
Stage pratique Javascript

Expressions régulières

variables - 1 -

Expression régulière

- L'objet RegExp
 - Manipulation des expressions régulières (des modèles permettant de manipuler des chaînes de caractères)
- Objectifs
 - Trouver des blocs d'une chaîne correspondant au modèle.
 - Tester la conformité d'une chaîne à un modèle
- Exemples
 - Tester un email
 - Tester un numérique
 - ...

variables - 2 -

Syntaxe

- Syntaxe 1
 - Expression = new RegExp("motif", 'options')
- Syntaxe 2
 - Expression = /motif/options
- Options
 - **g** : recherche globale sur la chaîne
 - **i** : recherche non sensible à la casse
 - **gi** : option g + option i.

variables - 3 -

Modèle

- Encoder tous les caractères spéciaux par \
 - \[\] \{ \} \.
- Début par chaîne "^chaîne"
- Fin par chaîne "chaîne\$"
- Abc qui contient la chaîne "Abc«
- Les symboles quantité
 - * 0 à n
 - + 1 à n
 - ? 1 ou 0
- Exemples
 - "abc+": "ab" suivie de n "c"
 - "abc*": "ab" suivie de 0 à n "c"
 - "abc?": "ab" suivie de 0 ou 1 "c"

variables - 4 -

Quantité

- Les accolades $\{X, Y\}$ permettent de donner des limites du nombre d'occurrences.
- Exemples
 - $A\{2\}$ A 2 fois
 - $A\{1,2\}$ A 1 à 2 fois

Parenthèses

- Les parenthèses () représentent une liste de caractères et de traiter le resultat
- Exemples
 - "a(bc)+": "a" suivie de "bc« 1 à n fois
 - "(un|une)": chaîne qui contient "un" ou "une«
 - "domaine\.(net) | (com) | (fr))«
- Caractère Lambda .
 - "a.c": un caractère entre a et c

Liste de caractères

- Liste []

- Exemples

- "[abc]": "a", "b", ou "c".
- "[a-z]": un caractère entre "a" et "z".
- "[^a-zA-Z]": pas de lettre.
- "[ab]": contient un "a" ou un "b"
- "[a-g]": un caractère compris entre "a" et "g"
-

Caractères spéciaux

- \b caractères situés au tout début ou à la fin d'un mot.
 - Par exemple "brelb" permet de capturer "Novembre"
- \d un caractère numérique. \d est équivalent à [0-9].
- \f un saut de page.
- \n un saut de ligne.
- \r un retour chariot.
- \s espace, retour chariot, tabulation, saut de ligne, saut de page
- \t tabulation
- \w caractère alphanumérique , équivalent à [a-zA-Z0-9_].
- \oNombre nombre en base octale (base 8).
- \xNombre nombre en base hexadécimale (base 16).

Blocs

- (...) permet de repérer un bloc, on peut alors dans les instructions de remplacement repérer ce bloc par \$n
 - Exemple inverser des blocs

Fonctions javascript

- `Expression.exec("chaine");`
 - Retourne un Array
- `Expression.test("chaine");`
 - Retourne un boolean
- `chaine.replace(<expr. régul.>, <remplacement>)`
- `Chaine. search(<expr. régul.>)`

variables - 10 -

Exemples

- chaîne="Il y a 3600 secondes dans 1 heure";
 - Extraire les 3600 de la chaîne
 - var reg=new RegExp("[0-9]+","g");
 - reg.exec(chaîne);
- Chaîne='12/11/2000'
 - Tester le format date
 - reg=new RegExp("^([0-9]{2}/[0-9]{2}/[0-9]{4})\$","g");
 - reg.test(chaîne1)
- Remplacer des chaînes
 - var chaîne="Les smartouchs et les smartphones sont beaux";
 - var reg=new RegExp("(smart)","g");
 - chaîne.replace(reg,"SMART");
- Testeur en ligne
 - <http://www.regular-expressions.info/javascriptexample.html>

variables - 11 -

Exemples

- `reg=/^[0-9]{5}$/; // test codepostal`

 - `if(reg.test('12345')==false) ...`

- `reg=/[0-9]/; // test presence nombre`

 - `if(reg.test('eee0rr').value)==true`

- Test email
 - `reg=/^[0-9a-zA-Z]+([_.-]?[0-9a-zA-Z]+)*@[0-9a-zA-Z]+[0-9,a-z,A-Z,..-]*\.(.){1}[a-zA-Z]{2,4}+$/;`