# 初めての正規表現

プログラム初心者向けの
正規表現　パート2

作成：Ituki Kirihara/NI

# 検索編

# 正規表現を使った検索

## 正規表現を書いて、その表現にマッチする文字列を探す

- 前回までの知識で、大体検索することが出来ます
- エディタなどの「検索」で、「正規表現で探す」モードにして試してください。
  - 例えば、16進数ぽい文字列を探すには、 0x[0-9A-Fa-f]+

# 最長一致と最短一致

## 正規表現の繰り返し個数を表す部分に?を付け加えると最短一致で探すことが出来ます

- 今までの正規表現の書き方は、最長一致での検索になります
  - 例えば、「abcdabcdabcd」という文字に対して、「aで始まってcで終わる文字列を検索したい場合
    - 「a.*c」と書くと、「abcdabcdabcd」（赤字部分）がヒットします
    - 正規表現はそのまま書くと、正規表現が一致する最大の部分を取り出してしまします。
  - 「aで始まってcで終わる一番短い文字列」を検索したい場合
    - 「a.*?c」と書きます。
    - これで「abcdabcdabcd」がヒットします。

# 最短一致の書き方

( 文字 ) { 整数A , 整数B }?の形式で書きます

**( 文字 ) { 整数A , 整数B }?**

- { 整数A , 整数B }の範囲内で、一番短い文字パターンにマッチします
- { 整数A , 整数B }は省略記号を使うことも出来ます
  - *?や+?、{2,}?なども有効です。??も有効なはずです

# 置換編

# 正規表現を使った置換

## 正規表現を書いて、その表現にマッチする文字列を書き換える

- 今までの知識で、大体置換することが出来ます
- エディタなどの「置換」で、「正規表現で探す」モードにして試してください。
  - 例えば、16進数ぽい文字列を「XXXX」に書き換えるには、検索対象： 0x[0-9A-Fa-f]+ 書き換え文字列：XXXX とします
- 「正規表現でマッチした文字を使って」書き換えることも出来ます。
  - ただし、言語依存部分もありますので、注意してください

# 正規表現を使った置換その２

## 正規表現の()内にマッチしたものをそのまま使って、文字列を書き換えられます

- マッチした文字の中で、使いたい部分を()で囲ってやると、その部分を取り出せる処理系が沢山あります。
  - 取り出す場合の文字は¥1や$1など処理系で変わります
    - 今回は¥1の処理系で書きます
- これを使うと、より高度な置換が出来ます
  - 例えば、16進数っぽい文字列を[ ] で囲みたい場合
    - 検索対象：(0x[0-9A-Fa-f]+) 書き換え文字列 [¥1]
    - 「16進数で0xFFは通常255と同値です」
    - →「16進数で[0xFF]は通常255と同値です」

# 正規表現を使った置換その３

## 正規表現の()内にマッチしたものは、()の９個目ぐらいまでは取り出せます

- 使いたい部分を()で何回か囲ってやると、その部分を順番に取り出せる処理系が沢山あります。
  - 取り出す場合の文字は¥1,¥2,…¥9で取り出せます
    - １０個以上取り出せるかは言語依存です
- これを使うと、さらに高度な置換が出来ます
  - 「16進数で0xFFは通常255と同値です」の0xFFと255を[]で囲みたい場合
    - 検索対象：(0x[0-9A-Fa-f]+)は通常([0-9]+)
    - 書き換え文字列： [¥1]は通常[¥2]
    - 「16進数で0xFFは通常255と同値です」
    - →「16進数で[0xFF]は通常[255]と同値です」

# 置換の時に取り出さない部分

( … )を(?: … )と書くと、正規表現の意味はそのままで、置換の時に取り出さない部分であることを明記できます

- 正規表現は()があちこちに出てきます。が、置換の結果として部分はそんなに多くありません
  - 例： １６進数っぽい文字列を()を沢山つけてみました
    - 正規表現： (0x)(([0-9A-Fa-f])+)は通常([0-9]+)
    - このままだと、¥1 = (0x)部分、¥2 = (([0-9A-Fa-f])+)部分 ¥3 = ([0-9A-Fa-f])部分、¥4 = ([0-9]+)部分となります
      - 言語系によっては、 ¥1 = (0x)部分、¥2 = (([0-9A-Fa-f])+)部分 ¥3 = ([0-9]+)部分となります
    - いらない()を(?:)に置き換えることで、どの部分が何を指すか明確にできます
    - 正規表現： (0x)((?:[0-9A-Fa-f])+)は通常([0-9]+)
      - ¥1 = (0x)部分、¥2 = ((?:[0-9A-Fa-f])+)部分 ¥3 = ([0-9]+)部分となります

# 言語依存の表記について

# 省略文字が存在する言語

一般的には、¥sは空白を、¥dは数字を、¥wは単語構成文字を差すことが多いですが、言語依存です。

- 以下は言語依存の話になります。
- 言語の仕様を確認した上で使ってください。
  - 省略文字(例)
    - ¥s は [ ¥t¥r¥n]
    - ¥d は [0-9]
    - ¥w は [A-Za-z0-9_]
  - 上記はわりとよく見ますが、¥sに全角スペース文字が入っていたり、¥dや¥wに全角文字が入っていたりと、言語によって違うので、使う言語の正規表現の章を参照してください。
  - こんなのも省略文字だったりします
    - ¥p{Alpha} …… [A-Za-z]のこと
    - [:alpha:] …… [A-Za-z]のことだったり、全角文字も入ってたり

これで基礎編は終了です！

# END