



# コマンドライン インターフェイス

ここでは、Catalyst 6500 シリーズ スイッチの SSL サービス モジュールを Command-Line Interface (CLI; コマンドライン インターフェイス) で使用する方法について説明します。Catalyst 6500 シリーズ スイッチの SSL サービス モジュールで使用する CLI は、Cisco IOS CLI に基づいています。このマニュアルで紹介していない Cisco IOS コマンドについては、次の最新の Cisco IOS マニュアルを参照してください。

- 『Cisco IOS Release 12.2 Configuration Fundamentals Configuration Guide』
- 『Cisco IOS Release 12.2 Command Reference』

この章で説明する内容は、次のとおりです。

- 「ヘルプの表示方法」(P.1-1)
- 「コマンド オプションの検索方法」(P.1-2)
- 「コマンド モードの概要」(P.1-5)
- 「コマンドの no 形式および default 形式の使用」(P.1-6)
- 「CLI スtring検索の使用」(P.1-7)

## ヘルプの表示方法

各コマンド モードで使用できるコマンドの一覧を表示するには、システム プロンプトで疑問符 (?) を入力します。また、状況依存ヘルプ機能を使用すると、任意のコマンドに関連するキーワードと引数の一覧を表示できます。

表 1-1 に、各コマンド モード、コマンド、キーワード、または引数に関するヘルプを表示するためのコマンドの一覧を表示します。

表 1-1 ヘルプの表示方法

コマンド	目的
コマンドの先頭部分?	入力した文字列で始まるコマンドの一覧を表示します (コマンドと疑問符の間にはスペースを入れないでください)。
コマンドの先頭部分<Tab>	途中まで入力したコマンド名を完全なコマンドにします。
?	特定のコマンド モードで使用できるすべてのコマンドの一覧を表示します。

表 1-1 ヘルプの表示方法 (続き)

コマンド	目的
コマンド?	コマンドに関連するキーワードの一覧を表示します。 コマンドと疑問符の間にはスペースを入れます。
コマンド キーワード?	キーワードに関連する引数の一覧を表示します。 キーワードと疑問符の間にはスペースを入れます。

次の例では、特定の文字列で始まるコマンドの一覧を表示する方法、および途中まで入力したコマンド名を完全なコマンドにする方法を示します。

```
ssl-proxy# tu?
tunnel

simpson1-2# tu
```

次の例では、特定のコマンド モードで使用できるすべてのコマンドの一覧を表示する方法を示します。

```
ssl-proxy(config)# ?
Configure commands:
  aaa                               Authentication, Authorization and
                                     Accountin
  access-list                       Add an access list entry
  alias                             Create command alias
  arp                               Set a static ARP entry
  async-bootp                       Modify system bootp parameters
  banner                             Define a login banner
  boot                             Modify system boot parameters
  bridge                             Bridge Group.
  buffers                           Adjust system buffer pool parameters
  cdp                               Global CDP configuration subcommands
  class-map                         Configure QoS Class Map
  .
  .
  .
<出力を省略>
```

次の例では、キーワードに関連する引数の一覧を表示する方法を示します。

```
ssl-proxy(config-if)# channel-group 1 mode ?
  auto           Enable PAgP only if a PAgP device is detected
  desirable     Enable PAgP unconditionally
  on            Enable Etherchannel only

ssl-proxy(config-if)#
```

## コマンド オプションの検索方法

ここでは、コマンドの構文を表示する方法の例を示します。構文は任意または必須のキーワードで構成されます。コマンドのキーワードを表示するには、コンフィギュレーション プロンプトで疑問符 (?) を入力するか、コマンドの一部を入力しその後スペースを入力してから疑問符を入力します。Catalyst 6500 シリーズ SSL サービス モジュールにより使用可能なキーワードと、各キーワードの簡単な説明が表示されます。たとえば、グローバル コンフィギュレーション モードで **ssl-proxy** コマンドのすべてのキーワードを表示するには、**ssl-proxy ?** と入力します。

表 1-2 はコマンド入力の補助のため疑問符 (?) を使用方法の例です。

表 1-2 コマンド オプションの検索方法

コマンド	コメント
<pre>ssl-proxy&gt; enable Password: &lt;password&gt; ssl-proxy#</pre>	<p><b>enable</b> コマンドとパスワードを入力して、特権 EXEC コマンドにアクセスします。</p> <p>プロンプトが <code>ssl-proxy#</code> に変わると、特権 EXEC モードが開始します。</p>
<pre>ssl-proxy# configure terminal Enter configuration commands, one per line. End with CNTL/Z. ssl-proxy(config)#</pre>	<p>グローバル コンフィギュレーション モードを開始します。</p> <p>プロンプトが <code>ssl-proxy(config)#</code> に変わると、グローバル コンフィギュレーション モードが開始します。</p>
<pre>ssl-proxy(config)# crypto ca trustpoint trustpoint-label ssl-proxy(ca-trustpoint)#</pre>	<p>コンフィギュレーション サブモードを開始します。</p> <p>プロンプトに <code>ssl-proxy(ca-trustpoint)#</code> のようにサブモードが表示されると、コンフィギュレーション サブモードが開始します。</p>
<pre>ssl-proxy(config)# interface type mod/port ssl-proxy(config-if)#</pre>	<p>また、グローバル コンフィギュレーション モードで <b>interface</b> グローバル コンフィギュレーション コマンドを入力すると、インターフェイス コンフィギュレーション モードが開始します。</p> <p>プロンプトが <code>ssl-proxy(config-if)#</code> に変わると、インターフェイス コンフィギュレーション モードが開始します。</p>
<pre>ssl-proxy(config-if)# channel-group ? group channel-group of the interface  ssl-proxy(config-if)#channel-group</pre>	<p>コントローラに対して設定するコマンドを入力します。この例では <b>channel-group</b> コマンドを使用しています。</p> <p>? を入力すると、コマンドラインで次に入力が必要な項目が表示されます。この例では、<b>group</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないため、コマンドを完了するにはさらに情報を入力する必要があります。ことがわかります。</p>

表 1-2 コマンド オプションの検索方法 (続き)

コマンド	コメント
<pre>ssl-proxy(config-if)# channel-group group ? &lt;1-256&gt; Channel group number  ssl-proxy(config-if)#channel-group group</pre>	<p><b>group</b> キーワードを入力した後に <b>?</b> を入力すると、コマンドラインで次に入力が必要な項目が表示されます。この例では、チャンネル グループ番号 (1 ~ 256) を入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないため、コマンドを完了するにはさらに情報を入力する必要があることがわかります。</p>
<pre>ssl-proxy(config-if)# channel-group 1 ? mode Etherchannel Mode of the interface  ssl-proxy(config-if)#</pre>	<p>チャンネル グループ番号を入力した後に <b>?</b> を入力すると、コマンドラインで次に入力が必要な項目が表示されます。この例では、<b>mode</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないため、コマンドを完了するにはさらに情報を入力する必要があることがわかります。</p>
<pre>ssl-proxy(config-if)# channel-group 1 mode ? auto Enable PAgP only if a PAgP device is detected desirable Enable PAgP unconditionally on Enable Etherchannel only  ssl-proxy(config-if)#</pre>	<p><b>mode</b> キーワードを入力した後に <b>?</b> を入力すると、コマンドラインで次に入力が必要な項目が表示されます。この例では、<b>auto</b>、<b>desirable</b>、または <b>on</b> キーワードを入力する必要があります。</p> <p>&lt;cr&gt; が表示されていないため、コマンドを完了するにはさらに情報を入力する必要があることがわかります。</p>
<pre>ssl-proxy(config-if)# channel-group 1 mode auto ? &lt;cr&gt;  ssl-proxy(config-if)#</pre>	<p>この例では、<b>auto</b> キーワードを入力しています。<b>auto</b> キーワードを入力した後に <b>?</b> を入力すると、コマンドラインで次に入力が必要な項目が表示されます。</p> <p>&lt;cr&gt; が表示されているため、<b>Return</b> を押してコマンドを終了できる状態であることがわかります。他のキーワードが一欄に表示されている場合、キーワードを追加入力するか、<b>Return</b> を押してコマンドを終了します。</p>
<pre>ssl-proxy(config-if)# channel-group 1 mode auto ssl-proxy(config-if)#</pre>	<p>この例では、<b>Return</b> を押してコマンドを終了しています。</p>

## コマンド モードの概要

ここでは、Cisco IOS ユーザ インターフェイスのコマンド モードについて説明します。

### Cisco IOS ユーザ インターフェイス

Cisco IOS ユーザ インターフェイスは、いくつかのモードに分かれています。使用できるコマンドの種類は、現在のモードによって異なります。各コマンド モードで使用できるコマンドの一覧を表示するには、システム プロンプトで疑問符 (?) を入力します。

Catalyst 6500 シリーズ スイッチでセッションを開始するときは、まずユーザ モード (EXEC モードとも呼ばれます) を開始します。EXEC モードでは、一部のコマンドしか使用できません。すべてのコマンドを使用できるようにするには、特権 EXEC モードを開始する必要があります。特権 EXEC モードを開始するには、通常、パスワードが必要です。特権 EXEC モードでは、すべての EXEC コマンドが使用でき、またグローバル コンフィギュレーション モードを開始することもできます。大部分の EXEC コマンドは一時的なものです。例として、特定のアイテムの現在の状態を表示する **show** コマンドや、カウンタやインターフェイスを消去する **clear** コマンドがあげられます。Catalyst 6500 シリーズ スイッチの再起動時に EXEC コマンドは保存されません。

コンフィギュレーション モードでは、実行コンフィギュレーションを変更できます。後でコンフィギュレーションを保存すると、これらのコマンドは Catalyst 6500 シリーズ スイッチを再起動しても保存されます。各種のコンフィギュレーション モードを使用できるようにするには、グローバル コンフィギュレーション モードを開始します。このモードからは、インターフェイス コンフィギュレーション モード、サブインターフェイス コンフィギュレーション モード、各種のプロトコル固有のモードを開始できます。

ROM モニタ モードは、Catalyst 6500 シリーズ スイッチが適切に起動できない場合に使用される独立したモードです。Catalyst 6500 シリーズ スイッチやアクセス サーバが起動時に有効なシステム イメージを検出できない場合、または起動時にコンフィギュレーション ファイルが破損していた場合、システムは ROM モニタ モードを開始する場合があります。

表 1-3 に、主なコマンド モードの概要を示します。

表 1-3 主なコマンド モードの概要

コマンド モード	アクセス方法	プロンプト	終了方法
ユーザ EXEC	ログイン。	ssl-proxy>	<b>logout</b> コマンドを使用します。
特権 EXEC	ユーザ EXEC モードから、 <b>enable EXEC</b> コマンドを入力します。	ssl-proxy#	ユーザ EXEC モードに戻る場合は、 <b>disable</b> コマンドを入力します。 グローバル コンフィギュレーション モードから、 <b>configure terminal</b> 特権 EXEC コマンドを入力します。
グローバル コンフィギュレーション	特権 EXEC モードから、 <b>configure terminal</b> 特権 EXEC コマンドを入力します。	ssl-proxy (config) #	特権 EXEC モードに戻る場合は、 <b>exit</b> または <b>end</b> コマンドを入力するか、Ctrl+Z を押します。 インターフェイス コンフィギュレーション モードを開始するには、 <b>interface</b> コンフィギュレーション コマンドを入力します。
グローバル コンフィギュレーション サブモード	グローバル コンフィギュレーション モードから、 <b>submode</b> コマンドを入力します。	ssl-proxy (config-submode) #	グローバル コンフィギュレーション サブモードに戻る場合は、 <b>exit</b> コマンドを入力します。

表 1-3 主なコマンド モードの概要 (続き)

コマンド モード	アクセス方法	プロンプト	終了方法
インターフェイス コンフィギュレーション	グローバル コンフィギュレーション モードから、 <b>interface</b> コマンドでインターフェイスを指定して開始します。	ssl-proxy(config-if)#	グローバル コンフィギュレーション モードに戻る場合は、 <b>exit</b> コマンドを入力します。 特権 EXEC モードに戻る場合は、 <b>exit</b> コマンドを入力するか、 <b>Ctrl+Z</b> を押します。 サブインターフェイス コンフィギュレーション モードを開始するには、 <b>interface</b> コマンドでサブインターフェイスを指定します。
サブインターフェイス コンフィギュレーション	インターフェイス コンフィギュレーション モードから、 <b>interface</b> コマンドでサブインターフェイスを指定します。	ssl-proxy(config-subinterface)#	グローバル コンフィギュレーション モードに戻る場合は、 <b>exit</b> コマンドを入力します。 特権 EXEC モードを開始するには、 <b>end</b> コマンドを使用するか、 <b>Ctrl+Z</b> を押します。
ROM モニタ	特権 EXEC モードから、 <b>reload EXEC</b> コマンドを入力します。システムの起動時、最初の 60 秒以内に <b>Break</b> キーを押します。	Rommon>	ROM モニタ モードを終了する場合は、 <b>boot</b> コマンドを入力してイメージをリロードする必要があります。ファイル名または他の起動命令を指定せずに <b>boot</b> コマンドを使用すると、システムはデフォルトのフラッシュ イメージ (オンボード フラッシュ メモリ上の最初のイメージ) から起動します。また、特定のフラッシュ イメージから起動するように指定することもできます ( <b>boot system flash filename</b> コマンドを使用)。

コマンド モードの詳細については、『*Configuration Fundamentals Configuration Guide*』の「Using the Command Line Interface」の章を参照してください。



(注)

EXEC レベルの Cisco IOS コマンド (**show**、**clear**、**debug** コマンドなど) をグローバル コンフィギュレーション モードやその他のモードから実行するには、**do** コマンドの前に EXEC コマンドを実行します。このコマンドの使用方法については、**do** コマンドの項を参照してください。

## コマンドの no 形式および default 形式の使用

大部分のコンフィギュレーション コマンドに、**no** 形式があります。通常、コマンドの機能をディセーブルにする場合に **no** 形式のコマンドを指定します。キーワード **no** を指定せずにコマンドを使用すると、ディセーブルにした機能が再びイネーブルになり、また、デフォルトでディセーブルに設定されている機能がイネーブルになります。たとえば、IP ルーティングはデフォルトでイネーブルに設定されています。IP ルーティングをディセーブルにするには、**no ip routing** コマンドを指定します。その後 **ip routing** コマンドを使用すると再度イネーブルになります。このマニュアルでは、コンフィギュレーション コマンドのすべての構文を紹介し、コマンドの **no** 形式の役割について説明します。

コンフィギュレーション コマンドには、**default** 形式を指定できます。コマンドの **default** 形式は、コマンドの設定値をデフォルトに戻します。大部分のコマンドはデフォルトでディセーブルに設定されているので、**default** 形式は **no** 形式と同じになります。ただし、デフォルトでイネーブルに設定されていて、なおかつ変数が特定のデフォルト値に設定されているコマンドもあります。このような場合、コ

マンドの **default** 形式を使用すると、コマンドがイネーブルになり、変数がデフォルト値に設定されます。このマニュアルでは、コマンドが **no** 形式と異なる場合のコマンドの **default** 形式について説明します。

## CLI スtring検索の使用

コマンド出力のパターンはStringと呼ばれています。CLI String検索機能を使用すると、**show** または **more** コマンドの出力を検索またはフィルタリングを行うことができ、また **--More--** プロンプトで検索やフィルタリングを行うことができます。この機能は、大量の出力をソートする場合や、出力から不要な情報を除外する場合に役立ちます。

検索機能を使用すると、指定された正規表現を含むものを最初の行から始めることができます。コマンド 1 つあたり最大で 1 つのフィルタを指定するか、**--More--** プロンプトで新しい検索を開始できます。

正規表現は、ソフトウェアが **show** または **more** コマンドの出力と一致させるために使用するパターン（語句、数値、または複雑なパターン）です。正規表現では大文字と小文字が区別され、また複雑な一致要件を設定することができます。単純な正規表現の例としては、「Serial」、「misses」、「138」などがあります。複雑な正規表現の例としては、「00210...」、「( is )」、「[Oo]utput」などがあります。

実行できるフィルタリングは 3 種類あります。

- **begin** キーワードを使用すると、出力は指定した正規表現を含む行から始まります。
- **include** キーワードを使用すると、出力には指定した正規表現を含む行が抽出されます。
- **exclude** キーワードを使用すると、出力では指定した正規表現を含む行が除外されます。

その後、このフィルタリングした出力を **--More--** プロンプトで検索できます。



(注)

CLI String検索機能では、前の出力にさかのぼって検索やフィルタリングを行うことはできません。また、CLI に HTTP アクセスしている場合は、フィルタリングを指定できません。

## 正規表現

正規表現は、コマンド出力内の 1 文字と一致する 1 文字にすることも、コマンド出力内の複数文字と一致する複数文字にすることもできます。ここでは、単一文字パターンおよび複数文字パターンを作成する方法、および繰り返し指定、選択、位置指定、およびカッコを用いたより複雑な正規表現を作成する方法について説明します。

### 1 文字のパターン

最も単純な正規表現は、コマンド出力の 1 文字と一致する 1 文字のパターンです。任意の文字 (A ~ Z, a ~ z) または数字 (0 ~ 9) を 1 文字のパターンとして使用できます。また、その他のキーボード文字 (!) や (~) など) も 1 文字のパターンとして使用できますが、一部のキーボード文字は正規表現では特別な意味を持ちます。表 1-4 に、特別な意味を持つキーボード文字の一覧を示します。

表 1-4 特別な意味を持つ文字

文字	特別な意味
.	空白スペースを含め、任意の 1 文字と一致します。
*	パターンの 0 個以上のシーケンスと一致します。

表 1-4 特別な意味を持つ文字 (続き)

文字	特別な意味
+	パターンの 1 個以上のシーケンスと一致します。
?	パターンの 0 または 1 文字と一致します。
^	文字列の先頭と一致します。
\$	文字列の末尾と一致します。
_ (アンダースコア)	コンマ (,), 左中カッコ ({}, 右中カッコ (}), 左カッコ ((), 右カッコ ()), 文字列の先頭、文字列の末尾またはスペースと一致します。

これらの特殊文字を 1 文字のパターンとして入力するには、各文字の前にバックslash (\) を付けて特別な意味を持たないようにします。次の例は、ドル記号、アンダースコア、+ 記号と一致する 1 文字のパターンです。

```
\$_\_+
```

1 文字のパターンの範囲を指定して、コマンドの出力と一致させることができます。たとえば、a、e、i、o、u のいずれかを含む文字列と一致する正規表現を作成できます。パターン マッチングが成功するには、これらの文字のうち 1 文字が文字列に必要です。1 文字のパターンの範囲を指定するには、1 文字のパターンを角カッコ ([ ]) で囲みます。次に例を示します。

```
[aeiou]
```

5 つの母音を示す小文字のアルファベットのいずれかと一致します。

```
[abcdABCD]
```

アルファベットの最初の 4 文字の小文字または大文字と一致します。

範囲の両端をダッシュ (-) でつなぐと、範囲指定を簡単にできます。上記の範囲は次のように簡略化できます。

```
[a-dA-D]
```

ダッシュを 1 文字のパターンとして範囲に入れる場合、ダッシュをもう 1 つ加え、その直前にバックslash (\) を入れます。

```
[a-dA-D\]
```

また、右角カッコ (]) も 1 文字のパターンとして範囲に入れることができます。そのためには、次のように入力します。

```
[a-dA-D\]]
```

上記の例では、アルファベットの最初の 4 文字の小文字または大文字、ダッシュ、右角カッコのいずれかと一致します。

範囲の最初にcaret (^) を入力すると、範囲のマッチング対象が逆になります。次の例では、範囲指定された文字以外のすべての文字と一致します。

```
[^a-dqsv]
```

次の例では、右角カッコ (]) と d を除くすべての文字と一致します。

```
[^\]]d]
```



## 複数文字のパターン

正規表現では、複数の文字を含むパターンを指定することもできます。複数文字の正規表現は、文字、数字、特別な意味がないキーボード文字を組み合わせで作成します。たとえば、「a4%」は複数文字の正規表現です。特別な意味があるキーボード文字の前にバックスラッシュを入力すると、特別な意味がなくなります。

複数文字のパターンでは、順序が重要です。「a4%」という正規表現は、a、4、% の順に並んでいる文字と一致しています。文字列の中に「a4%」という文字列がこの順序で存在しない場合、パターンマッチングは失敗します。次の複数文字の正規表現を見てみましょう。

**a.**

ここでは特別な意味を持つピリオドを使用しています。これは a の後に任意の 1 文字が続く文字列と一致します。この例では、「ab」、「a!」、「a2」はいずれもこの正規表現と一致します。

ピリオドの前にバックスラッシュを入力すると、ピリオドの特別な意味はなくなります。次の表現を見てみましょう。

**a\.**

この場合、「a.」という文字列だけがこの正規表現と一致します。

複数文字の正規表現には、すべての文字、すべての数字、すべてのキーボード文字、または文字、数字、その他のキーボード文字を組み合わせた文字列を使用できます。次の例はいずれも有効な正規表現です。

**telebit 3107 v32bis**

## 繰り返し指定

特別な文字を 1 文字のパターンや複数文字のパターンと組み合わせることにより、指定された正規表現の繰り返しと一致する複雑な正規表現を作成できます。表 1-5 に、正規表現の「繰り返し」を指定する特殊文字を示します。

**表 1-5**                    **繰り返し指定として使用する特殊文字**

文字	説明
*	1 文字または複数文字のパターンが 0 回以上続くことを表す。
+	1 文字または複数文字のパターンが 1 回以上続くことを表す。
?	1 文字または複数文字のパターンが 0 回または 1 回発生することを表す。

次の例では、a が任意の回数 (0 回を含む) 続いている文字列と一致します。

**a\***

次のパターンは、a が 1 回以上続いている文字列と一致します。

**a+**

次のパターンは、文字列 bb または bab と一致します。

**ba?b**

次のパターンは、アスタリスク (\*) が任意の回数続いている文字列と一致します。

**\\*\***

繰り返し指定を複数文字のパターンと組み合わせて使用するには、パターンをカッコで囲みます。次の例のパターンは、「ab」という文字列が任意の回数連続している文字列と一致します。

**(ab)\***

次の例はもっと複雑で、任意のアルファベットと数字の組み合わせが 1 回以上連続している文字列と一致します (0 回、つまり空白は対象とはなりません)。

**([A-Za-z][0-9])+**

繰り返し指定 (\*、+、?) を使用するマッチングでは、最初に最も長い構造体が一致します。入れ子になっている構造体は外側から順番にマッチします。連結されている構造体は、構造体の左端から順番にマッチします。この正規表現では文字が数字の前に指定されているので、「A9b3」とは一致しますが、「9Ab3」とは一致しません。

## 二者択一

選択ではストリングとのマッチングに選択パターンを指定できます。選択パターンは垂直線 (|) で区切ります。選択パターンのうち一方のみが一致します。例として、次の正規表現を見てみましょう。

**codex | telebit**

この場合、「codex」と「telebit」のうち一方の文字列とは一致しますが、「codex」と「telebit」の両方とは一致しません。

## 位置指定

正規表現パターンを文字列の先頭または末尾と一致させることができます。つまり、先頭または末尾に特定のパターンがある文字列を指定できます。表 1-6 に示す特殊文字を使用して、文字列の一部分を「位置指定」できます。

**表 1-6** 位置指定に使用する特殊文字

文字	説明
^	文字列の先頭と一致します。
\$	文字列の末尾と一致します。

次の正規表現は、文字列が「abcd」で始まる場合のみ一致します。

**^abcd**

一方、次の表現の範囲指定は、a、b、c、d を除く任意の 1 文字と一致します。

**[^abcd]**

次の例の表現は、「.12」で終わる文字列と一致します。

**\.12\$**

これらの位置指定文字列は、特殊文字としてのアンダースコア ( \_ ) とは異なります。アンダースコアは文字列の先頭 (^)、文字列の末尾 (\$)、カッコ (( )), スペース ( ), 中カッコ ( { } ), コンマ ( , ), アンダースコア ( \_ ) と一致します。アンダースコアを使用すると、文字列内の任意の位置のパターンを指定できます。

次に例を示します。

`_1300_`

これは文字列内の任意の位置にある「1300」と一致します。文字列の 1300 の前後にスペース、波カッコ、カンマ、またはアンダースコアを置くことができます。次に例を示します。

`{1300_`

この文字列はこの正規表現と一致しますが、21300 や 13000 は一致しません。

アンダースコアを使用すると、次のような多数の正規表現の組を置き換えることができます。

`^1300$ ^1300( ) ( )1300 {1300, ,1300, {1300} ,1300, (1300`

これは次のようにまとめられます。

`_1300_`

## 再使用のためのカッコ

「繰り返し指定」(P.1-9) に示すとおり、カッコと複数文字の正規表現を組み合わせて使用すると、連続回発生するパターンと一致させることができます。この他に、カッコと 1 文字または複数文字のパターンを組み合わせることにより、パターンを正規表現内の別の場所で再使用することができます。

前のパターンを再使用する正規表現を作成するには、再使用する特定のパターンをカッコで示し、バックスラッシュ (\) の後に整数を入力してそのパターンを再使用します。この整数は、正規表現パターン内でのカッコの順序を示しています。正規表現内で複数のパターンを再使用する場合、\1 は再使用する最初のパターン、\2 は 2 番目のパターンとなります。これ以降の整数についても同様です。

次の正規表現では、再使用のためのカッコを使用しています。

`a(.)bc(.)\1\2`

この正規表現で一致するパターンは、「a」の後に任意の文字 (文字 1) が続き、その後に「bc」が続き、その後に任意の文字 (文字 2) が続き、その後にもう一度文字 1 が続き、最後にもう一度文字 2 が続く文字列です。この正規表現では、たとえば `aZbcTZT` が一致します。ここでは文字 1 が Z、文字 2 が T であることが記憶され、正規表現内のその後の位置でこの Z と T が再使用されています。

