UCLA

Introducing Regular Expressions

Howard A. Stahl

Introducing Regular Expressions

- Textual Data Often Follows Very Precise Rules And Formats
- One Rule Can Apply To A Gargantuan Amount Of Data…

Introducing Regular Expressions

- The Expression Language Is Made Up Of Literal Text And MetaCharacters
  - Understanding The Power Of MetaCharacters Is Key

Introducing Regular Expressions

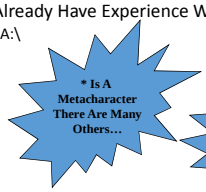
- The Expression Language Is Made Up Of Literal Text And MetaCharacters
  - Understanding The Power Of MetaCharacters Is Key
- Actually, You Already Have Experience With MetaCharacters
  - COPY *.DOC A:\

Introducing Regular Expressions

- The Expression Language Is Made Up Of Literal Text And MetaCharacters
  - Understanding The Power Of MetaCharacters Is Key
- Actually, You Already Have Experience With MetaCharacters
  - COPY *.DOC A:\

Introducing Regular Expressions

- The Expression Language Is Made Up Of Literal Text And MetaCharacters
  - Understanding The Power Of MetaCharacters Is Key
- Actually, You Already Have Experience With MetaCharacters
  - COPY *.DOC A:\



Introducing Regular Expressions

- I'll Describe Just Enough To Handle Our Robot Commands…
  - Character Matching Syntax
  - Quantifier Syntax Supports Repetition
- There Is ALot More Than I'm Describing
  - Things Can Get Very Complex Very Fast…

Character Matching Syntax

- A Text Character Matches Itself
  - A matches A, B matches B, etc.
  - May be concatenated dog matches dog
- **.** Is A MetaCharacter That Matches Any Single Character Except \n
  - b.d matches bad, bid, bed, b3d, bcd, b(d, etc.

Character Matching Syntax

- You Can Specify Sets Of Characters To Match
  - [aeiou] matches a single vowel in the set
  - [^aeiou] matches a character NOT in the set
  - [A-Z] use hyphen to match contiguous ranges
- You Can Also Specify Alternatives
  - (n|s|e|w) matches one of the letters in the set

Quantifier Syntax

- Quantifiers Specify Repetition Of Characters Or Groups
  - + Means One Or More
  - * Means Zero Or More
  - ? Means Zero Or One
  - {#} Means Exactly # Matches
  - {#,} Means AtLeast # Matches
  - {n,m} Means AtLeast N, No More Than M

Online Expression Tester

- A Nice Online Regular Expression Tester
  http://www.regexpal.com/
- Each Line Of Text Will Get Matched Against The Regular Expression
- Blue Text Means The String Matched The Regular Expression Pattern

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++
  `[a-zA-Z_][a-zA-Z_0-9]*`

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++
  `[a-zA-Z_][a-zA-Z_0-9]*`
- Write A Regular Expression For An Positive Integer

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++
  `[a-zA-Z_][a-zA-Z_0-9]*`
- Write A Regular Expression For An Positive Integer
  `[1-9][0-9]*`

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++
  `[a-zA-Z_][a-zA-Z_0-9]*`
- Write A Regular Expression For An Positive Integer
  `[1-9][0-9]*`
- Write A Regular Expression For A Non-Zero Floating Point Number

Regular Expression Examples

- Write A Regular Expression For A Variable Name In C++
  `[a-zA-Z_][a-zA-Z_0-9]*`
- Write A Regular Expression For An Positive Integer
  `[1-9][0-9]*`
- Write A Regular Expression For A Non-Zero Floating Point Number
  `[1-9][0-9]*[.][0-9]*`

Roving Robot Command Regular Expression

- Write A Regular Expression For A Roving Robot Command

Roving Robot Command Regular Expression

- Write A Regular Expression For A Roving Robot Command
- Please Recall The Format….
  - An Optional Leading + or -
  - An Integer (But Not Zero) (And Not More Than Three Digits)
  - A Direction (Either n,s,e,w,ne,nw,se,sw)

Roving Robot Command Regular Expression

- Write A Regular Expression For A Roving Robot Command
- Please Recall The Format….
  - An Optional Leading + or -
  - An Integer (But Not Zero) (And Not More Than Three Digits)
  - A Direction (Either n,s,e,w,ne,nw,se,sw)
- `[+\-]*[1-9][0-9]?[0-9]?(ne|nw|se|sw|n|s|e|w)`