Chat Log BootCamp JavaScript 2019_11_27 22_04

Antonio Contreras (to Everyone): 18:47: muy buenas Ferran

Ferran Puig Martínez (to Everyone): 18:47: Buenas tardes :)

[Lemoncode] Dani (to Everyone): 18:47: Muy buenas

Miguel Ángel Vázquez Vera (to Everyone): 18:52: Hola a todos!

Antonio Contreras (to Everyone): 18:52: buenas Miguel

Miguel Ángel Vázquez Vera (to Everyone): 18:52: Poco a poco

Miguel Ángel Vázquez Vera (to Everyone): 18:53: seguro que sí

Ferran Puig Martínez (to Everyone): 18:53: Bien, con mucho mono :P

Miguel Ángel Vázquez Vera (to Everyone): 18:54: Sí, para darle al coco

Miguel Ángel Vázquez Vera (to Everyone): 18:54: Normalmente no empezamos hasta y 5 o así

Miguel Ángel Vázquez Vera (to Everyone): 18:56: Yo soy artista. Lo poco que sé de programación lo he dado en este bootcamp

Miguel Ángel Vázquez Vera (to Everyone): 18:56: Xd

Ferran Puig Martínez (to Everyone): 18:56: Soy maquetador web, y bueno un poco de allí y de allá, más de javascript que otra cosa, pero muy autodidacta

Ferran Puig Martínez (to Everyone): 18:56: si jeje

Ferran Puig Martínez (to Everyone): 18:57: pero se necesita una buena base que si no...

Miguel Ángel Vázquez Vera (to Everyone): 18:57: Pues a tope

Miguel Ángel Vázquez Vera (to Everyone): 18:57: con javascript

Miguel Ángel Vázquez Vera (to Everyone): 18:58: Sí, mola mucho

Antonio Contreras (to Everyone): 18:58: te picas y te tiras horas xD

Ferran Puig Martínez (to Everyone): 18:58: La verdad que la formación de fp deja mucho que desear :(y la uni demasiada paja para depende que edad, lo mejor son bootcamps y cursos, almenos en mi opinión

Miguel Ángel Vázquez Vera (to Everyone): 18:59: Sí, completamente de acuerdo

Marcos García F. (to Everyone): 18:59: Buenas!! quizás lo sepa más Javier ¿pero quedaría algo de teoría de algoritmos III?

Marcos García F. (to Everyone): 18:59: q tal!

Antonio Contreras (to Everyone): 18:59: no ya se do todo de algoritmos 3

Antonio Contreras (to Everyone): 19:00: dio*

Marcos García F. (to Everyone): 19:00: no ejercicio no Antonio?

Amaia (to Everyone): 19:00: Buenas!!

Marcos García F. (to Everyone): 19:00: OK

Miguel Ángel Vázquez Vera (to Everyone): 19:01: xD

Marcos García F. (to Everyone): 19:01: si va saliendo la cosa

Ferran Puig Martínez (to Everyone): 19:01: Hoy creo que nos va a rebentar la cabeza jeje

Miguel Ángel Vázquez Vera (to Everyone): 19:02: Oye, Dani preséntate tú también

Miguel Ángel Vázquez Vera (to Everyone): 19:02: dani o daniel?

Miguel Ángel Vázquez Vera (to Everyone): 19:02: ok

stefano durante (to Everyone): 19:02: hola

Miguel Ángel Vázquez Vera (to Everyone): 19:03: Estupendo :)

Antonio Contreras (to Everyone): 19:03: jajaja

Miguel Ángel Vázquez Vera (to Everyone): 19:03: Antonio, tú tampoco te presentaste Xd

Miguel Ángel Vázquez Vera (to Everyone): 19:03: no?

Marcos García F. (to Everyone): 19:03: Hace falta estudiar teleco para programar bien bien??? jeje

Ana Gomez (to Everyone): 19:04: Buenas tardes

Antonio Contreras (to Everyone): 19:04: en lemoncode algunos venimos de fp

Marcos García F. (to Everyone): 19:04: vale me tranquiliza

Marcos García F. (to Everyone): 19:04: ok Antonio !

Ferran Puig Martínez (to Everyone): 19:04: si

Ana Gomez (to Everyone): 19:04: si

Antonio Contreras (to Everyone): 19:07: pues lo que ha dicho Dani darle caña preguntar dudas que es muy bueno y tiene paciencia

Gema Pujalte Iglesias (to Everyone): 19:08: Hola! Me perdí mucho?

Gema Pujalte Iglesias (to Everyone): 19:08: menos mal..que hoy pinta dificil

Antonio Contreras (to Everyone): 19:09: chicos dejo firepad por si necesitamos compartir https://demo.firepad.io/#In783j7gUq

Miguel Ángel Vázquez Vera (to Everyone): 19:09: gracias, Antonio

Antonio Contreras (to Everyone): 19:09: de nada!!!

Miguel Ángel Vázquez Vera (to Everyone): 19:10: Xd

Ferran Puig Martínez (to Everyone): 19:10: super

Ferran Puig Martínez (to Everyone): 19:11: parece klingon XD

Miguel Ángel Vázquez Vera (to Everyone): 19:11: buah

Antonio Contreras (to Everyone): 19:12: preguntas ?

Miguel Ángel Vázquez Vera (to Everyone): 19:12: No

Marcos García F. (to Everyone): 19:12: va bien

stefano durante (to Everyone): 19:12: de momento no

Oscar Moreno (to Everyone): 19:13: se basa en caracteres comodin?

Ferran Puig Martínez (to Everyone): 19:14: si

stefano durante (to Everyone): 19:14: si

Miguel Ángel Vázquez Vera (to Everyone): 19:14: Mmmm, tú sigue

Amaia (to Everyone): 19:14: si

Miguel Ángel Vázquez Vera (to Everyone): 19:16: la primera xD

Ferran Puig Martínez (to Everyone): 19:16: buff jeje

Marcos García F. (to Everyone): 19:16: la anterior

Oscar Moreno (to Everyone): 19:16: est en arameo

Paloma Robles Lopez (to Everyone): 19:16: con el if

stefano durante (to Everyone): 19:16: la primera

Ferran Puig Martínez (to Everyone): 19:16: la primera

[Lemoncode] Dani (to Everyone): 19:17: https://stackoverflow.com/questions/940577/javascript-regular-expression-email-validation?lq=1

Miguel Ángel Vázquez Vera (to Everyone): 19:17: En principio, no

Ana Gomez (to Everyone): 19:17: por ahora no

Miguel Ángel Vázquez Vera (to Everyone): 19:20: ok

Marcos García F. (to Everyone): 19:20: pero solo uno no?

Marcos García F. (to Everyone): 19:20: ok

stefano durante (to Everyone): 19:20: solo números y letras?

stefano durante (to Everyone): 19:20: ok

Miguel Ángel Vázquez Vera (to Everyone): 19:21: Interesante

stefano durante (to Everyone): 19:21: gracias

[Lemoncode] Dani (to Everyone): 19:21: https://regexr.com/

[Lemoncode] Dani (to Everyone): 19:21: https://www.regexpal.com/

Ferran Puig Martínez (to Everyone): 19:22: si

Ana Gomez (to Everyone): 19:22: si
Miguel Ángel Vázquez Vera (to Everyone): 19:22: sí
stefano durante (to Everyone): 19:22: si
Miguel Ángel Vázquez Vera (to Everyone): 19:22: tpl
Ferran Puig Martínez (to Everyone): 19:22: tpl
Ana Gomez (to Everyone): 19:23: tpl!
stefano durante (to Everyone): 19:23: tpl
Miguel Ángel Vázquez Vera (to Everyone): 19:23: salto de línea
Miguel Ángel Vázquez Vera (to Everyone): 19:23: ?
Paloma Robles Lopez (to Everyone): 19:24: salto de linea
Ferran Puig Martínez (to Everyone): 19:24: \n?
Antonio Contreras (to Everyone): 19:24: genial
Ana Gomez (to Everyone): 19:24: si
Miguel Ángel Vázquez Vera (to Everyone): 19:24: sí tpl
Amaia (to Everyone): 19:24: si
stefano durante (to Everyone): 19:24: tpl
Ferran Puig Martínez (to Everyone): 19:24: si, muy guai esta tool
Ferran Puig Martínez (to Everyone): 19:26: y que ventaja tiene?
Ferran Puig Martínez (to Everyone): 19:26: es más largo, no?
Marcos García F. (to Everyone): 19:26: es un objeto de js no?
Antonio Contreras (to Everyone): 19:26: si
Marcos García F. (to Everyone): 19:26: por el new
Miguel Ángel Vázquez Vera (to Everyone): 19:27: nop
stefano durante (to Everyone): 19:27: tpl
Ferran Puig Martínez (to Everyone): 19:27: tpl :)
Antonio Contreras (to Everyone): 19:27: guay
Miguel Ángel Vázquez Vera (to Everyone): 19:27: no
Marcos García F. (to Everyone): 19:27: si
stefano durante (to Everyone): 19:27: si
[Lemoncode] Dani (to Everyone): 19:27: https://codepen.io/pen/
Ana Gomez (to Everyone): 19:28: si
Manuel Larena Anchuela (to Everyone): 19:28: si
stefano durante (to Everyone): 19:29: si
Miguel Ángel Vázquez Vera (to Everyone): 19:29: tpl
Iván (to Everyone): 19:29: si
Ferran Puig Martínez (to Everyone): 19:29: si
Miguel Ángel Vázquez Vera (to Everyone): 19:30: tpl
Ana Gomez (to Everyone): 19:30: tpl
Marcos García F. (to Everyone): 19:30: tpl
[Lemoncode] Dani (to Everyone): 19:30: const myValueOk = "imagenA";
const myValueNotOk = "imag";
const pattern = /imagen./;
Miguel Ángel Vázquez Vera (to Everyone): 19:31: tpl
stefano durante (to Everyone): 19:31: tpl
Antonio Contreras (to Everyone): 19:31: vamos vamos jaja
Ana Gomez (to Everyone): 19:31: tpl
[Lemoncode] Dani (to Everyone): 19:33: const resultOk = pattern.test(myValueOk);
console.log(resultOk);

stefano durante (to Everyone): 19:33: tpl

Miguel Ángel Vázquez Vera (to Everyone): 19:33: tpl

[Lemoncode] Dani (to Everyone): 19:34:
```
const myValueOk = "imagenA";
const myValueNotOk = "imag";
const pattern = /imagen./;

const resultOk = pattern.test(myValueOk);
console.log(resultOk);

const resultNotOk = pattern.test(myValueNotOk);
console.log(resultNotOk);
```

Marcos García F. (to Everyone): 19:34: digamos que test es un método de RegExp?

Antonio Contreras (to Everyone): 19:34: exacto

Marcos García F. (to Everyone): 19:35: guay

[Lemoncode] Dani (to Everyone): 19:35: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp

Ferran Puig Martínez (to Everyone): 19:35: leyendo bien, jeje

[Lemoncode] Dani (to Everyone): 19:37: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

Miguel Ángel Vázquez Vera (to Everyone): 19:37: tpl

stefano durante (to Everyone): 19:37: tpl

Ana Gomez (to Everyone): 19:37: tpl

[Lemoncode] Dani (to Everyone): 19:38: holaimagen0

Ferran Puig Martínez (to Everyone): 19:39: ^

Antonio Contreras (to Everyone): 19:39: muy bien

[Lemoncode] Dani (to Everyone): 19:39: /^imagen./

Miguel Ángel Vázquez Vera (to Everyone): 19:40: tpl

Ana Gomez (to Everyone): 19:40: tpl

Manuel Larena Anchuela (to Everyone): 19:41: lo mismo que antes?

Miguel Ángel Vázquez Vera (to Everyone): 19:42: sí

Manuel Larena Anchuela (to Everyone): 19:42: si

Ana Gomez (to Everyone): 19:42: si

[Lemoncode] Dani (to Everyone): 19:42: /^imagen.$/

Miguel Ángel Vázquez Vera (to Everyone): 19:43: tpl

Ana Gomez (to Everyone): 19:43: tpl

Manuel Larena Anchuela (to Everyone): 19:43: tpl

Ferran Puig Martínez (to Everyone): 19:43: tpl

Gema Pujalte Iglesias (to Everyone): 19:43: perdon...que es eso de tpl?

Antonio Contreras (to Everyone): 19:44: tpl = to pa lante

Marcos García F. (to Everyone): 19:44: no pillo bien lo del falso valido

Miguel Ángel Vázquez Vera (to Everyone): 19:44: una reg exp

Gema Pujalte Iglesias (to Everyone): 19:44: habra que hacer uno de to patras

Miguel Ángel Vázquez Vera (to Everyone): 19:44: hahaha sí

Ana Gomez (to Everyone): 19:44: jajajja

stefano durante (to Everyone): 19:44: :-D

Marcos García F. (to Everyone): 19:44: holaimagen ¿por qué es válido?

Marcos García F. (to Everyone): 19:45: eso con un test es true ?

Antonio Contreras (to Everyone): 19:45: si

Antonio Contreras (to Everyone): 19:46: por que tu regular expresion encuentra una coincidencia en tu texto y te da verdadero

Marcos García F. (to Everyone): 19:46: ah vale!!

Marcos García F. (to Everyone): 19:47: ^

Marcos García F. (to Everyone): 19:47: OK!

stefano durante (to Everyone): 19:47: tpl

Amaia (to Everyone): 19:47: no me ha quedado muy claro lo del $

Amaia (to Everyone): 19:47: si

Amaia (to Everyone): 19:48: poner el $

Antonio Contreras (to Everyone): 19:49: muy bien amaia

Amaia (to Everyone): 19:49: ok, ahora si

Ester (to Everyone): 19:50: dani

Ester (to Everyone): 19:51: y si quiero que la cadena que chequee contenga uno de los caracteres comodines

Ester (to Everyone): 19:51: ?

Ester (to Everyone): 19:51: por ejemplo el caracter \

Ester (to Everyone): 19:51: perdon el caracter /

Antonio Contreras (to Everyone): 19:51: es el siguiente pasito

Ester (to Everyone): 19:51: si, lo del punto lo he visto en los apuntes

Ester (to Everyone): 19:52: pero mi duda sería con el caracter /

[Lemoncode] Dani (to Everyone): 19:52: /^imagen.\.jpg$/

Miguel Ángel Vázquez Vera (to Everyone): 19:52: esa barra anula el carácter anterior o posterior?

Miguel Ángel Vázquez Vera (to Everyone): 19:53: ok

Manuel Larena Anchuela (to Everyone): 19:53: ok

Ester (to Everyone): 19:54: si, pero vamos, no es que lo anule

Ester (to Everyone): 19:54: sino que no lo considera un caracter especial, no?

Antonio Contreras (to Everyone): 19:54: eso es

Miguel Ángel Vázquez Vera (to Everyone): 19:54: Sí, perdón, que me he expresado yo mal. No anula

Ester (to Everyone): 19:54: ok

Ester (to Everyone): 19:54: perfecto

Miguel Ángel Vázquez Vera (to Everyone): 19:54: eso sirve para strings también no?

Ester (to Everyone): 19:54: y si quiero dos //

Ester (to Everyone): 19:54: ?

Miguel Ángel Vázquez Vera (to Everyone): 19:54: si quieres poner comillas dentro de un string, por ejemplo

Ester (to Everyone): 19:54: una por una, no?

Ester (to Everyone): 19:55: ok

Miguel Ángel Vázquez Vera (to Everyone): 19:55: vale

stefano durante (to Everyone): 19:55: tpl

Miguel Ángel Vázquez Vera (to Everyone): 19:55: tpl

[Lemoncode] Dani (to Everyone): 19:57: const pattern = /^Jose/;

Miguel Ángel Vázquez Vera (to Everyone): 19:57: Oye, Dani

Miguel Ángel Vázquez Vera (to Everyone): 19:57: Aunque no sea un string, es case sensitive?

stefano durante (to Everyone): 19:57: true false false

Miguel Ángel Vázquez Vera (to Everyone): 19:57: ok

Miguel Ángel Vázquez Vera (to Everyone): 19:58: El ForEach() creo que no lo hemos dado, por cierto

[Lemoncode] Dani (to Everyone): 19:58: const pattern = /^Jose/;

```
const values = ['Jose Perez Gomez', 'Manuel Lozano', 'Maria Jose perez'];

values.forEach(value => {
  console.log(`Regex matchs with "${value}"? ->` pattern.test(value));
})
```

[Lemoncode] Dani (to Everyone): 19:59: const pattern = /^Jose/;

```
const values = ['Jose Perez Gomez', 'Manuel Lozano', 'Maria Jose perez'];

values.forEach(value => {
  console.log('Regex matchs with ' +value +'? ->', pattern.test(value));
})
```

stefano durante (to Everyone): 20:00: tpl

Ana Gomez (to Everyone): 20:00: tpl

Manuel Larena Anchuela (to Everyone): 20:00: tpl

Antonio Contreras (to Everyone): 20:00: dudas ?

Miguel Ángel Vázquez Vera (to Everyone): 20:01: que acabe así

Manuel Larena Anchuela (to Everyone): 20:01: el primero

Amaia (to Everyone): 20:01: el 1

Gema Pujalte Iglesias (to Everyone): 20:01: true,false,false?

Antonio Contreras (to Everyone): 20:02: muy bien gema

[Lemoncode] Dani (to Everyone): 20:02: const pattern = /txt$/;

```
const values = ['mitexto.txt', 'imagen.jpg', 'imagentxt.jpg'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

[Lemoncode] Dani (to Everyone): 20:02: []

[Lemoncode] Dani (to Everyone): 20:03: const pattern = /[1234567890]/;

```
const values = ['1', 'A', 'a'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 20:03: que incluye

stefano durante (to Everyone): 20:03: 1 0 0

Oscar Moreno (to Everyone): 20:03: que tiene que estar dentro del rango

Ana Gomez (to Everyone): 20:03: que admite esos valores?

Antonio Contreras (to Everyone): 20:03: excato

Antonio Contreras (to Everyone): 20:04: exacto*

Miguel Ángel Vázquez Vera (to Everyone): 20:04: uno o varios no?

Miguel Ángel Vázquez Vera (to Everyone): 20:04: Incluso si no están en orden?

Miguel Ángel Vázquez Vera (to Everyone): 20:04: vale

Miguel Ángel Vázquez Vera (to Everyone): 20:04: Ya está

Marcos García F. (to Everyone): 20:05: bien

Oscar Moreno (to Everyone): 20:05: entraria 12a cm true?

Antonio Contreras (to Everyone): 20:05: si ese caso si daría true oscar

Marcos García F. (to Everyone): 20:06: falso valido no?
Antonio Contreras (to Everyone): 20:06: si sería un falso positivo
Marcos García F. (to Everyone): 20:06: ok
Miguel Ángel Vázquez Vera (to Everyone): 20:08: ah
Miguel Ángel Vázquez Vera (to Everyone): 20:08: guay
Ana Gomez (to Everyone): 20:08: tpl
Amaia (to Everyone): 20:08: pero solo uno no?
stefano durante (to Everyone): 20:08: tpl
Amaia (to Everyone): 20:08: ok ok
Miguel Ángel Vázquez Vera (to Everyone): 20:09: Lo que hace es buscar que el primer caracter sea válido?
Miguel Ángel Vázquez Vera (to Everyone): 20:09: Y ya deja de buscar
Antonio Contreras (to Everyone): 20:09: eso es
Miguel Ángel Vázquez Vera (to Everyone): 20:09: ok
Miguel Ángel Vázquez Vera (to Everyone): 20:10: tpl
Manuel Larena Anchuela (to Everyone): 20:11: tpl
Oscar Moreno (to Everyone): 20:11: puedes repetir esto ultimo
Ana Gomez (to Everyone): 20:11: tpl
Oscar Moreno (to Everyone): 20:11: ?
Marcos García F. (to Everyone): 20:12: como el operador ! algo así...
Oscar Moreno (to Everyone): 20:12: perrfecto
Antonio Contreras (to Everyone): 20:13: guay
Gema Pujalte Iglesias (to Everyone): 20:13: y una pregunta,que me perdi...para que servian las barras del principio y el final?
Gema Pujalte Iglesias (to Everyone): 20:13: ah genial
Gema Pujalte Iglesias (to Everyone): 20:13: ok
Miguel Ángel Vázquez Vera (to Everyone): 20:14: true, false, true, false
Miguel Ángel Vázquez Vera (to Everyone): 20:14: ?
Antonio Contreras (to Everyone): 20:15: muy bien migue
Miguel Ángel Vázquez Vera (to Everyone): 20:15: gracias
Oscar Moreno (to Everyone): 20:15: puedes pegarlo en firepad este ejemplo?
Miguel Ángel Vázquez Vera (to Everyone): 20:15: y otra forma de limitar la longitud no?
[Lemoncode] Dani (to Everyone): 20:15: const pattern = /^...$/;

```
const values = ['abc', 'ab', 'a12', 'ab\n'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 20:16: sin usar .length
Miguel Ángel Vázquez Vera (to Everyone): 20:16: para un condicional por ejemplo
Miguel Ángel Vázquez Vera (to Everyone): 20:16: que guay
Miguel Ángel Vázquez Vera (to Everyone): 20:17: lo mismo?
Miguel Ángel Vázquez Vera (to Everyone): 20:17: pero sin los otros caracteres especial
Miguel Ángel Vázquez Vera (to Everyone): 20:18: vale
[Lemoncode] Dani (to Everyone): 20:18: \w -> [a-zA-Z0-9_]
Miguel Ángel Vázquez Vera (to Everyone): 20:18: ahora entiendo la validación del email
Marcos García F. (to Everyone): 20:18: y tres veces?
Miguel Ángel Vázquez Vera (to Everyone): 20:19: Sí, me refiero a que ahora entiendo lo que nos

has pasado al principio para validar el email

Marcos García F. (to Everyone): 20:19: jaja pero porqué 3 veces?

Marcos García F. (to Everyone): 20:19: aaah vale

Miguel Ángel Vázquez Vera (to Everyone): 20:20: true, true, true, false

Antonio Contreras (to Everyone): 20:20: muy bien

Oscar Moreno (to Everyone): 20:20: lo que dice miguel angel

Ferran Puig Martínez (to Everyone): 20:20: true, true, true, false

Antonio Contreras (to Everyone): 20:20: muy bien

[Lemoncode] Dani (to Everyone): 20:20:

```
const pattern = /^\w\w\w$/;

const values = ['aaa', 'abc', 'a12', 'ab'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 20:21: tpl

Ana Gomez (to Everyone): 20:21: tpl

stefano durante (to Everyone): 20:21: tpl

Oscar Moreno (to Everyone): 20:21: ^

Miguel Ángel Vázquez Vera (to Everyone): 20:21: [^]

Antonio Contreras (to Everyone): 20:21: eso es

Miguel Ángel Vázquez Vera (to Everyone): 20:22: tpl

Oscar Moreno (to Everyone): 20:22: el cuarto

Miguel Ángel Vázquez Vera (to Everyone): 20:22: el penúltimo

Antonio Contreras (to Everyone): 20:23: eso es

Miguel Ángel Vázquez Vera (to Everyone): 20:23: Una cosa

Oscar Moreno (to Everyone): 20:23: copialo en firepad

Oscar Moreno (to Everyone): 20:23: please

[Lemoncode] Dani (to Everyone): 20:23: // \w -> [^a-zA-Z0-9_]

```
const pattern = /^\W\W\W$/;

const values = ['aaa', '123', 'a12', '@#%', '@#'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 20:23: Hay alguna forma de no tener que repetir \w por cada caracter?

Miguel Ángel Vázquez Vera (to Everyone): 20:23: imagíante que quiero validar 20 caracteres

Antonio Contreras (to Everyone): 20:23: si

Antonio Contreras (to Everyone): 20:23: jajaja

Miguel Ángel Vázquez Vera (to Everyone): 20:24: me imaginaba que algo habría... javascript nunca defrauda

Miguel Ángel Vázquez Vera (to Everyone): 20:24: Xd

Antonio Contreras (to Everyone): 20:24: jajaj

Antonio Contreras (to Everyone): 20:24: lo hay lo hay

Ferran Puig Martínez (to Everyone): 20:24: tpl

Marcos García F. (to Everyone): 20:24: si
stefano durante (to Everyone): 20:24: tpl
Miguel Ángel Vázquez Vera (to Everyone): 20:24: tpl
Miguel Ángel Vázquez Vera (to Everyone): 20:25: sí
Ferran Puig Martínez (to Everyone): 20:25: si
Ana Gomez (to Everyone): 20:25: tpl
Ester (to Everyone): 20:25: pero tambien solo uno, no?
Manuel Larena Anchuela (to Everyone): 20:25: si
Miguel Ángel Vázquez Vera (to Everyone): 20:26: segundo
Miguel Ángel Vázquez Vera (to Everyone): 20:26: y tecero
Oscar Moreno (to Everyone): 20:26: segundo y tercero
Manuel Larena Anchuela (to Everyone): 20:26: segundo y tercero
Antonio Contreras (to Everyone): 20:26: eso es
Manuel Larena Anchuela (to Everyone): 20:26: primero y cuarto
Ana Gomez (to Everyone): 20:26: primero y cuarto
Ferran Puig Martínez (to Everyone): 20:26: 2 y 31, 4
Miguel Ángel Vázquez Vera (to Everyone): 20:26: sí
Ferran Puig Martínez (to Everyone): 20:26: 1, 4
Miguel Ángel Vázquez Vera (to Everyone): 20:27: una cosa, podrían combinarse no?
Antonio Contreras (to Everyone): 20:27: muy bien!!!
Antonio Contreras (to Everyone): 20:27: claro MIguel
Miguel Ángel Vázquez Vera (to Everyone): 20:27: \d\D
Miguel Ángel Vázquez Vera (to Everyone): 20:28: jo, esto mola mucho
[Lemoncode] Dani (to Everyone): 20:28: // \d -> [0-9]

```
// \D -> [^0-9]
const pattern = /^\d\D\D$/;

const values = ['aaa', '123', '012', '@#%', '12', '6@a'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Antonio Contreras (to Everyone): 20:28: acuerdate del ejemplo del email tenía combinación de muchos
Miguel Ángel Vázquez Vera (to Everyone): 20:28: sí, Antonio ^^
Miguel Ángel Vázquez Vera (to Everyone): 20:29: pero al igual que los otros, sólo un espacio o tabulador?
Miguel Ángel Vázquez Vera (to Everyone): 20:30: tpl
Ferran Puig Martínez (to Everyone): 20:30: si
Miguel Ángel Vázquez Vera (to Everyone): 20:31: \d
stefano durante (to Everyone): 20:31: \d*8
Ferran Puig Martínez (to Everyone): 20:31: /d/d/d/d/d/d...
stefano durante (to Everyone): 20:32: \s
Miguel Ángel Vázquez Vera (to Everyone): 20:32: \s
Antonio Contreras (to Everyone): 20:32: genial
Miguel Ángel Vázquez Vera (to Everyone): 20:32: \w
stefano durante (to Everyone): 20:32: \w
Miguel Ángel Vázquez Vera (to Everyone): 20:32: mmm

Oscar Moreno (to Everyone): 20:32: \w
Ana Gomez (to Everyone): 20:32: D
Ester (to Everyone): 20:32: no
Miguel Ángel Vázquez Vera (to Everyone): 20:32: pero también sería número
Ana Gomez (to Everyone): 20:32: no no..
Ester (to Everyone): 20:32: porque el \w puede ser un numero
Ferran Puig Martínez (to Everyone): 20:32: [a-Z]
stefano durante (to Everyone): 20:32: \D
Oscar Moreno (to Everyone): 20:32: o \D
Ferran Puig Martínez (to Everyone): 20:33: [A-Z]
Ester (to Everyone): 20:33: no \D tambien puede ser @
stefano durante (to Everyone): 20:33: ok
Antonio Contreras (to Everyone): 20:33: exacto Ferran
Ferran Puig Martínez (to Everyone): 20:34: el orden tiene preferencia?
Miguel Ángel Vázquez Vera (to Everyone): 20:34: Sí
Ferran Puig Martínez (to Everyone): 20:34: A-Za-z a-zA-Z?
Ferran Puig Martínez (to Everyone): 20:34: ok
David Ramirez (to Everyone): 20:35: para escribir los [ ] no necesitas escapar con \??

David Ramirez (to Everyone): 20:35: ok
stefano durante (to Everyone): 20:36: tpl
Miguel Ángel Vázquez Vera (to Everyone): 20:36: Una pregunta, y si quisieramos también ofrecer la opción de escribir el guión o el espacio?
Ana Gomez (to Everyone): 20:36: tpl
Miguel Ángel Vázquez Vera (to Everyone): 20:36: O ya me estoy liando?
Ester (to Everyone): 20:36: tpl
David Ramirez (to Everyone): 20:36: escapas para usar el corchete como string
[Lemoncode] Dani (to Everyone): 20:36: // \s -> [ ] (espacio, tabulador, nueva linea)

```
// \S -> [^   ] NO (espacio, tabulador, nueva linea)
const pattern = /^\d\d\d\d\d\d\d\d\s[a-zA-Z]$/;

const values = ['12345678 Q', '12345678 q','12345678-Q', 'abc45678-Q', 'abc45678 Q'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Antonio Contreras (to Everyone): 20:36: no no Miguel se puede hacer
Miguel Ángel Vázquez Vera (to Everyone): 20:36: vale
Miguel Ángel Vázquez Vera (to Everyone): 20:37: Otra pregunta tonta
Miguel Ángel Vázquez Vera (to Everyone): 20:37: los rangos pueden empezarse y acabarse donde quieras?
Miguel Ángel Vázquez Vera (to Everyone): 20:37: por ejemplo
Miguel Ángel Vázquez Vera (to Everyone): 20:38: c-k
Miguel Ángel Vázquez Vera (to Everyone): 20:38: no siempre a-z
Miguel Ángel Vázquez Vera (to Everyone): 20:38: L
Miguel Ángel Vázquez Vera (to Everyone): 20:38: como sea
Miguel Ángel Vázquez Vera (to Everyone): 20:38: aham
Miguel Ángel Vázquez Vera (to Everyone): 20:38: que guay

Miguel Ángel Vázquez Vera (to Everyone): 20:39: sí

Miguel Ángel Vázquez Vera (to Everyone): 20:39: me flipa

Antonio Contreras (to Everyone): 20:39: ajjajaj

Miguel Ángel Vázquez Vera (to Everyone): 20:39: Xd

Antonio Contreras (to Everyone): 20:39: cuando las entiendes molan mucho

Miguel Ángel Vázquez Vera (to Everyone): 20:39: sí

Miguel Ángel Vázquez Vera (to Everyone): 20:40: venga

stefano durante (to Everyone): 20:40: ok

Ferran Puig Martínez (to Everyone): 20:40: hasta luego!

Antonio Contreras (to Everyone): 20:45: si

Ferran Puig Martínez (to Everyone): 20:46: sip

David Ramirez (to Everyone): 20:46: si

stefano durante (to Everyone): 20:46: si

[Lemoncode] Dani (to Everyone): 20:47: const pattern = /^\d{8}\s[a-zA-Z]$/;

```
const values = ['12345678 Q', '12345678 q','12345678-Q', 'abc45678-Q', 'abc45678 Q'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 20:47: tpl

Ferran Puig Martínez (to Everyone): 20:47: muy práctico si :)

Antonio Contreras (to Everyone): 20:47: jajaja

Antonio Contreras (to Everyone): 20:47: ahora la cosa se queda más clara

Ferran Puig Martínez (to Everyone): 20:48: conn grupos no?

Antonio Contreras (to Everyone): 20:48: muy bien

Miguel Ángel Vázquez Vera (to Everyone): 20:48: sí

Ferran Puig Martínez (to Everyone): 20:48: si

Miguel Ángel Vázquez Vera (to Everyone): 20:49: true, false, false, true

stefano durante (to Everyone): 20:49: t f f t

Ferran Puig Martínez (to Everyone): 20:49: 1, 3, 4

Manuel Larena Anchuela (to Everyone): 20:49: primero y último

Ferran Puig Martínez (to Everyone): 20:49: el 3 no?

Antonio Contreras (to Everyone): 20:49: muy bien

Ferran Puig Martínez (to Everyone): 20:49: ahh ok ok números

Antonio Contreras (to Everyone): 20:50: jajaj muy bien Ferran

Amaia (to Everyone): 20:51: me he perdido un poco esto ultimo de despues del descanso

Amaia (to Everyone): 20:51: puedes repetir?

Amaia (to Everyone): 20:51: si

Amaia (to Everyone): 20:52: si

Amaia (to Everyone): 20:53: vale

Ester (to Everyone): 20:54: primero y segundo

Miguel Ángel Vázquez Vera (to Everyone): 20:54: 1, 2

Amaia (to Everyone): 20:54: el y el 2

stefano durante (to Everyone): 20:54: t t f f

Manuel Larena Anchuela (to Everyone): 20:54: el primero y segundo

Amaia (to Everyone): 20:54: el 1 y el 2 perdon

Miguel Ángel Vázquez Vera (to Everyone): 20:54: tpl

stefano durante (to Everyone): 20:54: tpl

Manuel Larena Anchuela (to Everyone): 20:54: tpl

Miguel Ángel Vázquez Vera (to Everyone): 20:56: que puede ponerse o no?

Ferran Puig Martínez (to Everyone): 20:56: puede o no haber?

Antonio Contreras (to Everyone): 20:57: muy bien

Amaia (to Everyone): 20:57: pero puede haber o no espacio?

Amaia (to Everyone): 20:57: o cualquier cosa?

Miguel Ángel Vázquez Vera (to Everyone): 20:57: 1 y 3

Ferran Puig Martínez (to Everyone): 20:57: 1,3

Amaia (to Everyone): 20:58: aaaaaa ok

Miguel Ángel Vázquez Vera (to Everyone): 20:58: pregunta sobre el caracter precedente

Miguel Ángel Vázquez Vera (to Everyone): 20:58: eso

Ferran Puig Martínez (to Everyone): 20:58: lo que ce Amaia sería .?

Ferran Puig Martínez (to Everyone): 20:58: seria .? ?

Miguel Ángel Vázquez Vera (to Everyone): 20:58: ?

Ferran Puig Martínez (to Everyone): 20:58: -> ?

Ana Gomez (to Everyone): 20:58: ?

Amaia (to Everyone): 20:59: guay

Miguel Ángel Vázquez Vera (to Everyone): 20:59: a tope

Antonio Contreras (to Everyone): 20:59: va muy bien

Antonio Contreras (to Everyone): 20:59: jajaja

Marcos García F. (to Everyone): 20:59: todo

Oscar Moreno (to Everyone): 20:59: de 0 a infinito?

Antonio Contreras (to Everyone): 21:00: muy bien

Antonio Contreras (to Everyone): 21:00: jajajaja

Miguel Ángel Vázquez Vera (to Everyone): 21:00: sí. perfect

Miguel Ángel Vázquez Vera (to Everyone): 21:01: 1, 2, 3

David Ramirez (to Everyone): 21:01: 2 y 3

Ester (to Everyone): 21:01: 1,2,3

Miguel Ángel Vázquez Vera (to Everyone): 21:01: el 0 lo incluye no?

stefano durante (to Everyone): 21:01: 1,2,3

[Lemoncode] Dani (to Everyone): 21:02: // {2} {2,} {2,4}

```
// ? {0,1}
// * {0,}

const pattern = /^imagen\d*\.jpg$/;

const values = ['imagen.jpg', 'imagen1.jpg', 'imagen01.jpg', 'imagen_2.jpg', 'imagenA2.jpg'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

David Ramirez (to Everyone): 21:02: si si

Miguel Ángel Vázquez Vera (to Everyone): 21:02: tpl

Ana Gomez (to Everyone): 21:02: tpl

Ester (to Everyone): 21:02: tpl

Miguel Ángel Vázquez Vera (to Everyone): 21:03: 2, 3

[Lemoncode] Dani (to Everyone): 21:03: // {2} {2,} {2,4}

```
// ? {0,1}
// * {0,}
// + {1,}

const pattern = /^imagen\d+\.jpg$/;

const values = ['imagen.jpg', 'imagen1.jpg', 'imagen01.jpg', 'imagen_2.jpg', 'imagenA2.jpg'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 21:04: ahora sí que sí podemos entender la validación del email
Miguel Ángel Vázquez Vera (to Everyone): 21:04: ^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$
Miguel Ángel Vázquez Vera (to Everyone): 21:04: Xd
Miguel Ángel Vázquez Vera (to Everyone): 21:04: quién lo diría
Antonio Contreras (to Everyone): 21:04: poco a poco si
Miguel Ángel Vázquez Vera (to Everyone): 21:04: cuando ha empezado la clase
Miguel Ángel Vázquez Vera (to Everyone): 21:04: que guay
Miguel Ángel Vázquez Vera (to Everyone): 21:05: sí
Ana Gomez (to Everyone): 21:06: si
stefano durante (to Everyone): 21:06: si
Miguel Ángel Vázquez Vera (to Everyone): 21:06: tpl
Ester (to Everyone): 21:06: si
Miguel Ángel Vázquez Vera (to Everyone): 21:07: historia sería necesario ponerlo entre paréntesis?
Miguel Ángel Vázquez Vera (to Everyone): 21:08: en este caso, digo
Miguel Ángel Vázquez Vera (to Everyone): 21:08: vale
Ferran Puig Martínez (to Everyone): 21:09: con
Antonio Contreras (to Everyone): 21:09: con paréntesis o sin ?
Ana Gomez (to Everyone): 21:09: no entiendo entonces por qué se pònen los parentesis en pre
Ferran Puig Martínez (to Everyone): 21:09: para mi es más claro
Miguel Ángel Vázquez Vera (to Everyone): 21:09: El hecho de poner paréntesis es para poder usarlo "?" no?
Ester (to Everyone): 21:10: que solo seria condicional la e
Oscar Moreno (to Everyone): 21:10: que prhistoria valdria?
Oscar Moreno (to Everyone): 21:10: sin la e?
Miguel Ángel Vázquez Vera (to Everyone): 21:10: pre valdría
Miguel Ángel Vázquez Vera (to Everyone): 21:10: no
Ana Gomez (to Everyone): 21:10: aaah vale vale
Miguel Ángel Vázquez Vera (to Everyone): 21:10: claro
Miguel Ángel Vázquez Vera (to Everyone): 21:11: sí
Ana Gomez (to Everyone): 21:11: perfecto
Antonio Contreras (to Everyone): 21:11: guay
Miguel Ángel Vázquez Vera (to Everyone): 21:11: tpl
Ester (to Everyone): 21:11: tpl
Ana Gomez (to Everyone): 21:11: tpl
Manuel Larena Anchuela (to Everyone): 21:11: tpl

stefano durante (to Everyone): 21:12: si

Miguel Ángel Vázquez Vera (to Everyone): 21:12: sí

Ana Gomez (to Everyone): 21:13: [A-Z]

Ferran Puig Martínez (to Everyone): 21:13: [A-Z]

Miguel Ángel Vázquez Vera (to Everyone): 21:13: eso

stefano durante (to Everyone): 21:14: {2}

Miguel Ángel Vázquez Vera (to Everyone): 21:14: {2}

David Ramirez (to Everyone): 21:14: [A-Z]{2}

Antonio Contreras (to Everyone): 21:14: muy bien

Ferran Puig Martínez (to Everyone): 21:14: [A->]{0,2}

Miguel Ángel Vázquez Vera (to Everyone): 21:14: {1,2}

Antonio Contreras (to Everyone): 21:14: eso es

Ana Gomez (to Everyone): 21:14: \d{4}

Antonio Contreras (to Everyone): 21:14: muy bien ana

stefano durante (to Everyone): 21:15: [A-Z]{1,2}

Miguel Ángel Vázquez Vera (to Everyone): 21:15: [A-Z]{1,2}

Miguel Ángel Vázquez Vera (to Everyone): 21:15: /\d{4}[A-Z]{3}

Miguel Ángel Vázquez Vera (to Everyone): 21:16: SÍ

Ester (to Everyone): 21:16: di

stefano durante (to Everyone): 21:16: si

Ferran Puig Martínez (to Everyone): 21:16: guai

Ester (to Everyone): 21:16: si

[Lemoncode] Dani (to Everyone): 21:17: // MA4050AZ

```
// 0642AFD -> \d{4}[A-Z]{3}

// (A)|(B)

const pattern = /^([A-Z]{1,2}\d{4}[A-Z]{1,2})|(\d{4}[A-Z]{3})$/;

const values = ['MA4050AZ', '0642AFD'];

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Ferran Puig Martínez (to Everyone): 21:17: realmente son iguales exceptuando que puede o no ir 2 letras delante, no?

Ferran Puig Martínez (to Everyone): 21:17: {0,2}

Ferran Puig Martínez (to Everyone): 21:17: ah ok

Ferran Puig Martínez (to Everyone): 21:17: y en el otro 3

[Lemoncode] Dani (to Everyone): 21:18: // MA4050AZ

```
// 0642AFD -> \d{4}[A-Z]{3}

// (A)|(B)

const pattern = /^([A-Z]{1,2}\d{4}[A-Z]{1,2})|(\d{4}[A-Z]{3})$/;

const values = ['MA4050AZ', '0642AFD', 'MA4444'];
```

```
values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Ferran Puig Martínez (to Everyone): 21:19: A que puede coger solo una letra y nos ería válido

Miguel Ángel Vázquez Vera (to Everyone): 21:19: Pero sería igual

Miguel Ángel Vázquez Vera (to Everyone): 21:20: que con el ejemplo de Madrid

Ferran Puig Martínez (to Everyone): 21:20: no, porque podría validar A3456AS

Ferran Puig Martínez (to Everyone): 21:20: y tiene que ser o 0 o 2

Ferran Puig Martínez (to Everyone): 21:20: letras

Miguel Ángel Vázquez Vera (to Everyone): 21:21: 1 o 3?

Ferran Puig Martínez (to Everyone): 21:22: no valdría

Miguel Ángel Vázquez Vera (to Everyone): 21:22: ok

Miguel Ángel Vázquez Vera (to Everyone): 21:22: sí

Ferran Puig Martínez (to Everyone): 21:22: si

Miguel Ángel Vázquez Vera (to Everyone): 21:22: En ese caso hay que poner dos pattern no?

David Ramirez (to Everyone): 21:22: entonces no hay ninguna expresión que haga eso no?

Antonio Contreras (to Everyone): 21:22: alguno se pierde ?

stefano durante (to Everyone): 21:22: yo me he perdido un poco

Gema Pujalte Iglesias (to Everyone): 21:22: si lo puedes repetir..

Ferran Puig Martínez (to Everyone): 21:23: guau aquí te descuidas y se complica mucho jeje

Ferran Puig Martínez (to Everyone): 21:23: si

David Ramirez (to Everyone): 21:23: si

stefano durante (to Everyone): 21:23: creo que si...pero l que proponía Ferran estaba bien no?

Gema Pujalte Iglesias (to Everyone): 21:24: creo que si

stefano durante (to Everyone): 21:25: si

Manuel Larena Anchuela (to Everyone): 21:25: si

Miguel Ángel Vázquez Vera (to Everyone): 21:26: en todos los grupos?

Miguel Ángel Vázquez Vera (to Everyone): 21:26: de 1 a 3

Miguel Ángel Vázquez Vera (to Everyone): 21:26: vale

Ester (to Everyone): 21:26: y ademas tiene que ir hasta 255

Ester (to Everyone): 21:26: no?

Antonio Contreras (to Everyone): 21:26: exacto ester

Ester (to Everyone): 21:26: claro, es que eso es lo que yo estaba pensando

Ester (to Everyone): 21:26: que como comprobabas eso

Miguel Ángel Vázquez Vera (to Everyone): 21:27: sí

Ester (to Everyone): 21:28: dani, y se puede comprobar que sea un numero negativo?

Ester (to Everyone): 21:28: aha

Ester (to Everyone): 21:28: si

Miguel Ángel Vázquez Vera (to Everyone): 21:28: \d{1,3}

Miguel Ángel Vázquez Vera (to Everyone): 21:29: ^

Antonio Contreras (to Everyone): 21:29: muy bien visto

Miguel Ángel Vázquez Vera (to Everyone): 21:29: $

Ana Gomez (to Everyone): 21:29: y $

Miguel Ángel Vázquez Vera (to Everyone): 21:30: \.

David Ramirez (to Everyone): 21:30: const pattern = /^\d{1,3}\.\d{1,3}\.\d{1,3}.\d{1,3}$/

Antonio Contreras (to Everyone): 21:30: guay

[Lemoncode] Dani (to Everyone): 21:30: // IP 127.3.12.12

```
const pattern = /^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$/;

const values = ['127.0.0.1', 'A.0.0.1', '-1.52.45.233', '10.98.199.1', 'AA 123.22.22.0 NN']; // SI  NO NO SI NO

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Gema Pujalte Iglesias (to Everyone): 21:31: pero porque siempre es {1,3}?
Gema Pujalte Iglesias (to Everyone): 21:31: creo que eso no lo he pillado
Gema Pujalte Iglesias (to Everyone): 21:31: si
David Ramirez (to Everyone): 21:32: Una duda, se podria hacer algo así?
Ferran Puig Martínez (to Everyone): 21:32: (\d{1,3}\.){3}?
David Ramirez (to Everyone): 21:32: const pattern = /^(\d{1,3}\.){4}$/
Antonio Contreras (to Everyone): 21:32: genial ferran
Manuel Larena Anchuela (to Everyone): 21:32: ^(\d{1,3}\.){3}d{1,3}$
Miguel Ángel Vázquez Vera (to Everyone): 21:32: eso
David Ramirez (to Everyone): 21:32: Miguel ahi las dao
[Lemoncode] Dani (to Everyone): 21:33: // IP 123.3.12.12

```
const pattern = /^(\d{1,3}\.){3}\d{1,3}$/;

const values = ['127.0.0.1', 'A.0.0.1', '-1.52.45.233', '10.98.199.1', 'AA 123.22.22.0 NN']; // SI  NO NO SI NO

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Miguel Ángel Vázquez Vera (to Everyone): 21:35: Xd
Ester (to Everyone): 21:35: hola
Ester (to Everyone): 21:35: me veis?
Miguel Ángel Vázquez Vera (to Everyone): 21:35: sí
Ester (to Everyone): 21:35: pues yo no veo nada en la pantalla
Ester (to Everyone): 21:35: la veo como congelada
Miguel Ángel Vázquez Vera (to Everyone): 21:35: se ve bien
stefano durante (to Everyone): 21:35: se ve bien
Ester (to Everyone): 21:35: pues entonces soy yo
Ester (to Everyone): 21:36: vale, salgo y entro de nuevo
Antonio Contreras (to Everyone): 21:36: reinicia si
Ester (to Everyone): 21:36: ok
Ferran Puig Martínez (to Everyone): 21:36: (/d/.?){8} el inicio?
stefano durante (to Everyone): 21:36: \d{1,8}
Antonio Contreras (to Everyone): 21:37: muy bien
Ester (to Everyone): 21:37: yatoy
Ferran Puig Martínez (to Everyone): 21:37: ok
Antonio Contreras (to Everyone): 21:37: genial
Ester (to Everyone): 21:37: si
Ester (to Everyone): 21:37: gracias

David Ramirez (to Everyone): 21:38: [A-Za-z]

Oscar Moreno (to Everyone): 21:38: [A-Za-z]

Ana Gomez (to Everyone): 21:38: [A-Za-z]{1}

Antonio Contreras (to Everyone): 21:38: muy bien

Ana Gomez (to Everyone): 21:38: aah ok

Manuel Larena Anchuela (to Everyone): 21:39: \.?

Miguel Ángel Vázquez Vera (to Everyone): 21:39: pero lo estás haciendo al final del grupo de números no?

Miguel Ángel Vázquez Vera (to Everyone): 21:40: (-\s_)?

stefano durante (to Everyone): 21:40: w

David Ramirez (to Everyone): 21:41: \d{8}.?[A-Za-z]

Ferran Puig Martínez (to Everyone): 21:41: (\s | \[-] | ?)

Ana Gomez (to Everyone): 21:41: ((\s)|(_)|(-))?

stefano durante (to Everyone): 21:41: ok cierto

Ana Gomez (to Everyone): 21:42: ok

Ana Gomez (to Everyone): 21:42: pero el que agrupa a todos no se quita no?

Ana Gomez (to Everyone): 21:42: porque tiene la ? despues

Ana Gomez (to Everyone): 21:42: bien bien! :)

Miguel Ángel Vázquez Vera (to Everyone): 21:43: yo creo que el último es mejor no validarlo

Miguel Ángel Vázquez Vera (to Everyone): 21:43: Xd

stefano durante (to Everyone): 21:43: jajajjaja

Miguel Ángel Vázquez Vera (to Everyone): 21:44: \d{2}\.?

Miguel Ángel Vázquez Vera (to Everyone): 21:44: y así

Antonio Contreras (to Everyone): 21:44: genial

Manuel Larena Anchuela (to Everyone): 21:44: \d{2}\.?\d{3}\.?\d{3}

[Lemoncode] Dani (to Everyone): 21:44: // DNI

```
// \w [a-zA-Z0-9_]

const pattern = /^\d{8}(-|\s|_)?[A-Za-z]$/;

const values = ['12345678Q', '12345678-Q', '12345678 Q', '12345678_Q', '12345678 q', '12.345.678
Q']; // SI

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

[Lemoncode] Dani (to Everyone): 21:45: // DNI

```
// \w [a-zA-Z0-9_]

const pattern = /^\d{2}\.?\d{3}\.?\d{3}(-|\s|_)?[A-Za-z]$/;

const values = ['12345678Q', '12345678-Q', '12345678 Q', '12345678_Q', '12345678 q', '12.345.678
Q']; // SI

values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.test(value));
})
```

Ana Gomez (to Everyone): 21:45: tpl

Manuel Larena Anchuela (to Everyone): 21:45: si
Miguel Ángel Vázquez Vera (to Everyone): 21:45: tpl
Manuel Larena Anchuela (to Everyone): 21:45: tpl
Ferran Puig Martínez (to Everyone): 21:45: (\d{1,3}\.?){3} -> simplificado?
Ester (to Everyone): 21:45: tpl
stefano durante (to Everyone): 21:45: tpl
Oscar Moreno (to Everyone): 21:46: copia ese en el firepad
Ester (to Everyone): 21:46: no, porque entonces valdria tambien 1.2.3
Miguel Ángel Vázquez Vera (to Everyone): 21:47: daría true
Ferran Puig Martínez (to Everyone): 21:47: ooook
Miguel Ángel Vázquez Vera (to Everyone): 21:47: claro
Antonio Contreras (to Everyone): 21:47: muy bien
Miguel Ángel Vázquez Vera (to Everyone): 21:48: Dani, esto se usa mucho en programación web?
Miguel Ángel Vázquez Vera (to Everyone): 21:48: guay
Miguel Ángel Vázquez Vera (to Everyone): 21:49: Me gusta mucho, la verdad
stefano durante (to Everyone): 21:49: para webpack utilizé alguna
David Ramirez (to Everyone): 21:49: Me ha encantado la clase Dani, no sabes la de veces que he visto estos codigajos por ahi y ahora por fin voy a entenderlos!
Miguel Ángel Vázquez Vera (to Everyone): 21:49: hahaha
stefano durante (to Everyone): 21:49: :-)
Ferran Puig Martínez (to Everyone): 21:49: super práctico :)
Miguel Ángel Vázquez Vera (to Everyone): 21:49: Yo nunca los he visto, pero estoy flipando igual
Ester (to Everyone): 21:49: Dani es un crack
Antonio Contreras (to Everyone): 21:50: cuando se entienden suben un montón la moral xD
Antonio Contreras (to Everyone): 21:50: si Ester Dani es muy bueno
Ester (to Everyone): 21:50: lo se
Miguel Ángel Vázquez Vera (to Everyone): 21:51: mmm
Miguel Ángel Vázquez Vera (to Everyone): 21:51: depende de dónde lo introduzca no?
Miguel Ángel Vázquez Vera (to Everyone): 21:51: quiero decir, si se almacena en un objeto, en un input...
[Lemoncode] Dani (to Everyone): 21:51: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions
Ferran Puig Martínez (to Everyone): 21:53: array
Ester (to Everyone): 21:54: porque es lo opcional
Ferran Puig Martínez (to Everyone): 21:54: que estan agrupados?
Ester (to Everyone): 21:54: lo que esta con or
Ester (to Everyone): 21:54: vamos
Ester (to Everyone): 21:54: si, me referia al OR
Ester (to Everyone): 21:54: ah, vale, es por los ()
[Lemoncode] Dani (to Everyone): 21:55: // DNI

```
// \w [a-zA-Z0-9_]

const pattern = /^(\d{2}\.?\d{3}\.?\d{3})(-|\s|_)?([A-Za-z])$/;

const values = ['12345678Q', '12345678-Q', '12345678 Q', '12345678_Q', '12345678 q', '12.345.678 Q']; // SI
```

```
values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', pattern.exec(value));
})
```

Ferran Puig Martínez (to Everyone): 21:55: entonces es una buena praxis ponerlo siempre en grupos? por si las moscas?

Ferran Puig Martínez (to Everyone): 21:56: ok

Antonio Contreras (to Everyone): 21:56: claro segun lo que necesites en tu proyecto

Miguel Ángel Vázquez Vera (to Everyone): 21:56: claro

Miguel Ángel Vázquez Vera (to Everyone): 21:56: A todo esto, Dani, ya que no hemos dado el forEach()

Miguel Ángel Vázquez Vera (to Everyone): 21:57: podríamos hacer lo mismo con cualquier for?

Antonio Contreras (to Everyone): 21:57: si si

Miguel Ángel Vázquez Vera (to Everyone): 21:57: vale ^^

[Lemoncode] Dani (to Everyone): 21:58: // DNI

```
// \w [a-zA-Z0-9_]

const pattern = /^\d{2}\.?\d{3}\.?\d{3}(-|\s|_)?([A-Za-z])$/;

const values = ['12345678Q', '12345678-Q', '12345678 Q', '12345678_Q', '12345678 q', '12.345.678 Q']; // SI

// pattern.exec(value)
// value.match(pattern)
values.forEach(value => {
  console.log('Regex matchs with ' + value +'? ->', value.match(pattern));
})
```

Manuel Larena Anchuela (to Everyone): 21:58: si

Ferran Puig Martínez (to Everyone): 21:58: no, sorry :/

Amaia (to Everyone): 21:58: si

Ester (to Everyone): 21:58: cual es la diferencia?

Ester (to Everyone): 21:58: es que no me he enterado

Ferran Puig Martínez (to Everyone): 21:59: aja

Ferran Puig Martínez (to Everyone): 22:00: ahh ok

Ferran Puig Martínez (to Everyone): 22:00: y porque no usar siempre el match?

Ferran Puig Martínez (to Everyone): 22:00: gasta más memoria o,...?

Ferran Puig Martínez (to Everyone): 22:00: ok

Oscar Moreno (to Everyone): 22:01: puedes explicar lo de las lineas

Oscar Moreno (to Everyone): 22:01: por que no vale para multilinea?

Ferran Puig Martínez (to Everyone): 22:03: ok, por mi bien

Miguel Ángel Vázquez Vera (to Everyone): 22:03: Venga

Miguel Ángel Vázquez Vera (to Everyone): 22:03: a tope

Oscar Moreno (to Everyone): 22:03: perfecto

Manuel Larena Anchuela (to Everyone): 22:03: vale

stefano durante (to Everyone): 22:03: guay

Ana Gomez (to Everyone): 22:03: por mi, perfecto

Ester (to Everyone): 22:03: perfecto

stefano durante (to Everyone): 22:04: Buenas noches

Ana Gomez (to Everyone): 22:04: genial!

Miguel Ángel Vázquez Vera (to Everyone): 22:04: hasta luego!!

Amaia (to Everyone): 22:04: vale, gracias

Manuel Larena Anchuela (to Everyone): 22:04: muchas gracias

Miguel Ángel Vázquez Vera (to Everyone): 22:04: gracias!

Iván (to Everyone): 22:04: adios

Ester (to Everyone): 22:04: hasta mañana!

Oscar Moreno (to Everyone): 22:04: gracias, buenas noches

Ana Gomez (to Everyone): 22:04: Me ha gustado mucho la clase! Gracias!

Ferran Puig Martínez (to Everyone): 22:04: Venga muchas gracias!

Amaia (to Everyone): 22:04: hasta mañana

Manuel Larena Anchuela (to Everyone): 22:04: hasta mañana

Ferran Puig Martínez (to Everyone): 22:04: buenas noches1

Ana Gomez (to Everyone): 22:04: Hasta mañana!

Patri Fdez. (to Everyone): 22:04: muchas gracias

Gema Pujalte Iglesias (to Everyone): 22:04: Bona nit!!