

Expressions régulières en pratique

- Bien adaptées à la description de langages :
 - Exemple les littéraux décimaux python
- Des extensions pratiques (qui restent régulières)
 - grep, lex, ...

Exemple 1

- Integer :

$$(1 + \dots + 9)(0 + \dots + 9)^* + 0^+$$

- Pointfloat :

$$(0 + \dots + 9)^*.(0 + \dots + 9)^+ + .(0 + \dots + 9)^+$$

Exemple 2

Commandes linux recherche de motifs, vi, grep, awk,...

Caractères communs aux ERb et ERe

Le tableau suivant liste les caractères spéciaux communs aux **ERb** et aux **ERe**

Caractère spécial	Signification
^	Début de ligne
\$	Fin de ligne
.	Un caractère quelconque
[liste_de_caractères]	Un caractère cité dans la liste
[^liste_de_caractères]	Un caractère qui n'est pas cité dans la liste
*	0 à n fois le caractère ou regroupement précédent
\<expression	Début d'un mot. Caractères pouvant faire partie d'un mot : [A-Za-z0-9_]
expression\>	Fin d'un mot
\<expression\>	Mot complet
\c	Protection du caractère spécial "c"

Caractères spécifiques aux ERe

Caractère spécial	Signification
?	0 ou 1 fois le caractère ou regroupement précédent
+	1 à n fois le caractère ou regroupement précédent
{n}	n fois le caractère ou regroupement précédent
{n,}	Au moins n fois le caractère ou regroupement précédent
{n,x}	Entre n et x fois le caractère ou regroupement précédent
(er1)	Regroupement
er1 er2 er3	Alternatives

Autres applications

- Vérifier les entrées
 - Injection
 - Format d'adresse (un exemple mail)
 - Les dates
- Les langages de programmation offrent tous des outils d'analyse des expressions régulières étendues (exemple librairie python re module)

Nous nous intéressons à la fonction mail de PHP pour les versions d'avant 5.4.42 et 5.5.27.

La fonction se présente comme suit :

```
mail(  
    string $to,  
    string $subject,  
    string $message,  
    array|string $additional_headers = [],  
    string $additional_params = ""  
)
```



The image shows a web form for site feedback. It contains three input fields: 'Your email address*' with the value 'marcus@wahh-mail.com', 'Subject' with the value 'Site problem', and 'Comment*' with the value 'Confirm Order page doesn't load'. Below the fields are two buttons: 'Submit comments' and 'Reset'.

Figure 10-6: A typical site feedback form

Et on peut envoyer un email en utilisant un serveur SMTP comme suit

```
<?php $message = "aaaa"; mail('mail@mail.org', 'Sujet', $message, $headers); ?>
```

en donnant notre adresse email sur le site pour être recontacté

Le champ `additional_headers` n'échappe ni les sauts de lignes, ni le contenu contrairement aux autres champs.

On va faire une injection SMTP en utilisant le champ `additional_headers`
le code ci-dessous

```
<?php $message = "aaaa"; mail('mail@mail.org', 'Sujet', $message, $headers); ?>
```

envoie un mail de la forme

To: mail@mail.org

Subject: Sujet

aaaa

Dans le formulaire qui demande notre adresse email, avec un passage à la ligne `%0d%0a` on peut ajouter le champ Bcc

me@monmail.org%0d%0aBcc:liste@mailinglist.org

Qui passe dans `$headers` devient

To: mail@mail.org

Subject: Sujet

From: me@monmail.org

Bcc:liste@mailinglist.org

aaaa

`%0d%0a` : fin de ligne + début
d'une nouvelle ligne soit `\r`
puis `\n`

The Official Standard: RFC 5322

Maybe you're wondering why there's no "official" fool-proof regex to match email addresses. Well, there is an official definition, but it's hardly fool-proof.

The official standard is known as [RFC 5322](#). It describes the syntax that valid email addresses must adhere to. You can (but you shouldn't—read on) implement it with the following regular expression. RFC 5322 leaves the domain name part open to implementation-specific choices that won't work on the Internet today. The regex implements the "preferred" syntax from [RFC 1035](#) which is one of the recommendations in RFC 5322:

```
\A(?:[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.([a-z0-9!#$%&'*/+=?^_`{|}~-]+)*)|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.|)([a-z0-9](?:[a-z0-9-]*[a-z0-9])?)|\\[(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.)\}{3}|(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\/z
```

Il n'y a pas \x0d ni \x0a, les encodages fin de ligne et retour à la ligne

<https://www.regular-expressions.info/email.html>