir(obj);</code>
<code>dirxml(object)</code>	Envia o nó XML <code>object</code> especificado para a janela do console e o exibe como uma árvore de nós XML.	<code>console.dirxml(xmlNode);</code>
<code>error(message)</code>	Envia <code>message</code> para a janela do console. O texto da mensagem está em vermelho e antecedido por um símbolo de erro. Os objetos transmitidos usando o comando são convertidos em um valor de cadeia de caracteres.	<code>console.error("error message");</code>
<code>group(title)</code>	Inicia um agrupamento das mensagens enviadas à janela do console e envia o <code>title</code> opcional como um rótulo de grupo. Os grupos podem ser aninhados e aparecer em um modo de exibição de árvore na janela do console. Os comandos <code>group*</code> podem facilitar a exibição da saída da janela do console em alguns cenários, como quando um modelo de componente está em uso.	<code>console.group("Level 2 Header");</code> <code>console.log("Level 2");</code> <code>console.group();</code> <code>console.log("Level 3");</code> <code>console.warn("More of level 3");</code> <code>console.groupEnd();</code> <code>console.log("Back to level 2");</code> <code>console.groupEnd();</code> <code>console.debug("Back to the outer level");</code>

Comando	Descrição	Exemplo
<code>groupCollapsed(title)</code>	Inicia um agrupamento das mensagens enviadas à janela do console e envia o <code>title</code> opcional como um rótulo de grupo. Os grupos enviados usando <code>groupCollapsed</code> aparecem em uma exibição recolhida por padrão. Os grupos podem ser aninhados e aparecer em um modo de exibição de árvore na janela do console.	O uso é o mesmo do comando <code>group</code> . Consulte o exemplo do comando <code>group</code> .
<code>groupEnd()</code>	Encerra o grupo atual. Requisitos: Visual Studio 2013	Consulte o exemplo do comando <code>group</code> .
<code>info(message)</code>	Envia <code>message</code> para a janela do console. A mensagem é prefaciada por um símbolo de informação.	<code>console.info("info message");</code> Para obter mais exemplos, consulte formatação de saída do console mais adiante neste tópico.
<code>log(message)</code>	Envia <code>message</code> para a janela do console. Se você transmitir um objeto, este comando o enviará para a janela do console e o exibirá num visualizador de objeto. Você pode usar o visualizador para inspecionar propriedades na janela do console.	<code>console.log("logging message");</code>
<code>msIsIndependentlyComposed(element)</code>	Usado em aplicativos da Web. Sem suporte em aplicativos da Store usando JavaScript.	Sem suporte.
<code>profile(reportName)</code>	Usado em aplicativos da Web. Sem suporte em aplicativos da Store usando JavaScript.	Sem suporte.
<code>profileEnd()</code>	Usado em aplicativos da Web. Sem suporte em aplicativos da Store usando JavaScript.	Não há suporte.
<code>select(element)</code>	Seleciona o HTML especificado <code>element</code> no Explorador do DOM (https://msdn.microsoft.com/pt-br/library/hh441474.aspx).	<code>console.select(element);</code>
<code>time(name)</code>	Inicia um temporizador que é identificado pelo parâmetro <code>name</code> opcional. Quando usado com <code>console.timeEnd</code> , calcula o tempo decorrido entre <code>time</code> e <code>timeEnd</code> e envia o resultado (medido em ms) ao console usando a cadeia de caracteres <code>name</code> como prefixo. Usado para habilitar a instrumentação de código do aplicativo para medir o desempenho.	<code>console.time("app start");</code> <code>app.start();</code> <code>console.timeEnd("app start");</code>
<code>timeEnd(name)</code>	Interrompe um temporizador que é identificado pelo parâmetro <code>name</code> opcional. Consulte o comando <code>time</code> do console.	<code>console.time("app start");</code> <code>app.start();</code> <code>console.timeEnd("app start");</code>

Comando	Descrição	Exemplo
<code>trace()</code>	Envia um rastreamento de pilha à janela do console. O rastreamento inclui a pilha de chamadas completa e informações como o nome do arquivo, o número da linha e o número da coluna.	<code>console.trace();</code>
<code>warn(message)</code>	Envia <code>message</code> para a janela do console, prefaciada por um símbolo de aviso. Os objetos transmitidos usando o comando são convertidos em um valor de cadeia de caracteres.	<code>console.warn("warning message");</code>

Comandos variados

Esses comandos também estão disponíveis na janela do Console do JavaScript (não estão disponíveis no código).

Comando	Descrição	Exemplo
<code>\$0</code> , <code>\$1</code> , <code>\$2</code> , <code>\$3</code> , <code>\$4</code>	Retorna o elemento especificado para a janela do console. <code>\$0</code> retorna o elemento selecionado atualmente no Explorador do DOM, <code>\$1</code> retorna o elemento selecionado anteriormente no Explorador do DOM e assim por diante, até o quarto elemento selecionado anteriormente.	<code>\$3</code>
<code>\$(id)</code>	Retorna um elemento por ID. Esse é um comando de atalho para <code>document.getElementById(id)</code> , onde <code>id</code> é uma cadeia de caracteres que representa a ID do elemento.	<code>\$("contenthost")</code>
<code>\$\$(selector)</code>	Retorna uma matriz de elementos que correspondem ao seletor especificado usando a sintaxe do seletor de CSS. Este é um comando de atalho para <code>document.querySelectorAll()</code> .	<code>\$\$(".itemlist")</code>
<code>cd()</code> <code>cd(window)</code>	Permite que você altere o contexto de avaliação da expressão, da janela de nível superior padrão da página para a janela do quadro especificado. Chamar <code>cd()</code> sem parâmetros reverte o contexto para a janela de nível superior.	<code>cd();</code> <code>cd(myframe);</code>
<code>select(element)</code>	Seleciona o elemento especificado em Explorador do DOM (https://msdn.microsoft.com/pt-br/library/hh441474.aspx).	<code>select(document.getElementById("element"));</code> <code>select(\$("#element"));</code> <code>select(\$1);</code>
<code>dir(object)</code>	Retorna um visualizador para o objeto especificado. Você pode usar o visualizador para inspecionar propriedades na janela do console.	<code>dir(obj);</code>

Verificando se um comando do console existe

Você pode verificar se um comando específico existe antes de tentar usá-lo. Este exemplo verifica a existência do comando `console.log`. Se `console.log` existir, o código irá chamá-lo.

JavaScript

```
if (console && console.log) {  
    console.log("msg");  
}
```

Examinando objetos na janela Console do JavaScript

Você pode interagir com qualquer objeto que esteja no escopo usando a janela Console do JavaScript. Para inspecionar um objeto fora do escopo na janela do console, use `console.log`, `console.dir`, ou outros comandos no seu código. Como alternativa, você pode interagir com o objeto da janela do console enquanto ele estiver no escopo, definindo um ponto de interrupção no seu código (**ponto de interrupção** > **Inserir ponto de interrupção**).

Formatação de saída do console

Se você transmitir diversos argumentos para `console.log`, o console os tratará como uma matriz e concatenará a saída.

JavaScript

```
var user = new Object();  
user.first = "Fred";  
user.last = "Smith";  
  
console.log(user.first, user.last);  
// Output:  
// Fred Smith
```

`console.log` também oferece suporte a padrões de substituição "printf" para a formatação da saída. Se você usar padrões de substituição no primeiro argumento, outros argumentos serão usados para substituir os padrões especificados na ordem em que foram usados.

Há suporte para os seguintes padrões de substituição:

- %s - cadeia de caracteres
- %i - inteiro
- %d - inteiro
- %f - flutuação
- %o - objeto
- %b - binário
- %x - hexadecimal
- %e - expoente

Estes são alguns exemplos de como usar padrões de substituição no `console.log`:

JavaScript

```
var user = new Object();
user.first = "Fred";
user.last = "Smith";
user.age = 10.01;
console.log("Hi, %s %s!", user.first, user.last);
console.log("%s is %i years old!", user.first, user.age);
console.log("%s is %f years old!", user.first, user.age);

// Output:
// Hi, Fred Smith!
// Fred is 10 years old!
// Fred is 10.01 years old!
```

Consulte também

Início rápido: Depurar o JavaScript (<https://msdn.microsoft.com/pt-br/library/dn688631.aspx>)

Início rápido: Depurar HTML e CSS (<https://msdn.microsoft.com/pt-br/library/hh441474.aspx>)

© 2017 Microsoft