

# WMNFE 2210

## FRONT-END DEVELOPMENT



### JavaScript #8 | Regular expressions





# ΠΕΡΙΕΧΟΜΕΝΑ

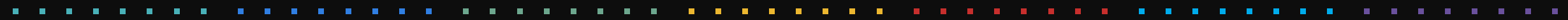


# Περιεχόμενα

- Regular expressions
  - Brackets
  - Ειδικοί χαρακτήρες
  - Quantifiers
  - Anchors
  - Modifiers
  - Μέθοδοι **RegExp**



# REGULAR EXPRESSIONS





# Regular expressions

## Intro

Τα **Regular expressions** (**RegExp**) είναι μια σειρά από χαρακτήρες που ορίζουν ένα μοτίβο (**pattern**).

Που μπορούν να χρησιμοποιηθούν τα μοτίβα αυτά;

- Αναζήτηση
- Έλεγχοι
- Αντικατάσταση



# Regular expressions

## Σύνταξη

Η σύνταξη τους γίνεται με τη χρήση του slash ως εξής: **/pattern/**

Ανάμεσα στα δύο slash βρίσκεται το μοτίβο και μετά το δεξιό slash πιθανόν να υπάρχουν flags που επηρεάζουν την λειτουργία του πρότυπου.

```
const re = /pattern/flags;
```

Εναλλακτικά, μπορούν να δημιουργηθούν και ως εξής:

```
const re = new RegExp('pattern', 'flags');
```



# Regular expressions

## Brackets

Ορίζουν ένα εύρος χαρακτήρων

Expression	Περιγραφή
<b>[abc]</b>	Οποιοσδήποτε από τους χαρακτήρες <b>abc</b>
<b>[^abc]</b>	Κανένας από τους χαρακτήρες <b>abc</b>
<b>[0-9]</b>	Οποιοδήποτε αριθμητικό ψηφίο
<b>[^0-9]</b>	Όχι αριθμητικό ψηφίο
<b>(x   y   z)</b>	Οποιαδήποτε από τις εναλλακτικές <b>x, y, z</b>



# Regular expressions

## Ειδικοί χαρακτήρες

Metacharacter	Περιγραφή
<b>\n</b>	Χαρακτήρας νέας γραμμής
<b>\r</b>	Χαρακτήρας επιστροφής
<b>.</b>	Χαρακτήρας εκτός των <b>\n</b> , <b>\r</b>
<b>\w \W</b>	Χαρακτήρας λέξης ή μη
<b>\d \D</b>	Αριθμητικό ψηφίο ή μη
<b>\s \S</b>	"Κενός" χαρακτήρας ή μη





# Regular expressions

## Ειδικοί χαρακτήρες

$\backslash w \Leftrightarrow [a-zA-Z0-9\_]$

$\backslash W \Leftrightarrow [^a-zA-Z0-9\_]$

$\backslash d \Leftrightarrow [0-9]$

$\backslash D \Leftrightarrow [^0-9]$

$\backslash s \Leftrightarrow [\backslash r \backslash n \backslash t]$

$\backslash S \Leftrightarrow [^ \backslash r \backslash n \backslash t]$



# Regular expressions

## Quantifiers

Quantifier	Περιγραφή
<b>n+</b>	Τουλάχιστον μια φορά
<b>n*</b>	Από 0 έως "άπειρες" φορές
<b>n?</b>	0 ή 1 φορές
<b>n{X}</b>	Ακριβώς X φορές
<b>n{X,Y}</b>	Από X έως Y φορές
<b>n{X,}</b>	Από X έως "άπειρες" φορές



# Regular expressions

## Quantifiers

$n^+ \Leftrightarrow n\{1,\}$

$n^* \Leftrightarrow n\{0,\}$

$n? \Leftrightarrow n\{0,1\}$

$nnn \Leftrightarrow n\{3\}$



# Regular expressions

## ⌈ Anchors

Anchor	Περιγραφή
<b>^n</b>	Στην αρχή
<b>n\$</b>	Στο τέλος



# Regular expressions

## Modifiers

Αλλάζουν τη συμπεριφορά ενός **Regex**

Modifier	Περιγραφή
<b>g</b>	Καθολικό ταίριασμα (global match)
<b>i</b>	Case-insensitive ταίριασμα
<b>m</b>	Multiline ταίριασμα



# Regular expressions

## Μέθοδοι **RegExp**

Μέθοδος	Περιγραφή
<b>exec()</b>	Επιστρέφει το πρώτο ταίριασμα, αν υπάρχει
<b>test()</b>	Ελέγχει για ταίριασμα. Επιστρέφει <b>true</b> ή <b>false</b>



# Regular expressions

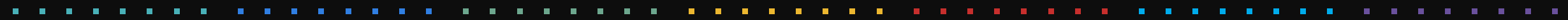
## Μέθοδοι **RegExp**

Μία τελευταία (εξαιρετικά χρήσιμη) μέθοδος, που αφορά στα **RegExp**, αλλά ανήκει στο αντικείμενο **String**. Επιστρέφει όλα τα ταιριάσματα ενός **RegExp** σε πίνακα, αρκεί να συμπεριληφθεί ο modifier **g**

```
str.match(/regex/g)
```



# CLASSWORK







# Classwork

## Classwork #1

Δημιουργήστε μία φόρμα η οποία θα περιέχει τα εξής πεδία:

- Όνομα
- Ηλικία
- E-mail

Εφαρμόστε ελέγχους εγκυρότητας στο **submit** event της φόρμας, με την χρήση των **isNaN()**, **indexOf()** και **length**, και σε περίπτωση λάθους εμφανίστε κατάλληλα μηνύματα στον χρήστη.

Π.χ στο **console** ή συγκεντρωτικά σε κάποιο **div**

```
<div id="error_messages"></div>
```



# Classwork

## Classwork #2

Δημιουργήστε ένα **RegExp** που να ελέγχει μια διεύθυνση email. Το μοτίβο πρέπει να υλοποιεί τους εξής κανόνες:

- Το πρώτο κομμάτι της διεύθυνσης περιέχει μόνο λατινικούς χαρακτήρες ή το χαρακτήρα '.'
- Ακολουθεί ο χαρακτήρας '@'
- Ακολουθεί το domain, που απαρτίζεται μόνο από λατινικούς χαρακτήρες ή νούμερα
- Ακολουθεί ο χαρακτήρας '.'
- Ακολουθεί το top-level domain που απαρτίζεται μόνο από λατινικούς χαρακτήρες, μήκους 2 έως 10

**ΣΗΜ:** Το παραπάνω **RegExp** δεν αποτελεί πλήρη κανόνα, για τον έλεγχο μιας οποιαδήποτε διεύθυνσης e-mail.



# Χρήσιμα links

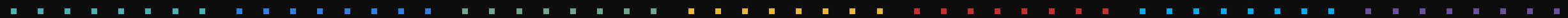
✂ Methods of RegExp and String  
<https://javascript.info/regexp-methods>

ES Regular expressions :: Eloquent JavaScript  
[https://eloquentjavascript.net/09\\_regexp.html](https://eloquentjavascript.net/09_regexp.html)

M Regular expressions - JavaScript | MDN  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp)

M RegExp - JavaScript | MDN  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp)

w JavaScript RegExp Reference  
[https://www.w3schools.com/jsref/jsref\\_obj\\_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp)





## Extra info

.\* RegExr: Learn, Build, & Test RegEx

<https://regexr.com/>

M Client-side form validation - Learn web development ...

<https://developer.mozilla.org/en-US/docs/Learn/Forms/...>

📄 regex - How to validate an email address using a reg...

<https://stackoverflow.com/questions/201323/how-to-va...>

W ASCII - Simple English Wikipedia, the free encyclopedia

<https://simple.wikipedia.org/wiki/ASCII>



# THANK YOU!