Προγραμματισμός Διαδικτύου

ΠαΜακ 2023

# HTML

## Basic

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title></title>

<link rel="stylesheet" href="style.css">

</head>

<body>

</body>

<script src="script.js"></script>

</html>

—----------------------------------------------------------------------

<hr> - horizontal line

<br>- break

<div></div>

—----------------------------------------------------------------------

## Lists

unordered: <ul>

<li></li>

</ul>

ordered: <ol>

<li></li>

</ol>

## Img/Video

img: <img src = “link” alt=””>

video: <video src = “link” height=”” width=”” controls>

## Link

text ref: <a href= “link”> text </a>

img ref: <a href= “link”> <img src = “link”> </a>

## Table

<table>

<caption>text</caption>

<thead></thead>

<tbody>

<tr>

<th></th>

<th></th>

<th></th>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

<td></td>

</tr>

</tbody>

<tfoot></tfoot>

</table>

colspan=””

rowspan=””

inline w/ th/td

## Button

<button>button-text</button>

# CSS

whole background: body{background-color: …. }

for classes: .class-name{}

for ids: #id-name{} -html-> <selector id=”id-name”>

for pseudo classes: selector:class-name{}

descendant selector: X Y (the Y’s that are in X’s)

child selector: X > Y

The [font-family](https://www.codecademy.com/resources/docs/css/typography/font-family) property defines the typeface of an element.

[font-size](https://www.codecademy.com/resources/docs/css/typography/font-size) controls the size of text displayed.

[font-weight](https://www.codecademy.com/resources/docs/css/typography/font-weight) defines how thin or thick text is displayed.

The [text-align](https://www.codecademy.com/resources/docs/css/typography/text-align) property places text in the left, right, or center of its parent container.

Text can have two different color attributes: [color](https://www.codecademy.com/resources/docs/css/colors/color) and [background-color](https://www.codecademy.com/resources/docs/css/background/background-color). color defines the color of the text, while background-color defines the color behind the text.

CSS can make an element transparent with the [opacity](https://www.codecademy.com/resources/docs/css/colors/opacity) property.

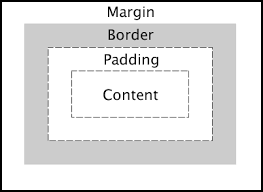
CSS can also set the background of an element to an image with the [background-image](https://www.codecademy.com/resources/docs/css/background/background-image) property.

The !important flag will override any style, however it should almost never be used, as it is extremely difficult to override.

## The box model

border:1px solid;

comprises a set of properties used to create space around and between HTML elements.

The height and width of a content area can be set in pixels or percentages.

* Borders: surround the content area and padding of an element. The color, style, and thickness of a [border](https://www.codecademy.com/resources/docs/css/borders/border) can be set with CSS properties.
* border-radius: roundness of corners (50% for circle)
* Padding: the space between the content area and the border. It can be set in pixels or percent.
* Margin: the amount of spacing outside of an element’s border.
  + - Horizontal margins add, so the total space between the borders of adjacent elements is equal to the sum of the right [margin](https://www.codecademy.com/resources/docs/css/margins/margin) of one element and the left margin of the adjacent element.
    - Vertical margins collapse, so the space between vertically adjacent elements is equal to the larger margin.
* margin: 0 auto: horizontally centers an element inside of its parent content area, if it has a width.
* The [overflow](https://www.codecademy.com/resources/docs/css/overflow/overflow) property can be set to [display](https://www.codecademy.com/resources/docs/css/display), hidden, or scroll, and dictates how HTML will render content that overflows its parent’s content area.
* The [visibility](https://www.codecademy.com/resources/docs/css/visibility) property can hide or show elements.

## Positioning

text-align: center/ end/ justify/ left/right/ start;

* The [position](https://www.codecademy.com/resources/docs/css/position/position) property allows you to specify the position of an element.
* When set to relative, an element’s position is relative to its default position on the page.
* When set to absolute, an element’s position is relative to its closest positioned parent element. It can be pinned to any part of the web page, but the element will still move with the rest of the document when the page is scrolled.
* When set to fixed, an element’s position can be pinned to any part of the web page. The element will remain in view no matter what.
* When set to sticky, an element can stick to a defined offset position when the user scrolls its parent container.
* The [z-index](https://www.codecademy.com/resources/docs/css/position/z-index) of an element specifies how far back or how far forward an element appears on the page when it overlaps other elements.
* The [display](https://www.codecademy.com/resources/docs/css/display) property allows you to control how an element flows vertically and horizontally in a document.
* inline elements take up as little space as possible, and they cannot have manually adjusted [width](https://www.codecademy.com/resources/docs/css/sizing/width) or [height](https://www.codecademy.com/resources/docs/css/sizing/height).
* block elements take up the width of their container and can have manually adjusted heights.
* inline-block elements can have set width and height, but they can also appear next to each other and do not take up their entire container width.
* The [float](https://www.codecademy.com/resources/docs/css/display/float) property can move elements as far left or as far right as possible on a web page.
* You can clear an element’s left or right side (or both) using the [clear](https://www.codecademy.com/resources/docs/css/display/clear) property.

## Fonts

* Text can appear bold or thin with the [font-weight](https://www.codecademy.com/resources/docs/css/typography/font-weight) property.
* Text can appear in italics with the [font-style](https://www.codecademy.com/resources/docs/css/typography/font-style) property.
* The vertical spacing between lines of text can be modified with the [line-height](https://www.codecademy.com/resources/docs/css/typography/line-height) property.
* Fallback fonts are used when a certain font is not installed on a user’s computer font-family: ‘font1’, ‘font2’ …..
* The [word-spacing](https://www.codecademy.com/resources/docs/css/typography/word-spacing) property changes how far apart individual words are.
* The [letter-spacing](https://www.codecademy.com/resources/docs/css/typography/letter-spacing) property changes how far apart individual letters are.
* The text-align property changes the horizontal alignment of text.
* Local fonts can be added to a document with the @font-face property and the path to the font’s source.

## FlexBox

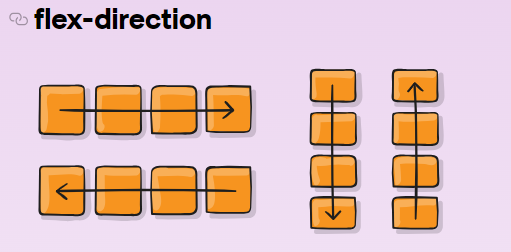
(<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>)

.container {

display: flex; /\* or inline-flex \*/

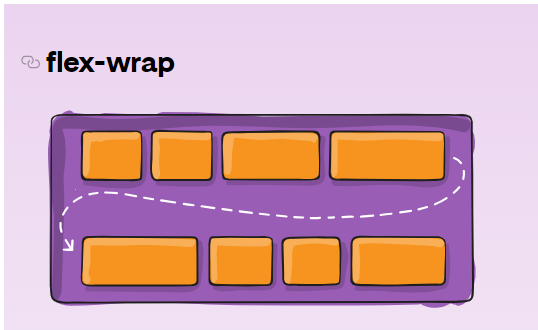
}

### *Properties for the Parent (Flex container)*

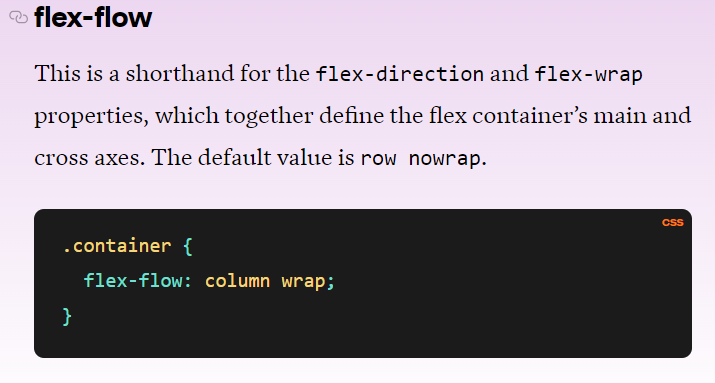


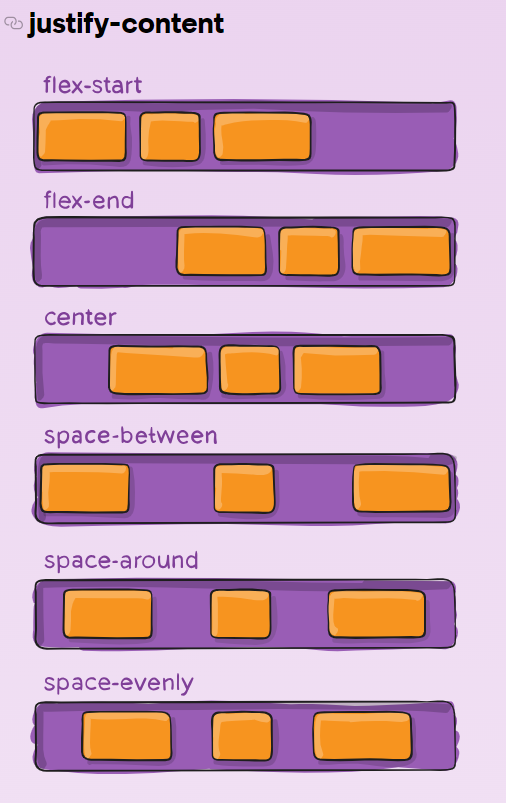
καθορίζει την κατεύθυνση με την οποία τοποθετούνται τα αντικείμενα στο container, και αποδέχεται τις εξής αξίες:

* row: Τα αντικείμενα τοποθετούνται στην ίδια κατεύθυνση με το κείμενο.
* row-reverse: Τα αντικείμενα τοποθετούνται σε κατεύθυνση αντίθετη από το κείμενο.
* column: Τα αντικείμενα τοποθετούνται από πάνω προς τα κάτω.
* column-reverse: Τα αντικείμενα τοποθετούνται από κάτω προς τα πάνω.

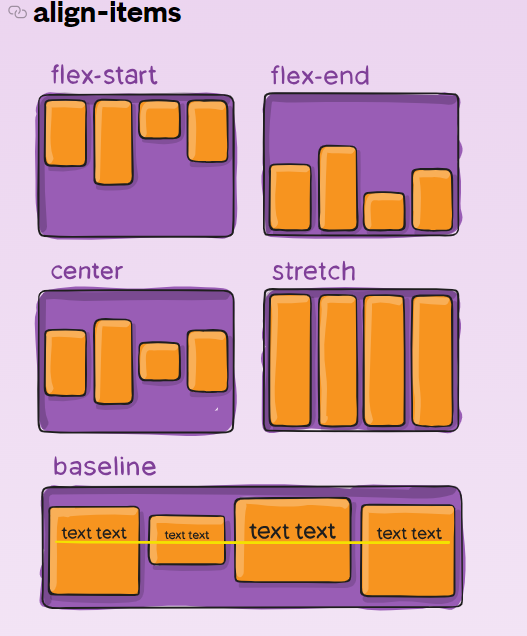


* nowrap: Όλα τα αντικείμενα τοποθετούνται σε μία μόνο γραμμή.
* wrap:
* wrap-reverse: Τα αντικείμενα τυλίγονται σε επιπρόσθετες γραμμές με αντίστροφη σειρά



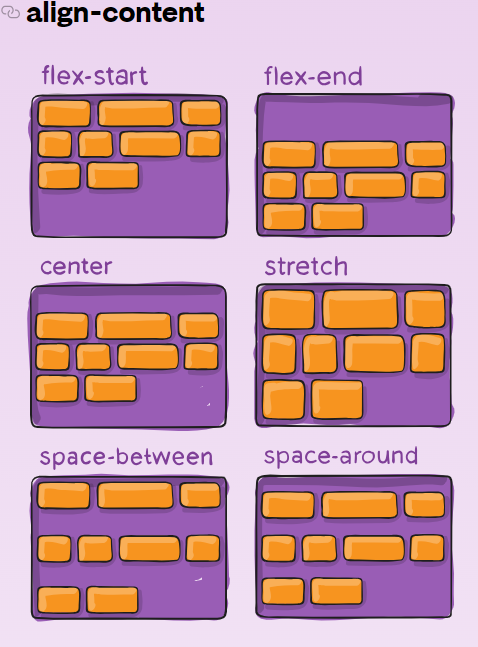
ευθυγραμμίζει αντικείμενα στον οριζόντιο άξονα και αποδέχεται τις εξής αξίες:

* flex-start: Τα αντικείμενα ευθυγραμμίζονται στην αριστερή πλευρά του container.
* flex-end: Τα αντικείμενα ευθυγραμμίζονται στην δεξιά πλευρά του container.
* center: Τα αντικείμενα ευθυγραμμίζονται στο κέντρο του container.
* space-between: Τα αντικείμενα εμφανίζονται με ίσο διάστημα ανάμεσά τους.
* space-around: Τα αντικείμενα εμφανίζονται με ίσο διάστημα γύρω τους.

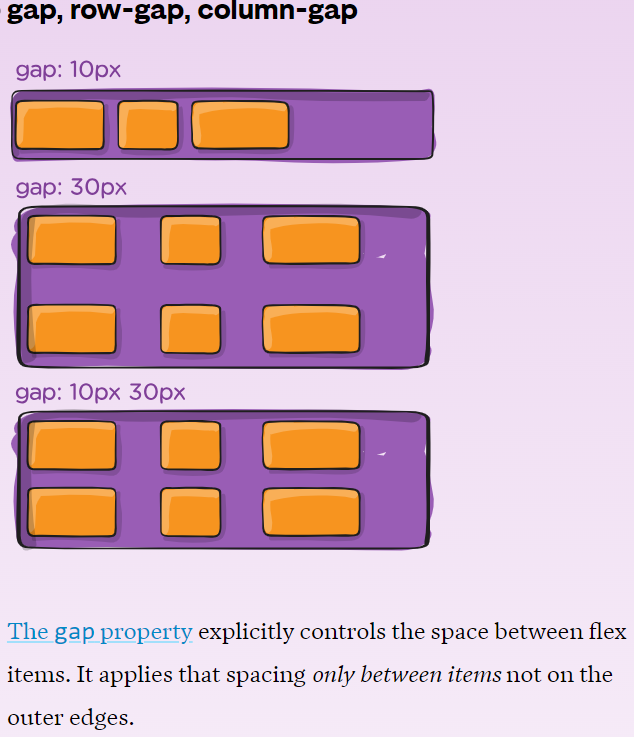


ευθυγραμμίζει αντικείμενα στον κάθετο άξονα και αποδέχεται τις εξής αξίες:

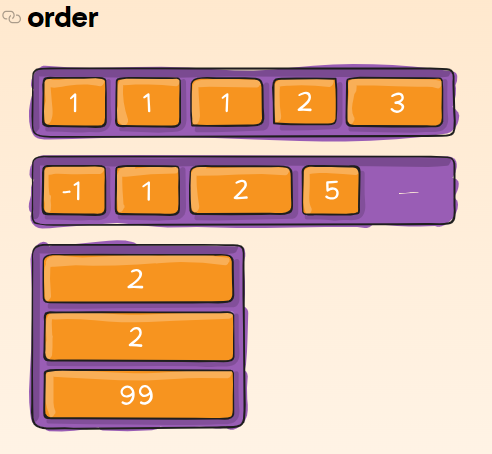
* flex-start: Τα αντικείμενα ευθυγραμμίζονται στο πάνω μέρος του container.
* flex-end: Τα αντικείμενα ευθυγραμμίζονται στο κάτω μέρος του container.
* center: Τα αντικείμενα ευθυγραμμίζονται στο κάθετο κέντρο του container.
* baseline: Τα αντικείμενα εμφανίζονται στην βασική γραμμή του container.
* stretch: Τα αντικείμενα τεντώνονται για να ταιριάξουν στο container.

για να ορίσετε πως πολλαπλές γραμμές χωρίζονται με διάστημα η μία από την άλλη. Αυτή η ιδιότητα παίρνει τις εξής αξίες:

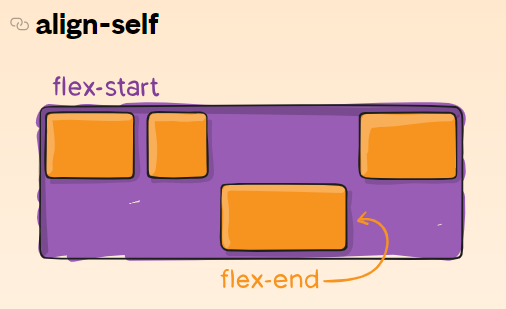
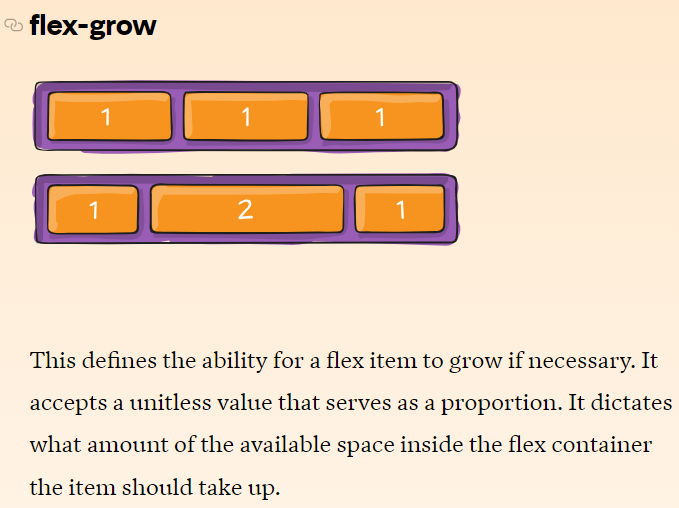
* flex-start: Οι γραμμές τοποθετούνται στο πάνω μέρος του container.
* center: Οι γραμμές τοποθετούνται στο κάθετο κέντρο του container.
* flex-end: Οι γραμμές τοποθετούνται στο κάτω μέρος του container.
* space-between: Οι γραμμές εμφανίζονται με ίσο διάστημα μεταξύ τους.
* space-around: Οι γραμμές εμφανίζονται με ίσο διάστημα γύρω τους.
* stretch: Οι γραμμές τεντώνονται για να ταιριάξουν στο container.



### *Properties for the Children (Flex items)*



order: Μερικές φορές η αντιστροφή της σειράς ταξινόμησης των σειρών ή των στηλών ενός container δεν είναι αρκετή. Σ' αυτές τις περιπτώσεις, μπορούμε να εφαρμόσουμε την ιδιότητα order σε μεμονωμένα αντικείμενα. Από προεπιλογή, τα αντικείμενα έχουν αξία 0, αλλά μπορούμε να χρησιμοποιήσουμε αυτή την ιδιότητα για να την ορίσουμε σε μία θετική ή αρνητική ακέραιη αξία.



This allows the default alignment (or the one specified by align-items) to be overridden for individual flex items.

# Grid

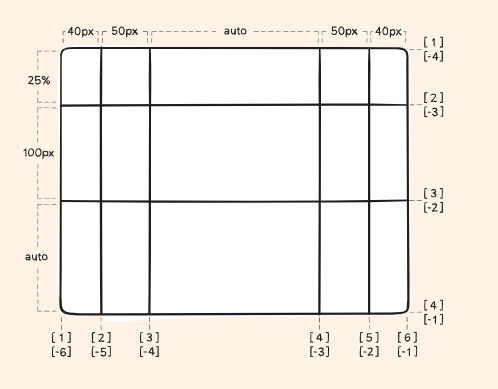
(<https://css-tricks.com/snippets/css/complete-guide-grid/>)

.container {

display: grid | inline-grid;

}

### *Properties for the Parent (Grid container)*



-grid-template-columns: [line-name] track-size

-grid-template-rows: [line-name] track-size

-grid-column-start/grid-column-end: line-name track-size

από που ξεκινάει και που τελειώνει μια στήλη μπορεί να είναι και span x ή x/y (start/end)

-grid-row-start/grid-row-end:line-name track-size

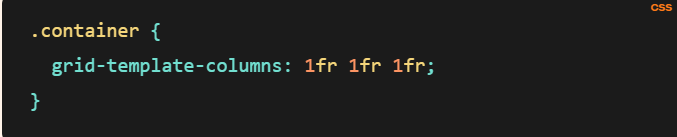
από που ξεκινάει και που τελειώνει μια στήλη μπορεί να είναι και span x ή x/y (start/end)

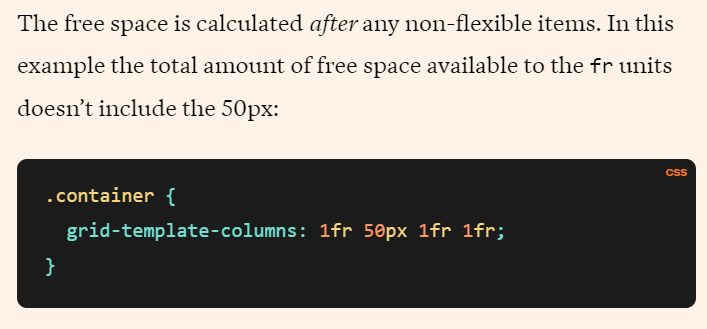
* <track-size> – can be a length, a percentage, or a fraction of the free space in the grid
* <line-name> – an arbitrary name of your choosing (optional)

If your definition contains repeating parts, you can use the [repeat() notation](https://www.digitalocean.com/community/tutorials/css-css-grid-layout-repeat-notation?utm_medium=content_acq&utm_source=css-tricks&utm_campaign=&utm_content=awareness_bestsellers) to streamline things:



The fr unit allows you to set the size of a track as a fraction of the free space of the grid container. For example, this will set each item to one third the width of the grid container:



!

### *Properties for the Children (Grid items)*

# JavaScript

<script src="js file name"></script>

print : console.log();

get element: document.getElementById(“”);

make a button do something: <button onclick=”function() ορισμένη στο script.js”>button-text</button> in html

or

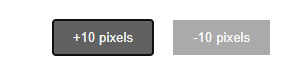
buttonID.addEventListener('click', () => {

}); in js

id.innerHTML to extract the text from an element and Number(id.innerHTML) to turn a string into a number

remember to declare as const all the ids youll be using and remember to set them as id NOT CLASS

# Έτοιμα Buttons (CSS)



.buttons{

display: flex;

gap: 20px;

justify-content: center;

}

button{

color: white;

background-color: rgb(97, 97, 97);

border: none;

align-items: center;

padding: 10px 20px;

cursor: pointer;

}

.buttons button:hover {

background-color: rgb(170, 170, 170);

}

# Έτοιμο Box(CSS)

.box{

margin-top: 10px;

width: 100px;

height: 100px;

border: 2px solid;

align-self: center;

text-align: center;

line-height: 75px;

}

OR

#box{

display: flex;

align-items: center;

justify-content: center;

margin-top: 10px;

width: 100px;

height: 100px;

border: 2px solid;

text-align: center;

line-height: 100px;

align-self: center;

}

# PHP

## index

<!DOCTYPE html>

<html>

<head>

<title>Θεοχαρίδη Φαίδρα - ics22069</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0, maximum-scale=1.0, minimal-ui">

</head>

<body>

<h1>Θεοχαρίδη Φαίδρα - ics22069</h1>

<br>

<ul>

<li><a href="">name</a></li> //add other pages

</ul>

</body>

</html>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069"; //allaje analoga me to onoma tou db

$tabname = "ics22069"; //allaje analoga me to onoma tou pinaka

// Create connection

$conn = mysqli\_connect($servername, $username, $password);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Attempt create database query execution

$sql = "CREATE DATABASE IF NOT EXISTS $dbname";

if (mysqli\_query($conn, $sql)) {

echo "Database created successfully <br>";

} else {

echo "Error creating database: " . mysqli\_error($conn) . "<br>";

}

mysqli\_close($conn);

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Create connection to already existin db

// $conn = mysqli\_connect($servername, $username, $password, //$dbname);

// Check connection

//if (!$conn) {

// die("Connection failed: " . mysqli\_connect\_error());

//}

// sql to create table

$sql = "CREATE TABLE IF NOT EXISTS $tabname (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

website VARCHAR(50) NOT NULL

)"; //change analoga me ti zhtaei

// table creation message

if (mysqli\_query($conn, $sql)) {

echo "Table $tabname created successfully <br>";

} else {

echo "Error creating table: " . mysqli\_error($conn) . "<br>";

}

mysqli\_close($conn);

?>

## Εμφάνιση όλων των εγγραφών

<!DOCTYPE html>

<html>

<head>

<title>Θεοχαρίδη Φαίδρα - ics22069</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0, maximum-scale=1.0, minimal-ui">

</head>

<body>

<h1>Θεοχαρίδη Φαίδρα - ics22069</h1>

<br>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069"; //allaje analoga me to onoma tou db

$tabname = "ics22069"; //allaje analoga me to onoma tou pinaka

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Outputs all database entries for $tabname

$sql = "SELECT \* FROM $tabname"; //change analoga me to ti zhtaei

$result = mysqli\_query($conn, $sql);

echo "<br>";

if (mysqli\_num\_rows($result) > 0) {

while($row = mysqli\_fetch\_assoc($result)) {

echo "id: " . $row["id"]. " - Name: " . $row["name"]. " - Website: " . $row["website"]. "<br>"; //allaje analoga me to ti zhtaei

}

} else {

echo "0 results <br>";

}

mysqli\_close($conn);

?>

<ul>

<li><a href="index.php">Αρχική</a></li>

//add other pages

</ul>

</body>

</html>

## Εμφάνιση μερικών εγγραφών

<!DOCTYPE html>

<html>

<head>

<title>Θεοχαρίδη Φαίδρα - ics22069</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0, maximum-scale=1.0, minimal-ui">

</head>

<body>

<h1>Θεοχαρίδη Φαίδρα - ics22069</h1>

<br>

<ul>

<li><a href="home.php">Αρχική</a></li>

<li><a href="show.php">Εμφάνιση</a></li>

<li><a href="set.php">Καταχώρηση</a></li>

</ul>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069"; //allaje analoga me to onoma tou db

$tabname = "ics22069"; //allaje analoga me to onoma tou pinaka

$conn = mysqli\_connect($servername, $username, $password, $dbname);

//emfanish stoixeiwn pou periexoun sugkekrimeno string

// Outputs all database entries for $tabname

$sql = "SELECT \* FROM $tabname";

$result = mysqli\_query($conn, $sql);

echo "<br>";

if (mysqli\_num\_rows($result) > 0) {

// output data of each row

while($row = mysqli\_fetch\_assoc($result)) {

if (str\_contains($row["id"], '21027')) { //change analoga me ekfwnhsh

echo "ID (ends with ID " . $row["id"] .") : " . " - Name: " .

$row["name"]. " - Website: " . $row["website"]. "<br>";

}

}

} else {

echo "0 results";

}

mysqli\_close($conn);

// Outputs all database entries for $tabname

$sql = "SELECT kati FROM $tabname"; //allaje

$result = mysqli\_query($conn, $sql);

echo "<br>";

if (mysqli\_num\_rows($result) > 0) {

while($row = mysqli\_fetch\_assoc($result)) {

echo "id: " . $row["id"]. " - Email: " . $row["email"]. "<br>";

}

} else {

echo "0 results <br>";

}

mysqli\_close($conn);

?>

</body>

</html>

!sel. 120, emfanish se morph table

## Καταχώρηση Νέων Δεδομένων (μέσω φόρμας)

<!DOCTYPE html>

<html>

<head>

<title>Θεοχαρίδη Φαίδρα - ics22069</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0, maximum-scale=1.0, minimal-ui">

<style>

.error {

color: #FF0000;

}

</style>

</head>

<body>

<h1>Θεοχαρίδη Φαίδρα - ics22069</h1>

<br>

<?php

// define variables and set to empty values

$nameErr = $websiteErr = ""; //allaje analoga me thn efvnhsh

$name = $id = $website = ""; //allaje analoga me thn efvnhsh

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$id = test\_input($\_POST["id"]);

//check name field

if (empty($\_POST["name"])) {

$nameErr = "Name is required";

} else {

$name = test\_input($\_POST["name"]);

$nameErr = "";

}

//check website field

if (empty($\_POST["website"])) {

$websiteErr = "Website is required";

} else {

$website = test\_input($\_POST["website"]);

// check if URL address syntax is valid (this regular expression also allows dashes in the URL)

if(!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",$website)) {

$websiteErr = "Invalid URL";

}

}

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

// Insert to table

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069";

$tabname = "ics22069";

$conn = mysqli\_connect($servername, $username, $password, $dbname);

if(!empty($website) && empty($websiteErr) && !empty($name) && empty($nameErr)){

// Inserts user info to database

$sql = "INSERT INTO $tabname (id, name, website) VALUES ('$id', '$name', '$website')"; //allaje analoga me thn ekfvnhsh

if (mysqli\_query($conn, $sql)) {

echo "New record created successfully <br>";

} else {

echo "Error: " . $sql . "<br>" . mysqli\_error($conn) . "<br>";

}

}

mysqli\_close($conn);

?>

<p><span class="error">\* required field.</span></p>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

ID: <input type="text" name="id" value="<?php echo $id;?>">

<br><br> //id field

Name: <input type="text" name="name" value="<?php echo $name;?>">

<span class="error">\* <?php echo $nameErr;?></span>

<br><br> //name field

Website: <input type="text" name="website" value="<?php echo $website;?>">

<span class="error">\* <?php echo $websiteErr;?></span>

<br><br> //website field

<input type="submit" name="submit" value="Submit">

</form> //change fields analoga me thn ekfwnhsh

<ul>

<li><a href="index.php">Αρχική</a></li> //add analoga me ekfwnhsh

</ul>

</body>

</html>

## Διαγραφή Δεδομένων

<!DOCTYPE html>

<html>

<head>

<title>Θεοχαρίδη Φαίδρα - ics22069</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1.0, maximum-scale=1.0, minimal-ui">

<style>

.error {

color: #FF0000;

}

</style>

</head>

<body>

<h1>Θεοχαρίδη Φαίδρα - ics22069</h1>

<br>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069";

$tabname = "ics22069";

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { //allaje analoga me ekfwnhsh

$id = $\_POST["id"];

// sql to delete a record

$sql = "DELETE FROM $tabname WHERE id=$id"; //allaje

if (mysqli\_query($conn, $sql)) {

echo "Record deleted successfully";

} else {

echo "Error deleting record: " . mysqli\_error($conn);

}

mysqli\_close($conn);

?>

<h2>Φόρμα Διαγραφής</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<label for="id">ID:</label>

<input type="text" name="id" id="id" required>

<br><br>

<input type="submit" value="Υποβολή">

</form>

<ul>

<li><a href="index.php">Αρχική</a></li>

//add other pages

</ul>

</body>

</html>

## Διόρθωση Δεδομένων

<!DOCTYPE html>

<html>

<head>

<title>Σελίδα Διόρθωσης</title>

</head>

<body>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ics22069";

$tabname = "ics22069";

// Σύνδεση με τη βάση δεδομένων

$conn = new mysqli($servername, $username, $password, $dbname);

// Έλεγχος σύνδεσης

if ($conn->connect\_error) {

die("Σφάλμα κατά τη σύνδεση με τη βάση δεδομένων: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { //allaje analoga me ekfwnhsh

$id = $\_POST["id"];

$name = $\_POST["name"];

// Τροποποίηση δεδομένων στον πίνακα

$sql = "UPDATE $tabname SET name='$name' WHERE id='$id'"; //allaje analoga me ekfwnhsh

if ($conn->query($sql) === TRUE) {

echo "<p>Τροποποίηση δεδομένων επιτυχής!</p>";

} else {

echo "<p>Απέτυχε η τροποποίηση δεδομένων: " . $conn->error . "</p>";

}

}

$conn->close();

?>

<h2>Φόρμα Διόρθωσης</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<label for="id">ID:</label>

<input type="text" name="id" id="id" required>

<br><br>

<label for="name">Name:</label>

<input type="text" name="name" id="name" required>

<br><br>

<input type="submit" value="Υποβολή">

</form>

</body>

</html>