

Expressions régulières

| | | | |
|-------|--|--------|---|
| ... | Ce symbole indique une expression régulière | | |
| ^ | Négation de ce qui suit immédiatement le circonflexe dans l'expression régulière | \d | Chiffre. =[0-9] |
| () | Regroupement par ensemble logique | \D | Non-chiffre. Tout ce qui s'imprime plus \t et \s moins \n |
| [] | Crochets. Ensemble ou intervalle. Séparateur : tiret « - » | \w | Mot = séquence de lettres-chiffres-_. = [0-9azA-Z_] |
| { } | Accolades. Quantificateurs de l'expression qui précède. Séparateur : virgule « , » | \W | Non-mot = [^0-9a-zA-Z_] |
| \ | Barre oblique inverse. Caractère d'échappement : Change la nature du caractère suivant. Littéral ↔ Métacaractère | \s | Espace « blanc » = [\f\n\r\t\v] |
| . | Point. N'importe quel type de caractères sauf une nouvelle ligne | \S | Non-espaces-vides = [^\f\n\r\t\v] |
| | Opérande « OU ». Équivaut à un U en logique | \xhh | Où hh est un nombre hexadécimal d'exactly 2 positions |
| * | 0(zéro) caractère ou plus de l'expression qui précède | \uxxxx | Où xxxx est un Unicode en décimal d'exactly 4 positions |
| + | 1 (un) caractère ou plus de l'expression qui précède | \c¥ | ¥ représente un caractère de contrôle |
| ? | 0(zéro) ou 1(un) caractère de l'expression qui précède | .* | Crée une expression gourmande |
| \$ | Ce qui précède immédiatement le signe de dollar dans l'expression est recherché en fin de ligne | .*? | Tempère l'expression régulière qui précède |
| ^ | Ce qui suit immédiatement le circonflexe dans l'expression est recherché en début de ligne | \e | Escape |
| \b... | L'expression « ... » placée au début d'un mot | \v | Tabulateur vertical |
| ...\b | L'expression « ... » placée à la fin d'un mot | \f | Avancer le chariot d'une ligne = form feed |
| \B... | L'expression « ... » placée ailleurs qu'au début d'un mot | \t | Tabulation horizontale |
| ...\B | L'expression « ... » placée ailleurs qu'à la fin d'un mot | \n | Nouvelle ligne |
| \A... | L'expression « ... » placée au début du fichier | \r | Retour de chariot |
| ...\Z | L'expression « ... » placée à la fin du fichier | | |
| | | | |
| | Métacaractères | | Caractères de catégories |
| | Quantificateurs | | Gourmandise et tempérance |
| | Ancrages | | Caractères spéciaux |

Les 8 cavaliers de Napoléon franchirent les 4 ponts ensemble

| | |
|---|---|
| <p>Avant : signifie que la cible précède le bloc, dans l'ordre séquentiel direct des données</p> <p>Arrière : signifie que le bloc précède la cible, dans l'ordre séquentiel direct des données</p> <p>Positive : la donnée ciblée (la cible) est recherchée à l'endroit indiqué (avant pour les assertions avant et après pour les assertions arrière)</p> <p>Négative : la donnée ciblée (la cible) n'existe pas à l'endroit indiqué (avant pour les assertions avant et après pour les assertions arrière)</p> | |
| Assertions positives | Assertions négatives |
| Assertion avant positive = positive lookahead cible(?=bloc) Faire sortir le 8 seulement en assertion positive \d(?=[\s\w+]+\sNapoléon) Napoléon et les caractères qui précèdent constituent le bloc | Assertion avant négative = Negative lookahead cible(?!bloc) Faire sortir le 4 seulement en assertion négative \d(?![\s\w+]+\sNapoléon) Napoléon et caractères qui précèdent constituent le bloc |
| Assertion arrière positive = Positive lookbehind (?<=bloc)cible Faire sortir le 4 seulement en assertion négative (?<=Napoléon[\s\w+]+\s)\d Napoléon et les caractères qui suivent constituent le bloc | Assertion arrière négative = Negative lookbehind (?<!=bloc)cible Faire sortir le 8 seulement en assertion négative (?<!=Napoléon[\s\w+]+\s)\d Napoléon et caractères qui suivent constituent le bloc |
| | |
| | |
| | |
| | |
| | |

Expressions régulières

| | | | |
|-------|--|--------|---|
| ... | Ce symbole indique une expression régulière | | |
| ^ | Négation de ce qui suit immédiatement le circonflexe dans l'expression régulière | \d | Chiffre. =[0-9] |
| () | Regroupement par ensemble logique | \D | Non-chiffre. Tout ce qui s'imprime plus \t et \s moins \n |
| [] | Crochets. Ensemble ou intervalle. Séparateur : tiret « - » | \w | Mot = séquence de lettres-chiffres-_. = [0-9azA-Z_] |
| { } | Accolades. Quantificateurs de l'expression qui précède. Séparateur : virgule « , » | \W | Non-mot = [^0-9a-zA-Z_] |
| \ | Barre oblique inverse. Caractère d'échappement : Change la nature du caractère suivant. Littéral ↔ Métacaractère | \s | Espace « blanc » = [\f\n\r\t\v] |
| . | Point. N'importe quel type de caractères sauf une nouvelle ligne | \S | Non-espaces-vides = [^\f\n\r\t\v] |
| | Opérande « OU ». Équivaut à un U en logique | \xhh | Où hh est un nombre hexadécimal d'exactly 2 positions |
| * | 0(zéro) caractère ou plus de l'expression qui précède | \uxxxx | Où xxxx est un Unicode en décimal d'exactly 4 positions |
| + | 1 (un) caractère ou plus de l'expression qui précède | \c¥ | ¥ représente un caractère de contrôle |
| ? | 0(zéro) ou 1(un) caractère de l'expression qui précède | .* | Crée une expression gourmande |
| \$ | Ce qui précède immédiatement le signe de dollar dans l'expression est recherché en fin de ligne | .*? | Tempère l'expression régulière qui précède |
| ^ | Ce qui suit immédiatement le circonflexe dans l'expression est recherché en début de ligne | \e | Escape |
| \b... | L'expression « ... » placée au début d'un mot | \v | Tabulateur vertical |
| ...\b | L'expression « ... » placée à la fin d'un mot | \f | Avancer le chariot d'une ligne = form feed |
| \B... | L'expression « ... » placée ailleurs qu'au début d'un mot | \t | Tabulation horizontale |
| ...\B | L'expression « ... » placée ailleurs qu'à la fin d'un mot | \n | Nouvelle ligne |
| \A... | L'expression « ... » placée au début du fichier | \r | Retour de chariot |
| ...\Z | L'expression « ... » placée à la fin du fichier | | |
| | | | |
| | Métacaractères | | Caractères de catégories |
| | Quantificateurs | | Gourmandise et tempérance |
| | Ancrages | | Caractères spéciaux |

Les 8 cavaliers de Napoléon franchirent les 4 ponts ensemble

| | |
|---|---|
| <p>Avant : signifie que la cible précède le bloc, dans l'ordre séquentiel direct des données</p> <p>Arrière : signifie que le bloc précède la cible, dans l'ordre séquentiel direct des données</p> <p>Positive : la donnée ciblée (la cible) est recherchée à l'endroit indiqué (avant pour les assertions avant et après pour les assertions arrière)</p> <p>Négative : la donnée ciblée (la cible) n'existe pas à l'endroit indiqué (avant pour les assertions avant et après pour les assertions arrière)</p> | |
| Assertions positives | Assertions négatives |
| Assertion avant positive = positive lookahead cible(?=bloc) Faire sortir le 8 seulement en assertion positive \d(?=[\s\w+]+\sNapoléon) Napoléon et les caractères qui précèdent constituent le bloc | Assertion avant négative = Negative lookahead cible(?!bloc) Faire sortir le 4 seulement en assertion négative \d(?![\s\w+]+\sNapoléon) Napoléon et caractères qui précèdent constituent le bloc |
| Assertion arrière positive = Positive lookbehind (?<=bloc)cible Faire sortir le 4 seulement en assertion négative (?<=Napoléon[\s\w+]+\s)\d Napoléon et les caractères qui suivent constituent le bloc | Assertion arrière négative = Negative lookbehind (?<!=bloc)cible Faire sortir le 8 seulement en assertion négative (?<!=Napoléon[\s\w+]+\s)\d Napoléon et caractères qui suivent constituent le bloc |
| | |
| | |
| | |
| | |
| | |