PHP - RegEx

RegEx

- Regular Expressions – metoda na odnajdywanie w łańcuchu tekstowym wybranego łańcucha-wzorca.
- Funkcja preg_match() w PHP zwraca wartość 1, jeśli we wskazanym tekście da się odnaleźć wzorzec. 0, jeśli wzorca nie da się znaleźć
- Np. jeśli wzorcem jest „sz”, a łańcuchem tekstowym jest „poduszka”, to funkcja preg_match() zwraca wartość 1.
- Jeśli wzorcem jest „ga”, a łańcuchem tekstowym jest „pudło”, to funkcja preg_match() zwraca wartość 0.

Budowanie wyrażenia RegEx

- Wyszukiwany wzorzec warto przypisać do nowej zmiennej. Szukany wzorzec powinno umieścić się pomiędzy symbolami delimiterów, np. ukośników.
- Wewnątrz delimiterów można zbudować dowolne żądane wyrażenie-wzorzec, np.:

  $wzor = ’’/ok/’’;

- RegEx pozwala na zbudowanie wyrażeń korzystających z symboli „uniwersalnych”.

Wyrażenia i symbole RegEx - zbiory

Symboli [ ] można użyć, aby zbudować zbiór dopuszczalnych wartości:

- [abc] – znajdź literę a, b lub c
- [a-z] – znajdź literę od a do z (małe litery)
- [A-Z] – znajdź literę od A do Z (wielkie litery)
- [a-zA-Z] – znajdź dowolną literę
- [0-9] – znajdź dowolną cyfrę
- [^0-9] – znajdź dowolny symbol, który **nie** jest cyfrą (symbol ^ oznacza tu negację)

Wyrażenia i symbole RegEx – znaki specjalne

- . – znajdź dowolny symbol
- ^ - znajdź na początku tekstu
- $ - znajdź na końcu tekstu
- \d – znajdź dowolną cyfrę
- \D – znajdź dowolną nie-cyfrę
- \s – znajdź dowolną spację lub odstęp
- \S – znajdź dowolny znak, niebędący spacją lub odstępem
- \w – znajdź dowolną literę lub cyfrę
- \W – znajdź dowolny znak, niebędący literą lub cyfrą
- ?= - znajdź „gdzieś w tekście”, tzw. „positive lookahead”

Wyrażenia i symbole RegEx – symbole liczące

- a+ - znajdź tekst, w którym jest minimum jedno „a”
- a* - znajdź tekst, w którym jest 0 lub więcej „a”
- a? – znajdź tekst, w którym jest 0 lub 1 „a”
- a{2} – znajdź tekst, w którym są 2 „a” pod rząd
- a{2, 5} – znajdź tekst, w którym są minimum 2, ale maksimum 5 „a” pod rząd
- a{2,} – znajdź tekst, w którym są minimum 2 „a” pod rząd

Jak sprawdzić, czy hasło jest bezpieczne?

Założenia:

- minimum 10 znaków,
- minimum jedna wielka litera,
- minimum jedna mała litera,
- minimum jedna cyfra,
- minimum jeden znak specjalny

TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 secs	22 mins	1 hour	8 hours
9	Instantly	2 mins	19 hours	3 days	3 weeks
10	Instantly	58 mins	1 month	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15 bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	6tn years	100 tn years	7qd years

HIVE SYSTEMS

-Data sourced from HowSecureismyPassword.net

Jak sprawdzić, czy hasło jest bezpieczne?

- Minimum 1 wielka litera w haśle:

  (?=.*?[A-Z])

  znajdź gdzieś w tekście 0 lub więcej dowolnych symboli
  obok których będzie dowolna wielka litera

- Minimum 1 mała litera w haśle:

  (?=.*?[a-z])

  znajdź gdzieś w tekście 0 lub więcej dowolnych symboli
  obok których będzie dowolna mała litera

Jak sprawdzić, czy hasło jest bezpieczne?

- Minimum 1 cyfra w haśle:

  (?=.*?[0-9])

  znajdź gdzieś w tekście 0 lub więcej dowolnych symboli
  obok których będzie dowolna cyfra

- Minimum 1 znak specjalny w haśle:

  (?=.*?[\!\@\#\$\%\^\&*\-\?])

  znajdź gdzieś w tekście 0 lub więcej dowolnych symboli
  obok których będzie znak specjalny z tej grupy

Łącząc wszystko razem

(?=.*?[A-Z]) (?=.*?[a-z]) (?=.*?[0-9]) (?=.*?[\#\?\!\@\$\%\^\&*\-])

dowolna wielka litera

dowolna mała litera

dowolna cyfra

dowolny znak specjalny

Kombinacja symboli ?=.*? pozwala na znalezienie dowolnej ilości dowolnych innych znaków (w tym także żadnego), poprzedzających poszukiwaną wartość. Dzięki temu, nie trzeba zwracać uwagi na kolejność wielkich i małych znaków w haśle.

Ale to jeszcze nie koniec

/^(?=.*?[A-Z]) (?=.*?[a-z]) (?=.*?[0-9]) (?=.*?[#?!@$%^&*-]).{10,}$/

dowolna wielka litera

dowolna mała litera

dowolna cyfra

dowolny znak specjalny

.{10,} oznacza: znajdź w sumie minimum 10 dowolnych symboli

^ na początku i $ na końcu można rozumieć jako „pomiędzy startem, a końcem tego łańcucha tekstowego”

Zatem całe to wyrażenie ma znaleźć po jednym poszukiwanym symbolu (wielka, mała, cyfra, symbol), a w sumie to ma w tym tekście być ich minimum 10.

Można to rozbić na kilka kolejnych sprawdzeń

- Utwórz kilka zmiennych, którym przypiszesz wzorce sprawdzające każdy z warunków osobno:
  - zmienna ze wzorcem na wielkie litery
  - zmienna ze wzorcem na małe litery
  - zmienna ze wzorcem na cyfry
  - zmienna ze wzorcem na znaki specjalne (każdy znak specjalny w tym zakresie poprzedź symbolem backslash \ (np. [\!\@\#\$...]
  - zmienna ze wzorcem na 10 symboli dowolnych
- Wykorzystaj funkcję preg_match, aby sprawdzić, czy hasło spełnia wszystkie 5 warunków.

Sprawdzacz haseł

Podaj hasło: SilneHaslo123$

Sprawdź swoje hasło

Twoje hasło to: SilneHaslo123$

Mała litera: 1

Wielka litera: 1

Cyfra: 1

Znak specjalny: 1

Długość: 1