

# Python正規表現reモジュールまとめ @TakaWeekendProg

## 基本メソッド

メソッド	機能	例文
match	文頭の一致	<pre>text = "隣の客はよく柿食う客だ" print(re.match("隣の客", text)) # &lt;re.Match object; span=(0, 3), match='隣の客'&gt;</pre>
search	文字列の一致	<pre>print(re.search("よく柿食う客", text)) # &lt;re.Match object; span=(4, 10), match='よく柿食う客'&gt; print(re.search("よく柿食う客", text).span()) # (4,10)</pre>
fullmatch	文字列の完全一致	<pre>if re.fullmatch("隣の客はよく柿食う客だ", text):     print('OK') # OK</pre>
split	文章の分割	<pre>print(re.split("客", text)) # ['隣の', 'はよく柿食う', 'だ']</pre>
findall	全検索	<pre>print(re.findall("¥w¥w客", text)) # ['隣の客', '食う客']</pre>
sub	文字列の置換	<pre>print(re.sub("柿", "リンゴ", text)) # 隣の客はよくリンゴ食う客だ</pre>
compile	オブジェクト化	<pre>pattern = re.compile('..客') print(pattern.search(text,3)) # &lt;re.Match object; span=(7, 10), match='食う客'&gt; pattern = re.compile('[^あ-ん]') print(pattern.findall(text)) # ['隣の', '客', '柿', '食', '客']</pre>

## 正規表現シンタックス (文法)

シンタックス	効果
[文字列]	[]で囲まれた文字の集合をあらわす
[^文字列]	[^]で囲まれた文字の補集合をあらわす
{数字}	繰り返し回数
{数字,数字}	繰り返し範囲
+	1回以上繰り返し
*	0回以上繰り返し
^	文頭
\$	文末
.	任意の1文字
¥	特殊文字のエスケープ
( )	グループ化
¥s	空白文字
¥S	空白文字以外
¥d	10進数字
¥D	10進数字以外
¥w	英数字と_
¥W	英数字と_以外
[a-z]	アルファベット小文字
[a-zA-Z]	アルファベット
[0-9]	数字
[a-zA-Z0-9_]	英数字と_
[^a-zA-Z0-9_]	英数字以外

## 正規表現によりメールアドレスの記載例

```
^[A-Za-z0-9]{1}[A-Za-z0-9_.-+]*@[([a-zA-Z0-9][a-zA-Z0-9-]*[a-zA-Z0-9]*¥.)+[a-zA-Z]{2,}$
```