var Wyrazenie = new RegExp("pani?" , "gi")

lub

var Wyrazenie = /pani?/gi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metaznak** | **Znaczenie** | **Przykład wyrażenia** | **Zgodne ciągi z wyrażeniem** | **Niezgodne ciągi z wyrażeniem** |
| ^ | początek wzorca | ^za | zapałka, zadra, zapłon, zarazek | kazanie, poza, bazar |
| $ | koniec wzorca | az$ ^.arka$ | uraz, pokaz barka, warka | azymut, pokazy parkan |
| . | dowolny pojedynczy znak | .an.a | panda, Wanda, panna, kania | rana, konia |
| [...] | dowolny z wymienionych znaków; możemy podawać kolejne znaki lub wpisywać zakres - na przykład [a-z] oznacza wszystkie małe litery. Wymieniając specjale znaki z końca tej tabeli nie musimy poprzedzać znakiem \ | [a-z]an[nd]a [a-z][a-zA-Z0-9.-][pus] | pana, panda, wanna pas, mAs, p2p, m3u, b-s, z.u | Wanda, kania Bas, bal, balu, mp3 |
| [^...] | dowolny z niewymienionych znaków | kre[^st] | krew, krem | kres, kret |
| | | dowolny z rozdzielonych znakiem ciągów | [nz]a|pod|przed trzynasty|13-ty|13 | na, za, pod, przed trzynasty, 13-ty, 13 |  |
| (...) | zawężenie zasięgu | g(ż|rz)eg(ż|rz)(u|ó)łka (ósmy|8-my|8)(maj|maja) | gżegżółka, gżegrzółka, gżegrzułka, grzegrzułka ósmy maja, 8-my maj, 8 maja |  |
| ? | zero lub jeden poprzedzający znak lub element; elementem może być na przykład wyrażenie umieszczone wewnątrz nawiasów (...) | ro?uter (ósmy|8(-my)?)maja? | router, ruter ósmy maja, ósmy maj, 8-mymaja, 8-my maj, 8 maja, 8 maj |  |
| + | jeden lub więcej poprzedzających znaków lub elementów; elementem może być na przykład wyrażenie umieszczone wewnątrz nawiasów (...) | [0-9]+[abc] pan+a (tam)+ | 10a, 1b, 003c, 42334b pana, panna, pannnna tam, tamtam, tamtamtam | a, b, c, z, 14, 03, 12d, 1231z paa, panda, ta, tamta, mat |
| \* | zero lub więcej poprzedzających znaków lub elementów; elementem może być na przykład wyrażenie umieszczone wewnątrz nawiasów (...) | [0-9]\*[abc] pora\*n\*a\* | 10a, 1b, 003c, 42334b, a, b, c por, poa, poranna, poraannnaa, pornnna | k, 2335, porada, panna |
| {4} | dokładnie 4 poprzedzające znaki lub elementy | [0-9]{4} | 8765, 8273, 2635 | 12345, 234, 2123456 |
| {4,} | 4 lub więcej poprzedzających znaków lub elementów | [ah]{4,} | haha, haaaaahaha, ahaaa | haa, ha, hehe, aha |
| {2,4} | od 2 do 4 poprzedzających znaków lub elementów | p.{2,4}a | piana, pola, polana | psa, poranna |
| \. | znak kropki | [0-9]{,3}\.[0-9]{,3}\.[0-9]{,3} | 128.0.0.2 | 128-0-0-2 |
| \\* | znak \* | \\*.+ | \*nic | nic\*, nic |
| \/ | znak / | ^\/\/$ | // |  |
| \? | znak ? | ^.+\?$ | Czy to jest kot? | Czy to jest kot |
| \: | znak : | ^.+\:$ | Oto one: | :nic |
| \. | znak . | \.+ | ...... |  |
| \^ | znak ^ | .\*\^ | To jest ^ | To jest & |
| \+ | znak + | [0-9]+\+[0-9]+ | 928374+29832 | 23873-32787 238738278 |
| \\ | znak \ | c\:\\ | c:\ |  |
| \= | znak = | [0-9]+\+[0-9]+\=[0-9]+ | 11+12=23 | 11+12+23 |
| \| | znak | | x \|\| y | x || y |  |

**Flagi**

Poza wymienionymi metaznakami istnieją specjalne parametry (flagi), które oddziałują na wyszukiwanie wzorców.:

[?](http://www.doman.art.pl/kursjs/kurs/regular.html)

|  |  |
| --- | --- |
| 1  2  3 | var Wyrazenie = /[a-z]\*/mg  var Wyrazenie = new RegExp("[a-z]\*","g") |
| **znak Flagi** | **znaczenie** |
| i | powoduje niebranie pod uwagę wielkości liter |
| g | powoduje zwracanie wszystkich psujących fragmentów, a nie tylko pierwszego |
| m | powoduje wyszukiwanie w tekście kilku linijkowym. W trybie tym znak początku i końca wzorca (^$) jest wstawiany przed i po znaku nowej linii (\n). |
| s | powoduje włączenie przeszukiwania jedno linijkowego. W trybie tym znak . oznacza nową linijkę |

**Klasy znaków**

Dodatkowo Javascript udostępnia specjalne klasy znaków. Zamiast wyszukiwać wszystkich liter za pomocą [a-zA-Z\_] możemy skorzystać z klasy znaków **\w**.

|  |  |
| --- | --- |
| **Klasa znaków** | **znaczenie** |
| \s | znak spacji, tabulacji lub nowego wiersza |
| \S | znak nie będący spacją, tabulacją lub znakiem nowego wiersza |
| \w | każdy znak będący literą, cyfrą i znakiem \_ |
| \W | każdy znak nie będący literą, cyfrą i znakiem \_ |
| \d | każdy znak będący cyfrą |
| \D | każdy znak nie będący cyfrą |

Metoda **test()** służy do sprawdzania, czy dane wyrażenie znajduje się w tekście:

var tekst = "Fantomas robi mase - marchewkowo-marcepanowa";

var wzor = /mas/g;

var tablicaWynikow = tekst.match(wzor);

if (tablicaWynikow.length) {

    console.log(tablicaWynikow.join('-'));

}

Aby wyszukać w tekście kod pocztowy uzyjemy wyrazenia:

[?](http://www.doman.art.pl/kursjs/kurs/regular.html)

|  |  |
| --- | --- |
| 1  2  3  4 | var WzorKoduP = /[0-9]{2}-[0-9]{3}/g;  //lub  var WzorKoduP = /[\d]{2}-[\d]{3}/g; |

**[0-9]{2}** - powinny znaleźć się 2 cyfry  
**-** - po 2 cyfrach powinien znaleźć się znak -  
**[0-9]{3}** - po kturym powinny znaleźć się 3 cyfry  
**g** - mają być zwrócone wszystkie wyszukane pasujące ciągi

Nasz wzór możemy wykorzystać w skrypcie za pomocą metody **match** (zwracającej pasujące ciągi):

[?](http://www.doman.art.pl/kursjs/kurs/regular.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | var tekst = "Moj kod pocztowy to nie 12-323 ani tez 03-400 ... jest po prostu inny.";  var WzorKoduP = /[0-9]{2}-[0-9]{3}/g; //parametr g nakazuje zwrócenie wszystkich znalezionych ciągów (normalnie zwracany jest  tylko pierwszy)  var a = tekst.match(WzorKoduP);  if (a) {//jeżeli w a znajdują się pasujące ciągi      for (x=0; x<a.length; x++) {          document.write(a[x]+"<br />");      }  } |