# Les expressions régulières

## Création et initialisation

Fichier d'en-tête : #include <regex>

Initialisation : std::regex pattern { “bla bla bla” }; Utilisation des raw string

## Fonctionnalités principales

Sera détaillé par la suite.

- match : vérifier qu'une chaîne donnée correspond à un motif
- recherche : trouve une sous-chaîne correspondant à un motif dans une chaîne
- remplacement : recherche une sous-chaîne et la remplace

Parcourir : regex_iterator et regex_token_iterator. Sera vu plus tard ?

## Introduction au langage des regex

Définitions :

- séquence cible (*Target sequence*) : chaine sur laquelle est appliqué la recherche
- motif (*pattern*) : chaîne dans un langage spéciqug
- correspondance (*match*) : sous-chaîne de la séquence cible qui correspond au motif

## Matching

Vérifier qu'une chaîne correspond à un motif. Par exemple, vérifier un code postal, une date, un nombre, etc.

```
std::regex pattern { R"(\d{5})" }; -> 5 chiffres
```

Caractères génériques :

- . 1 caractère quelconque
- * 0 ou plus
- + 1 ou plus
- ? optionnel (0 ou 1)

Début et fin :

- ^ début
- $ fin

Comptage :

- {n} exactement n fois
- {n,} au moins n fois
- {n,m} entre n et m fois

Groupe et classes [] ()

Classes de caractères (et abréviation)

http://stormimon.developpez.com/dotnet/expressions-regulieres/

| Chapitre précédent | Sommaire principal | Chapitre suivant |
| --- | --- | --- |