基礎プログラミングI

第8回　正規表現

正規表現とは

データから何かを検索するときに、そのデータに含まれている文字列のパターンを指定する方法の一つ。

正規表現の例

「サトウ」と「サイトウ」のどちらかを探したい場合：

サイ？トウ

「サトウ」または「サイトウ」両方にマッチする。“?”を指定すると、「その直前の文字があってもなくてもよい」という意味になる。

正規表現の例

イ[カガ]ラシ

「イカラシ」でも「イガラシ」でもマッチする。“[]”は「その中のどれかの文字が来れば良い」という指定になる。

Rubyで使う正規表現

正規表現“sai?toh”はRubyでは、

/sai?toh/

と表記する。

正規表現の特殊文字

検索のときに特別な意味を持つ文字のことを**メタキャラクタ**という。

代表的なメタキャラクタ

.（ピリオド）

任意の1文字にマッチする。例えば、正規表現ai.awaは“ai”の後ろに何でも良いので何か1文字が来て、その後に“awa”が続くようなもの全てにマッチする。

正規表現	照合する文字列	マッチするか？
/ai.awa/	aikawa	○
	aizawa san	○
	maisawa	○
	aimoto	×
	ai awa	○

代表的なメタキャラクタ

[ ]（大括弧）

その中に列挙した文字のどれか1文字にマッチする。例えば、正規表現[abc]は、文字a, b, cののどれかにマッチする。

正規表現	照合する文字列	マッチするか？
/ai[sz]awa/	aikawa	×
	aizawa san	○
	maisawa	○
	aimoto	×
	ai awa	×

代表的なメタキャラクタ

?

直前のパターンが0回か1回出現する。例えば、正規表現sai?tohは、?の直前がiなので、iが0個でも1個でもよい。

*

直前のパターンが0回以上出現する。例えば、正規表現sai*tohは、*の直前がiなので、iが0個以上なら何個でもよい。正規表現“.*”は、直前のパターンが.(任意の文字)なので、どんな文字列でもマッチする。例えば、「東北.*大学」とすると、「東北」と「大学」の間は「東北～大学」が全てマッチする。

+

直前のパターンが1回以上出現する。

代表的なメタキャラクタ

^

正規表現の先頭に指定した場合のみ、**文字列の先頭**にマッチする。例えば、正規表現^satoは、先頭がsatoから始まるもののみにマッチする。

正規表現	照合する文字列	マッチするか？
/^sato/	sato san	○
	satoh desu	○
	I am sato	×
	Sato!	×

代表的なメタキャラクタ

メタキャラクタ	意味
$	正規表現の末尾に指定した場合のみ、**文字列の末尾**にマッチする。
|（縦棒, パイプ）	**左右のパターンどちらか**にマッチする。
( )	正規表現の一部を括って一つのかたまりにする（グルーピング）。
\	「\」＋「アルファベット1文字」で改行や空白など特殊なパターンを表現する。メタキャラクタ自身を検索するときにも用いる。

正規表現オプション

/ /で括った正規表現の直後に文字を追加すると正規表現の検索方法を変えることができる。例えば、/abcde/iとすると、大文字と小文字を区別せずに検索するようになる。

正規表現オプションの例：

- **i** (大文字と小文字を区別しない)
- **m** (文字列中に改行が含まれていても “.” でマッチするようになる。)

正規表現の指定方法

プログラム中で正規表現を指定するには以下のどれかを使う。

- **/パターン/**
- **%r, パターン,**
- **Regexp.new(パターン文字列)**

%r,パターン,

%r(パターン), %r|パターン|, %r!パターン!でも可。/を多く含む文字列を検索する場合など、複雑な正規表現を指定するときに必要。

Regexp.new(パターン文字列)

文字列を正規表現に変換する。

例えば、

patten = “sai?toh”

という値がありこれを正規表現/sai?toh/に変換したいときは、

regexp = Regexp.new(pattern)

とする。

まとめ

- データから何かを検索するときに、**正規表現**を用いる。
- Rubyでは、正規表現を/パターン/と表記する。
- **メタキャラクタ**を用いて検索方法を指定することができる。