

jQueryセクター

jQueryは、エッジケースにのみ使用する必要があります

jQueryはできるだけ使用しないようにすることをお勧めします。ウェブアプリケーションは絶えず変化しておりUIとページ構造への小さな更新は、使用されるjQueryを破壊する可能性があります。

WalkMeは、すぐに使用できるエレメント選択で、jQueryを使用せずに、基盤となるアプリケーションの変更に対応します。

概要

jQueryは、HTMLのページ操作に使用できるJavascriptライブラリです。jQueryセクターを使用することでID、クラス、タイプ、属性、属性値などの特性に基づきHTML構造内のエレメントを識別できます。

WalkMeエディタで最も一般的なjQueryセクターは次の目的で使用されます。

- **オンボーディングのゴール作成**：画面上のエレメントに基づき、オンボーディングのゴールを作成します。
- **Selected Element (選択済みエレメント) のランチャー**：ランチャーの [Selected Element (選択済みエレメント)] タブにjQueryセクターを追加します。
- **値の取得**：画面上のユーザー変数またはテキストから値を取得し、ダイナミック・アナウンスに組み込むか、ルールにて使用します。
- **ページ条件の確認**：WalkMeに指示することで、画面上のエレメントの属性を分析し、ルールエンジンのルールタイプとして使用します。

使用方法

要素を右クリックして [要素の検査] を選択し、HTML で要素を見つけます。その要素が画面と開発者パネルで強調表示されます。jQueryセクターを構築するために、エレメント固有の特性（場所、クラス、ID、スタイルなど）を特定します。エレメントは、その場所または属性（クラス、ID、およびスタイル）によって識別できます。

© Digital Adoption Institute

- [jQueryの基礎と応用](#)のコースをご覧ください
- DAIアカウントはまだお持ちでないですか？ [サインアップはこちらから](#)。

ステップ1 jQueryセレクターを構築します

WalkMeでjQueryの要素を使用するための基本的な構文は、**要素 セレクター 値**、となります。

要素を特定すると、特に興味度の高いものを調べるjQueryセレクターを作成します。

jQueryセレクターの例は次のとおりです。

- id create-accountのul要素 : ul#create-account
- class navのul要素 : ul.nav
- class navおよびclass flタイプのul要素 : ul.nav.fl

[便利なjQueryセレクターのリストについては、以下を参照してください。](#)

ステップ2 : コンソールにてセレクターをテストします

1. 開発者パネルのコマンドラインにwmjQuery(“”)と入力し、jQueryセレクターを“”の間に挿入して、**Enter**キーを押します。
 - 注 : wmjQueryは、Editor拡張機能が有効になっているブラウザでのみ使用できます。
</nas/content/live/kbwalkme/wp-content/uploads/2017/09/jquery-console-video.mp4>
2. クエリが求める値を返す場合は、セレクターが機能しています。
 - 注 : 選択済みの要素を画面に表示するには、[その他]リンクをクリックします。
3. 返されたオブジェクトにカーソルを合わせることで、強調表示された要素が表示されます。

ステップ3 WalkMeでjQueryセレクターを使用します

</nas/content/live/kbwalkme/wp-content/uploads/2017/09/walkme-editor.mp4>

jQueryの要素の使用方法に応じ、jQueryはWalkMeの次の場所で使用できます。

- jQueryの要素を使用し、[Options]オプション)]メニューの [Selected Element]選択済み要素)] タブにて、ステップまたはLauncherの要素を識別します。
- jQueryの要素を使用し、[Additional Step Triggers]追加のステップトリガー)]メニューでクリックまたはカーソル移動したときに、ウォークスルーの次のステップをトリガーします。
- jQueryの要素で画面上の値を取得し、ステップ用バルーンまたはシャウトアウトのリッチテキストエディターでダイナミック・テキストとして使用します。
- jQueryの要素により、[ゴール]タブでゴールを作成します。
- ルールエンジンのjQueryセレクターをルールタイプとして使用します (jQueryと同様に画面上の要素のルールタイプと同様に使用されます) 。

ヒント

WalkMeのビルドではjQueryを控えめに、常に比較的緩い条件と組み合わせて使用することを忘れないでください。詳細については、次の記事を参照してください。[WalkMeパフォーマンスのベストプラクティス](#)

利用可能なルールエンジンのjQuery演算子：

- **True**：1つ以上のエレメントが発見されたことを意味します。
- **False**：エレメントが発見されなかったことを意味します。
- **Visible**：エレメントが表示されていることを意味します。
- **Text Is / Is Not**：テキストがテキスト入力と等しいことを意味します。
 - 注：セレクターが複数のエレメントを返す場合、すべてのエレメントのテキストが1つのエレメントとして結合されます。
- **Greater Than / Less Than**：値が整数より大きい、または小さいことを意味します。
- **Like / Not Like**：複雑なテキスト構文を意味します。
- **Length Is / Is Not / Greater Than / Less Than**：整数に等しい、より大きい、または小さいことを意味します。
 - クエリが返すエレメント数をクエリ実行するときに使用されます。
 - コンソールで正確なクエリの長さを確認します。`wmjQuery("element.value").length`

注

WalkMeのEditorがサポートするのはjQuery1.8.3以下で使用可能なjQuery構文で記述されたセレクターのみです。ただし、ウェブサイト上ではWalkMeのjQuery[wjQuery]バージョンとの競合の心配なく、他の目的で任意のバージョンのjQueryを使用できます。

[WalkMeのjQueryセレクターオブティマイザー](#)も使用することをお勧めします。

便利なjQueryセレクター

クエリ	説明
エレメント	エレメントに関するクエリの実行
element[attribute]	エレメントの属性に関するクエリの実行
element[attribute][attribute]	エレメントの複数属性に関するクエリの実行

element[attribute="attribute value"]	エレメントの属性およびその値に関するクエリの実行
element[attribute^="text"]	属性の先頭にあるテキストをクエリ
element[attribute*="text"]	属性に含まれるテキストをクエリ
element[attribute!="text"]	属性に含まれないテキストをクエリ
element child-element	子エレメントのクエリ（スペースは、最初のエレメントが2番目の親エレメントであることを示します）
element:checked	チェックボックスのステータスに関するクエリの実行（チェックまたは未チェック）
[element=""]:visible	エレメントの表示に関するクエリの実行
element:contains(Text)	クエリのエレメントにテキストが含まれています
element:eq(#)	特定結果に対して順番にクエリを実行（カウントは0から始まります）

jQueryチートシート

その他のjQueryのヒントと例については、[jQueryチートシート](#)を参照してください。

jQueryセレクターのテスト

jQueryセレクターを選択するときは、ウェブサイトやアプリケーションに対するパフォーマンスへの悪影響が最少のものを選択するため、いくつかのセレクターを比較することを推奨します。そのためには、以下の手順に従ってください。

1. ウェブブラウザの開発ツールを開きます。
2. 開発ツールの[コンソール]タブに移動します。
3. コンソールに以下を入力します。
 - `var time1 = window.performance.now();` `wmjQuery('<your jQuery selector>');`
 - `var time2 = window.performance.now();` `console.log(time2 - time1);`
4. **Enter**キーを押します。
5. コンソールでこのコマンドを数回実行し、所要時間のおおよその平均を確認します。
6. これと同じコマンドをいくつかの代替jQueryセレクターにて実行するとともに、最少時間で実行できるものを選択します。

注

- Shadow DOMがサポートされていれば、EditorでShadow DOMエレメントをキャプチャしてjQueryを使って識別できます。
- ただし、DevToolsコンソールでwmjQueryを実行することはできません。代わりに以下の構文を使います。

```
_walkmeInternals.ctx.get('jQuerySelectorRunnerShadow').run('enter your selector here')
```