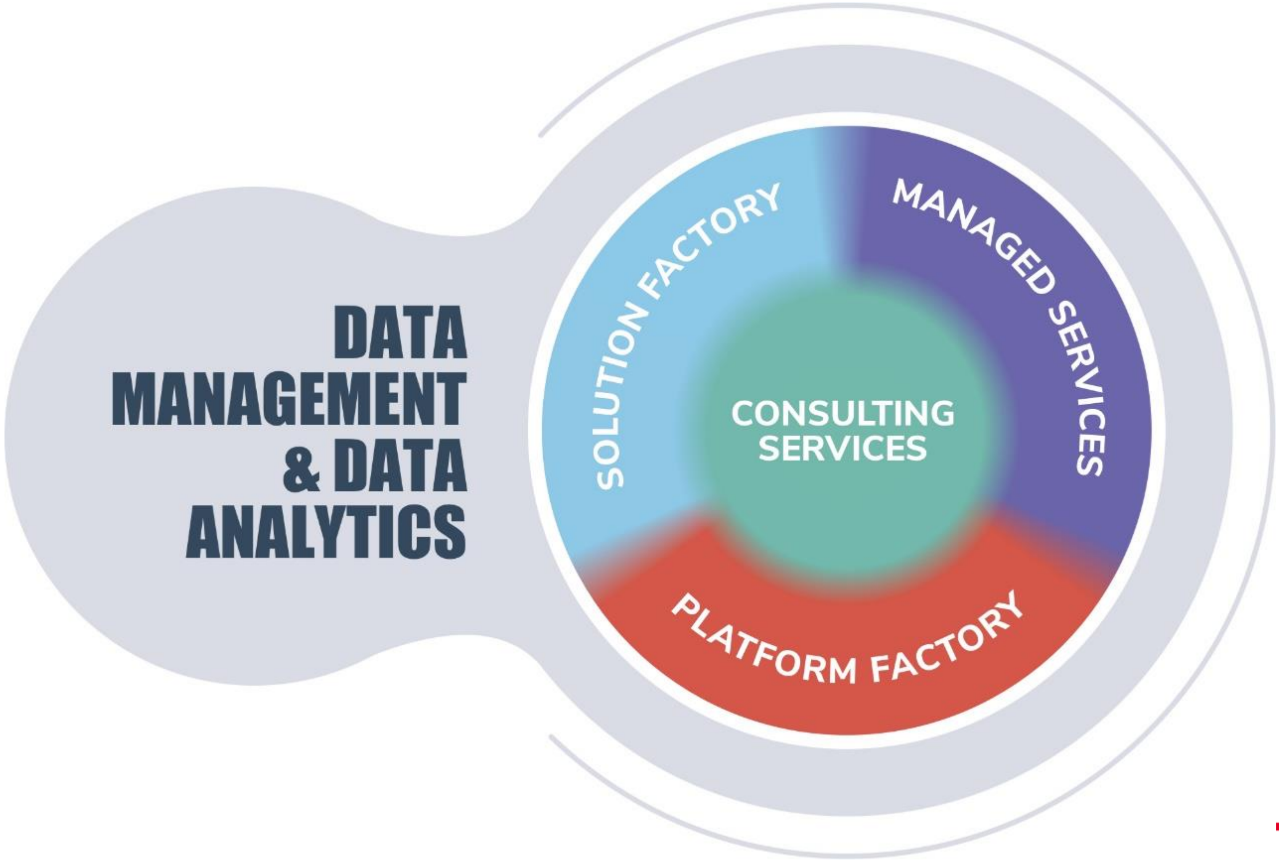
No Fear of Regular Expressions

Philipp Salvisberg

phsalvisberg # DOAG2018

trivadis makes IT easier.

About Us – Added Value from Data



trivadis makes IT easier.

About Me

- Trivadian since April 2000
  - Senior Principal Consultant, Partner
  - Member of the Board of Directors
  - @phsalvisberg
  - https://www.salvis.com/blog
  - https://github.com/PhilippSalvisberg
- Database centric development with Oracle database
- Model Driven Software Development
- Author of free SQL Developer Extensions PL/SQL Unwrapper, PL/SQL Cop, utPLSQL, plscope-utils, oddgen and Bitemp Remodeler

Agenda

Why Use Regular Expressions?

When String Functions Are Not Enough

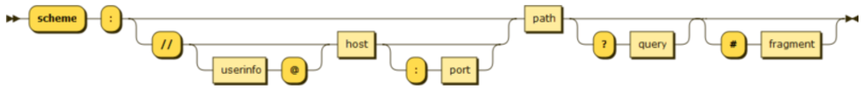
- Deeply nested function calls
  - replace
  - substr
  - instr
- Looping through strings
  - character by character

Use Cases in SQL

- Input Validation, Filter Condition
  - regexp_like
- Find
  - regexp_count, regexp_instr, regexp_substr
- Find & Replace
  - regexp_replace

Additional Use Cases

- Split Input (Tokenizer)
  - – lines, words, columns, …
- Partial Parser
  - – comments, literals, URLs, …



trivadis makes IT easier.

Pattern Matching

trivadis
makes IT easier.

Single Character

Match Pattern

```
t
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

5 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

One Row per Match in SQL

```
WITH
   base AS (
      SELECT q'["Whether you think you can or think you can't ]'
             || '- you are right."'
             || chr(10) || '-- Henry Ford (1863 - 1947)' AS text,
             't' AS pattern
        FROM dual
   )
-- main
 SELECT regexp_substr(text, pattern, 1, level) AS matched_text,
        regexp_instr(text, pattern, 1, level) AS at_pos
   FROM base
CONNECT BY level <= regexp_count(text, pattern)
```

trivadis
makes IT easier.

Multiple Characters

Match Pattern

```
thin
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

2 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Any Character Wildcard – .

Match Pattern

```
c.n
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

2 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Escape Special Characters – \

Match Pattern

```
\.
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

0..1 Matches – Optionality – ?

Match Pattern

```
c?.n
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

5 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

0..n Matches – *

Match Pattern

```
you.*n
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Nongreedy Matches (as few as possible) – ?

Match Pattern

```
you.*?n
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

3 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

1..n Matches – +

Match Pattern

```
-+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

3 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Exact Match – {n}

Match Pattern

```
-{2}
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Match Ranges – {m,n}

Match Pattern

```
-{1,3}
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

3 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Alphanumeric Wildcard – \w

Match Pattern

```
\w+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

17 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Non-alphanumeric Wildcard – \W

Match Pattern

```
\W+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

18 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Digit Wildcard – \d

Match Pattern

```
\d+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

2 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Non-digit Wildcard – \D

Match Pattern

```
\D+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

3 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Whitespace Wildcard (Space, HT, VT, FF, CR, LF) – \s

Match Pattern

```
\s+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

18 matches

```
"Whether you think you can or think you can't - you are right."↩
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Non-whitespace Wildcard – \S

Match Pattern

```
\S+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

19 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis
makes IT easier.

Character Class – [xyz]

to cover umlauts use, e.g.
[[:alpha:]']+

Match Pattern

```
[a-zA-Z']+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

14 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis
makes IT easier.

Negated Character Class – [^xyz]

Match Pattern

```
[^a-zA-Z']+
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

15 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Beginning of Line or String – ^

Match Pattern

```
^-
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

0 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Multiline Mode – m

Match Pattern

```
^-
```

Match Parameter

```
m
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Using Match Parameter in SQL

```
WITH
   base AS (
      SELECT q'["Whether you think you can or think you can't ]'
             || '- you are right."'
             || chr(10) || '-- Henry Ford (1863 - 1947)' AS text,
             '^-' AS pattern,
             'm' AS param
        FROM dual
   )
-- main
 SELECT regexp_substr(text, pattern, 1, level, param) AS matched_text,
        regexp_instr(text, pattern, 1, level, 0, param) AS at_pos
   FROM base
CONNECT BY level <= regexp_count(text, pattern, 1, param)
```

trivadis
makes IT easier.

End of Line or String – $

Match Pattern

```
"$
```

Match Parameter

```
m
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Ignore Case Mode – i

Match Pattern

```
he
```

Match Parameter

```
i
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

3 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Case-sensitive Mode – c

Match Pattern

```
he
```

Match Parameter

```
c
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

2 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

Period Matches Newline Mode – n

Match Pattern

```
.+
```

Match Parameter

```
n
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis
makes IT easier.

Ignore Whitespace in Pattern Mode – x

Match Pattern

```
h  e nr y
```

Match Parameter

```
ix
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

Alternatives – |

Match Pattern

```
think|can't|can
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

4 matches

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis
makes IT easier.

Numbered Groups – (xyz)

Match Pattern

```
^("|')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+(\(((\d+)\s*-\s*(\d+)\))$
```

Text

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

1 match, 9 groups (0=first group/full match, 1=", 2=Whether…right. 3=", …, 8=1947)

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

trivadis makes IT easier.

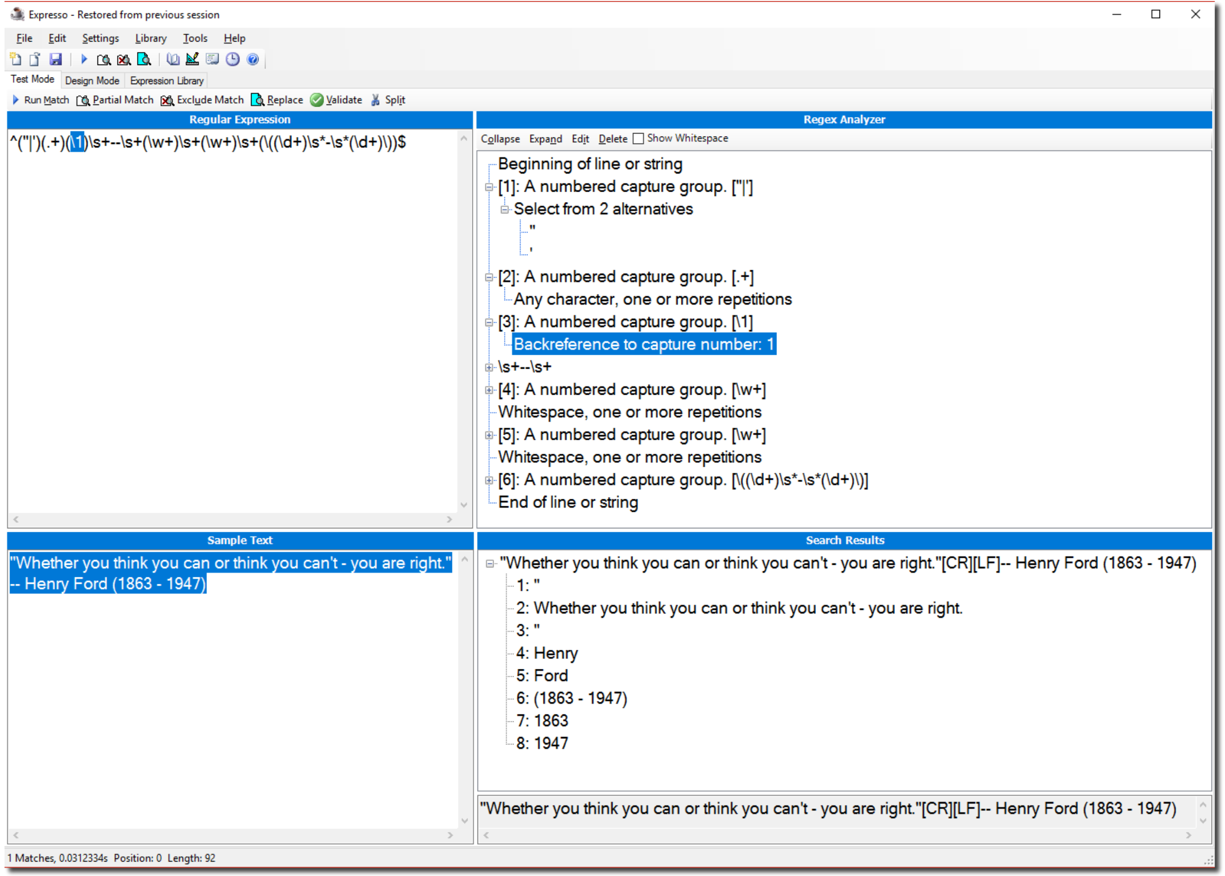
One Row per Group in SQL

```
WITH
   base AS (
      SELECT q'["Whether you think you can or think you can't ]'
             || '- you are right."'
             || CHR(10) || '-- Henry Ford (1863 - 1947)' AS text,
             '^("|'')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+(\((\d+)\s*-\s*(\d+)\))$'
             AS pattern
        FROM dual
   )
-- main
 SELECT level-1 AS group_no,
        regexp_substr(text, pattern, 1, 1, null, level-1) AS matched_group_text
   FROM base
CONNECT BY level <= regexp_count(pattern, '[^\\]?\(') + 1;
```

trivadis makes IT easier.

Tools

Expresso


Expresso - Restored from previous session
File Edit Settings Library Tools Help
Test Mode Design Mode Expression Library
Run Match Partial Match Exclude Match Replace Validate Split
Regular Expression
Regex Analyzer
Collapse Expand Edit Delete Show Whitespace
Sample Text
Search Results

```
^("|')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+(\((\d+)\s*-\s*(\d+)\))$
```

```
Beginning of line or string
[1]: A numbered capture group. ["|']
    Select from 2 alternatives
        "
        '
[2]: A numbered capture group. [.+]
    Any character, one or more repetitions
[3]: A numbered capture group. [\1]
    Backreference to capture number: 1
\s+--\s+
[4]: A numbered capture group. [\w+]
Whitespace, one or more repetitions
[5]: A numbered capture group. [\w+]
Whitespace, one or more repetitions
[6]: A numbered capture group. [\((\d+)\s*-\s*(\d+)\)]
End of line or string
```

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

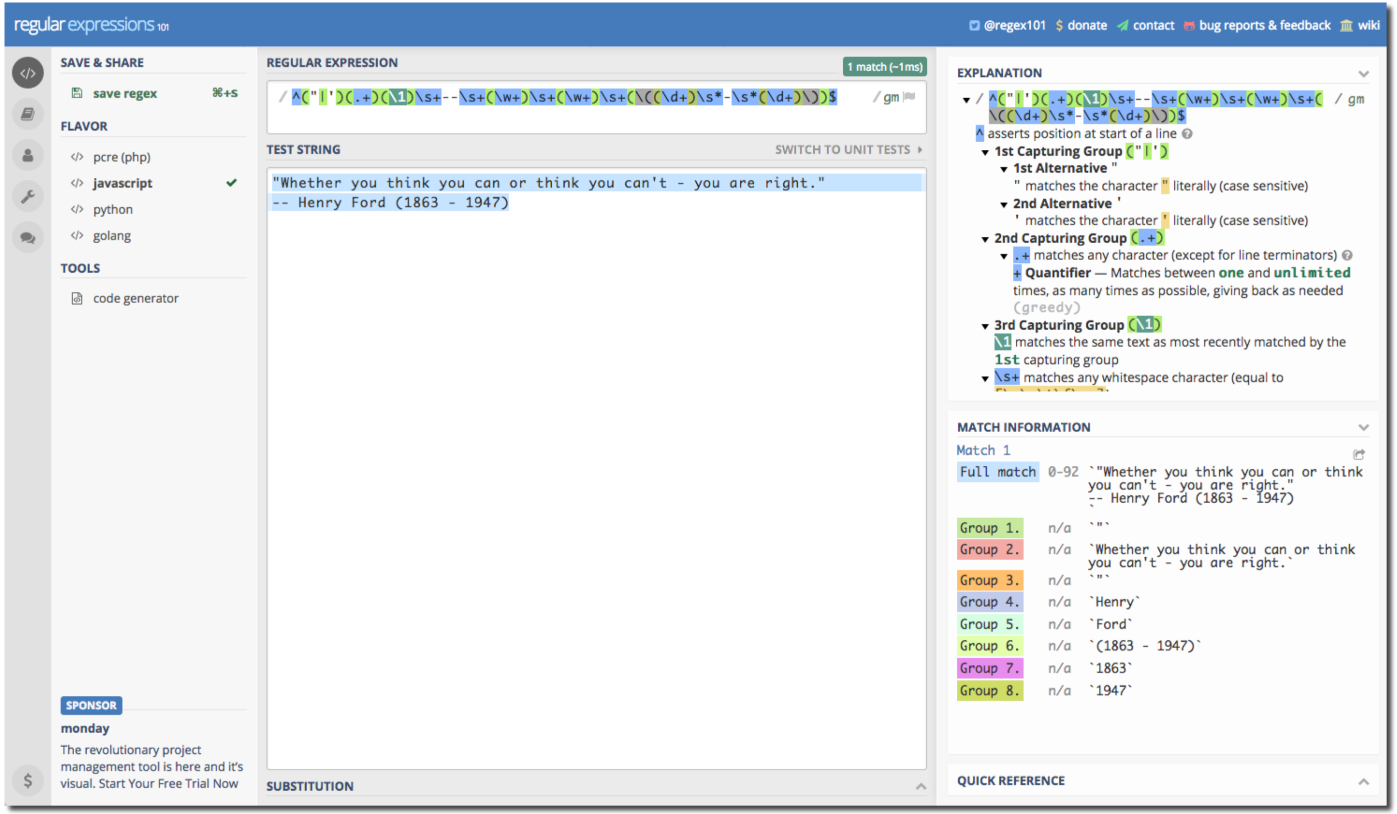
```
"Whether you think you can or think you can't - you are right."[CR][LF]-- Henry Ford (1863 - 1947)
    1: "
    2: Whether you think you can or think you can't - you are right.
    3: "
    4: Henry
    5: Ford
    6: (1863 - 1947)
    7: 1863
    8: 1947
```

```
"Whether you think you can or think you can't - you are right."[CR][LF]-- Henry Ford (1863 - 1947)
```

```
1 Matches, 0.0312334s  Position: 0  Length: 92
```

trivadis
makes IT easier.

regex101.com



monday

The revolutionary project management tool is here and it's visual. Start Your Free Trial Now

REGULAR EXPRESSION — 1 match (~1ms)

```
/ ^("|')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+(\(\(\d+)\s*-\s*(\d+)\))$ / gm
```

TEST STRING

```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

EXPLANATION

```
/ ^("|')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+( / gm
\((\d+)\s*-\s*(\d+)\))$
```

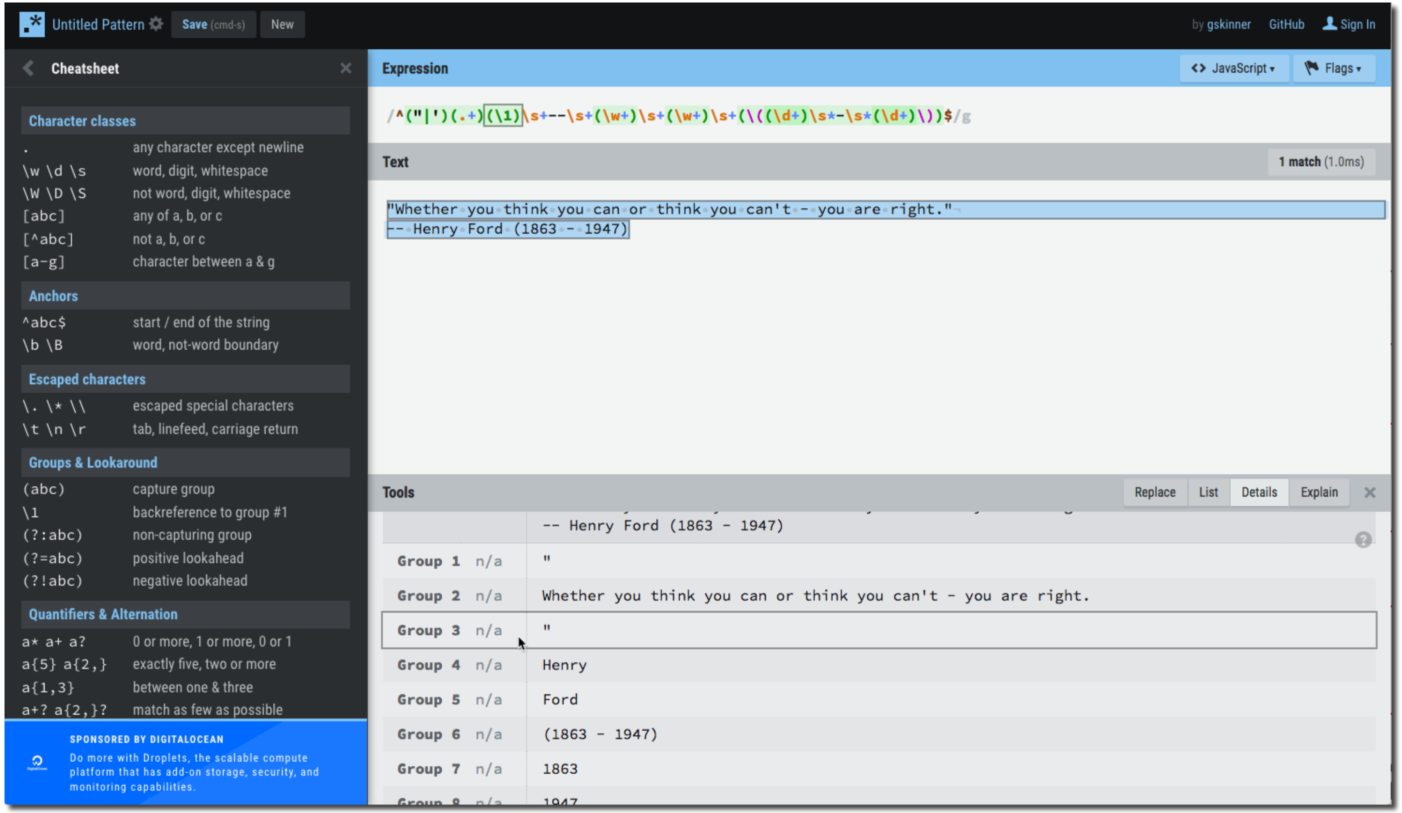
- ^ asserts position at start of a line
  - 1st Capturing Group ("|')
    - 1st Alternative "
      - " matches the character " literally (case sensitive)
    - 2nd Alternative '
      - ' matches the character ' literally (case sensitive)
  - 2nd Capturing Group (.+)
    - .+ matches any character (except for line terminators)
      - + **Quantifier** — Matches between one and unlimited times, as many times as possible, giving back as needed (greedy)
  - 3rd Capturing Group (\1)
    - \1 matches the same text as most recently matched by the 1st capturing group
  - \s+ matches any whitespace character (equal to

MATCH INFORMATION

```
Match 1
Full match  0-92  `"Whether you think you can or think
                  you can't - you are right."
                  -- Henry Ford (1863 - 1947)
                  `
Group 1.    n/a   `"`
Group 2.    n/a   `Whether you think you can or think
                  you can't - you are right.`
Group 3.    n/a   `"`
Group 4.    n/a   `Henry`
Group 5.    n/a   `Ford`
Group 6.    n/a   `(1863 - 1947)`
Group 7.    n/a   `1863`
Group 8.    n/a   `1947`
```

trivadis
makes IT easier.

regexr.com



Character classes

`.`	any character except newline
`\w \d \s`	word, digit, whitespace
`\W \D \S`	not word, digit, whitespace
`[abc]`	any of a, b, or c
`[^abc]`	not a, b, or c
`[a-g]`	character between a & g

Anchors

`^abc$`	start / end of the string
`\b \B`	word, not-word boundary

Escaped characters

`\. \* \\`	escaped special characters
`\t \n \r`	tab, linefeed, carriage return

Groups & Lookaround

`(abc)`	capture group
`\1`	backreference to group #1
`(?:abc)`	non-capturing group
`(?=abc)`	positive lookahead
`(?!abc)`	negative lookahead

Quantifiers & Alternation

`a* a+ a?`	0 or more, 1 or more, 0 or 1
`a{5} a{2,}`	exactly five, two or more
`a{1,3}`	between one & three
`a+? a{2,}?`	match as few as possible

SPONSORED BY DIGITALOCEAN
Do more with Droplets, the scalable compute platform that has add-on storage, security, and monitoring capabilities.

```
/^("|')(.+)(\1)\s+--\s+(\w+)\s+(\w+)\s+(\((\d+)\s*-\s*(\d+)\))$/g
```

1 match (1.0ms)

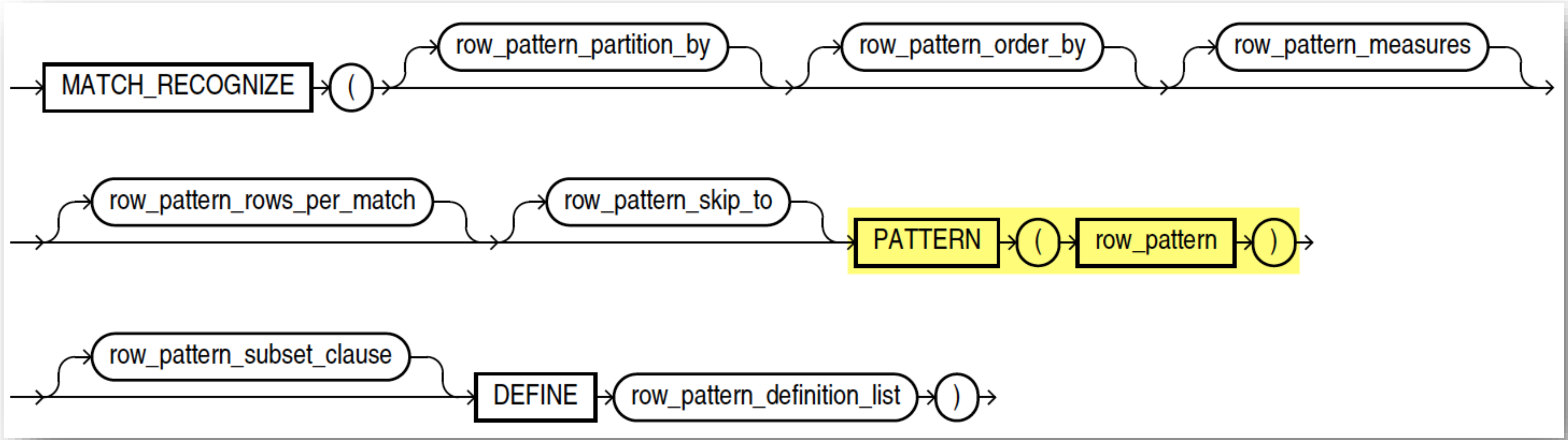
```
"Whether you think you can or think you can't - you are right."
-- Henry Ford (1863 - 1947)
```

		-- Henry Ford (1863 - 1947)
Group 1	n/a	"
Group 2	n/a	Whether you think you can or think you can't - you are right.
Group 3	n/a	"
Group 4	n/a	Henry
Group 5	n/a	Ford
Group 6	n/a	(1863 - 1947)
Group 7	n/a	1863
Group 8	n/a	1947

trivadis makes IT easier.

Row Pattern Matching

row_pattern_clause



Source: SQL Language Reference 18c

trivadis makes IT easier.

Pattern Examples

Known RegEx grammar

- PATTERN (strt up+ down up+)
- PATTERN (^ d $ | ^ i $ | (^ o u $))
- PATTERN ((strt down*)?? up)

Extended grammar

- PATTERN ({- a -} b+ {- c+ -})
- PATTERN (PERMUTE (x{3}, b c?, d))

trivadis makes IT easier.

Core Messages

Simple, but not Self-explanatory

- Strings: t, thin
- Greedy quantifiers: ?, *, +, {2}, {1,3}
- Ungreedy quantifiers: ??, *?, +?, {2}?, {1,3}?
- Character classes: ., \., \w, \W, \d, \D, \s, \S, [a-z], [^a-z]
- Positions: ^, $
- Alternatives: |
- Numbered groups: (xyz), \1, \2, …, \9
- Match parameters: m, i, c, n, x

trivadis makes IT easier.

Use Tools to Build and Understand Complex RegEx

- Expresso, regex101.com, regexr.com, …
- Quick References
- RegEx Libraries
- Explanations

trivadis makes IT easier.

Trivadis @ DOAG 2018 #opencompany

- Booth: 3rd floor – next to the escalator
- We share our know how!
  Simply drop by, live presentations
  and documents archive
- T-Shirts, contest and much more
- We look forward to your visit