

# Web Development Workshop

## HTML - CSS

*By Lampros Makris*

*special thanks : Konstantinos Stavrou*

# Workshop Overview

## 1<sup>st</sup> Day (17/10)

- **Web Development Introduction**
- **HTML Basics**
- **CSS Basics**
- **Project Introduction**

## 2<sup>nd</sup> Day(20/10)

- **Advanced HTML**
- **Advanced CSS**
- **HTML & CSS in Action**
- **Project Discussion**

# Before we start

**A few words about me....**

**Lampros 30**

**sleed.gr**

**tedxkidsilissos.com**

**<https://www.united-europe.eu/>**

**DIT**

**Functional Programming**

**Web Development**

**Ecommerce**

**Prestashop**

# Before we start

**Let 's get to know us better....**

**Let 's play a game....**

**Only 10 questions....**

**<https://play.kahoot.it>**

# Web Development Introduction

# What is Web Development ?

Web development is the coding or programming that enables website functionality.

Web development is also known as website development.

Web development ranges from creating plain text pages to complex web-based applications, social network applications and electronic business applications.

The **web development hierarchy** is as follows:

- **Client-side coding (Front-End)**
- **Server-side coding (Back-end)**
- **Database Technology (Data Model)**

*techopedia*

# Client-side

Client-side refers to a specific part of **client/server architecture**, which is a network structure distinguishing clients or computers ordering information from servers, hardware pieces that deliver that information and process requests.

In a traditional client/server structure, clients **consist of physical personal** computers or desktop computer stations. These use web browsers or other connections to make demands on servers.

Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices and even processing power.

**Client-side programming** is writing code that **will run on the client**, and is done in languages that **can be executed by the browser**, such as **JavaScript**.

# Web Browser

A web browser is a **software program that allows a user to locate, access, and display web pages**. In common usage, a web browser is usually shortened to "browser."

Browsers are used primarily for displaying and accessing websites on the Internet, as well as other content created using **Hypertext Markup Language (HTML)** and Extensible Markup Language (**XML**)

Browsers **translate web** pages and websites delivered using Hypertext Transfer Protocol (**HTTP**) into **human readable content**. They also have the ability to display other protocols and prefixes, such as secure HTTP (**HTTPS**), File Transfer Protocol (**FTP**), email handling (mailto:), and files (file:).

*techopedia*



# Most Popular Web Browsers

- **Google Chrome**
- **Mozilla Firefox (Linux)**
- **Internet Explorer (Windows)**
- **Safari (Macintosh)**
- **Opera**
- **Blink (Web Development Specific)**

*webopedia*

# Server-side

Server-side applications or operations are done within the hardware structures that are serving the requests of clients.

**Server-side programming** is writing code that runs on the server, **using languages** supported by the server (**such as Java, PHP, C#**)

**JavaScript**, in occasions is possible to be executed on the server-side (AJAX).

*webopedia*

# Web Server

A Web server is a system that **delivers content or services to end users over the Internet**. A Web server consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication.

The most simple definition is that a Web server runs a website **by returning HTML files over an HTTP connection**.

A better definition might be that a Web server is any Internet server **that responds to HTTP requests** to deliver content and services.

*techopedia*

# Popular Web Servers

- **Apache HTTP Server (52% Websites Global)**
- **Nginx (30% Websites Global)**
- **Apache Tomcat (Java Servlet - 1% Websites Global)**
- **Node.js (Javascript Environment - 0.2% Websites Global)**

# Database Model

Database Systems can be based on different Data Models or Database Models respectively. A Data Model is a collection of concepts and rules for the description of the structure of the Database. Structure of the Database means the Data types, the constraints and the relationships for the description or storage of Data respectively.

The most often **used data models** are:

- **Relational Model (RDBMS - SQL | Oracle , MySQL , SQL SERVER)**
- **Non Relational Model (NRDBMS - NoSQL | MongoDB – CouchDB - BigTable )**

*webopedia*

# HTTP Request

HyperText Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web.

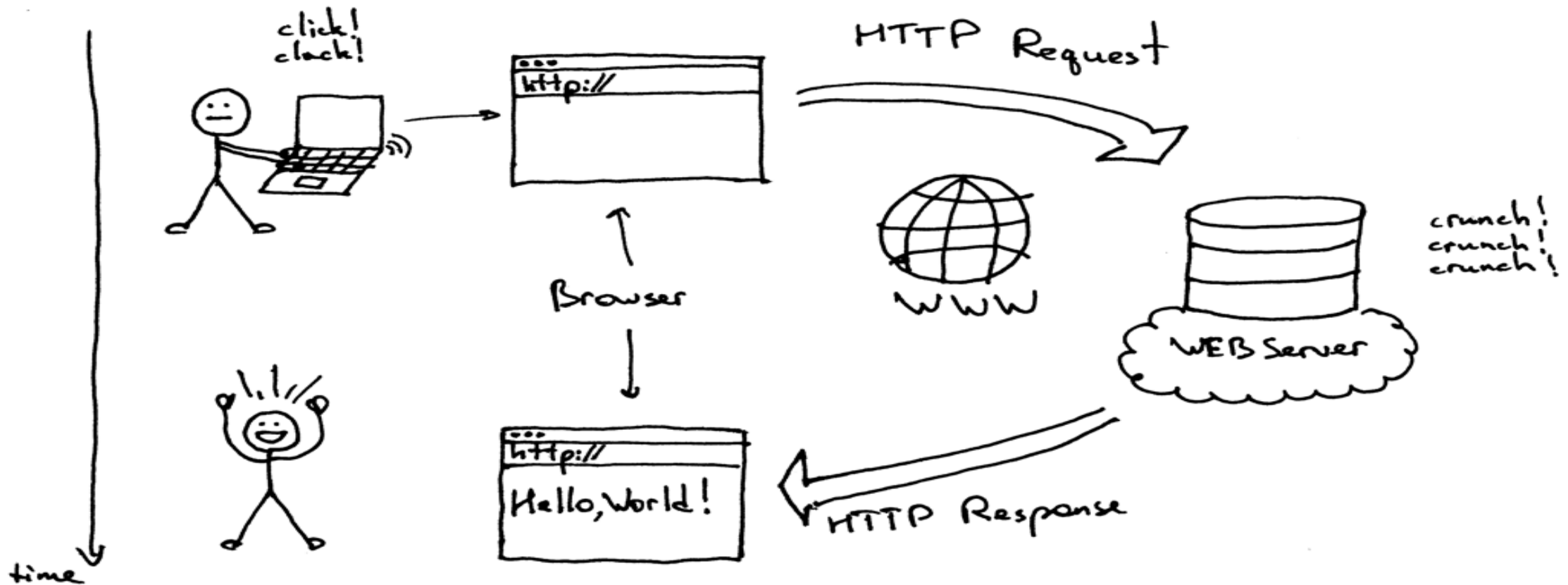
HTTP uses a client-server model where the web browser is the client and communicates with the webserver that hosts the website.

The browser uses HTTP, which is carried over TCP/IP to communicate to the server and retrieve Web content for the user.

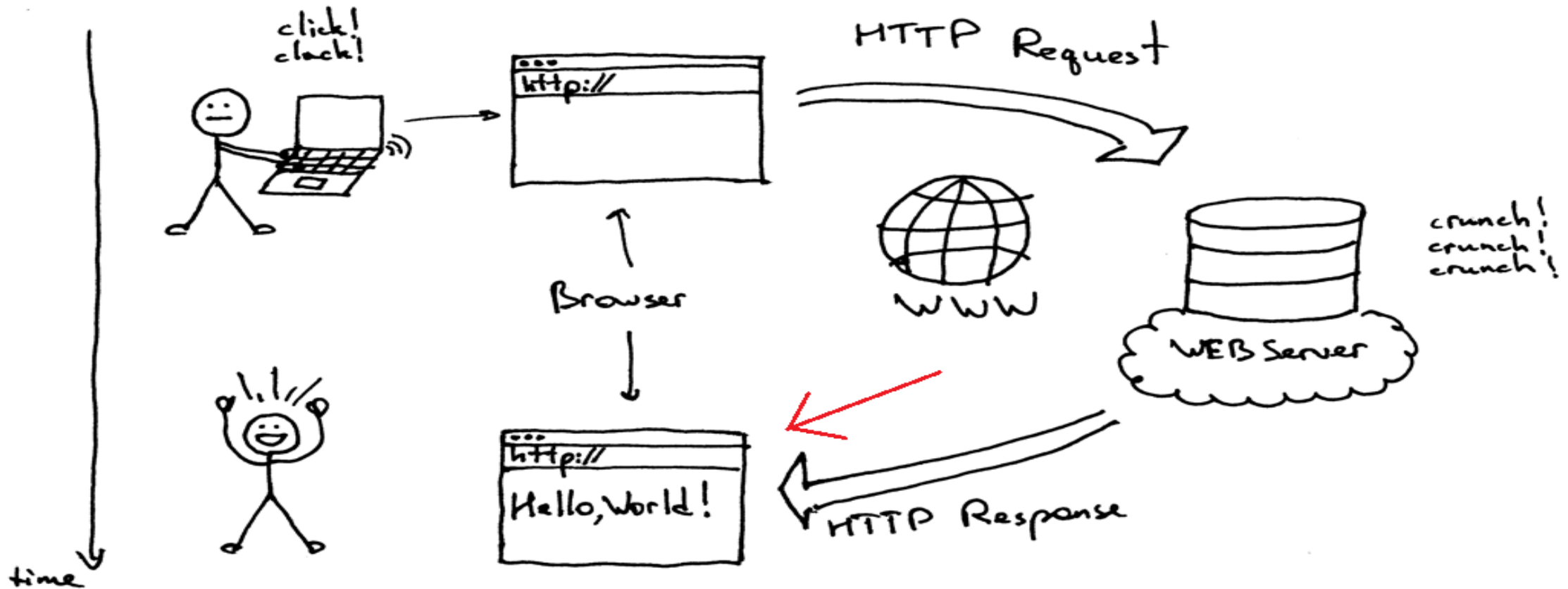
HTTP is a widely used protocol and has been rapidly adopted over the Internet because of its simplicity. It is a stateless and connectionless protocol.

*techopedia*

# The flow when a user enters a URL

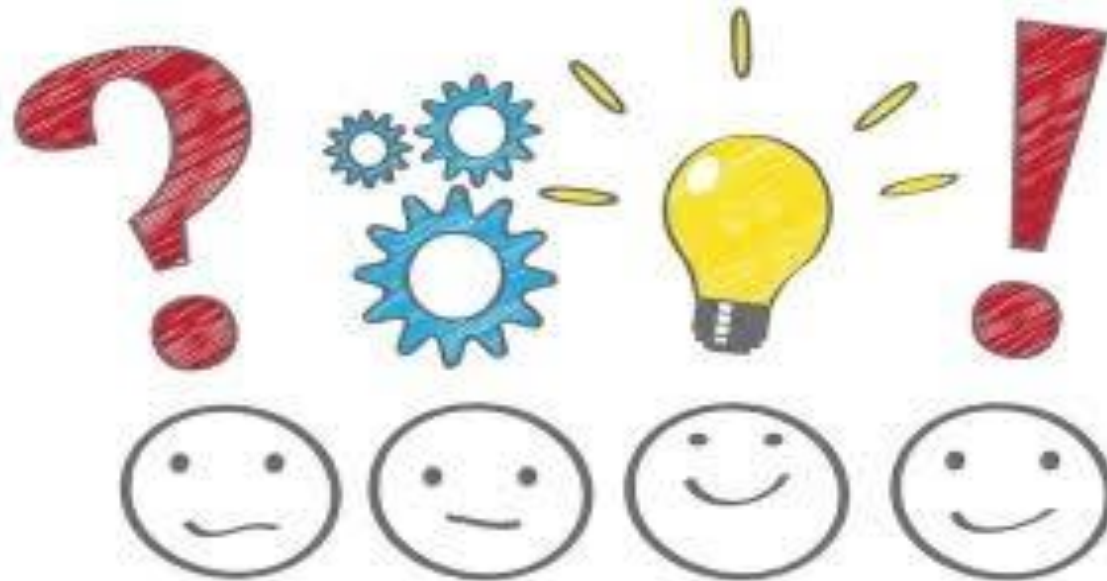


# Red arrow is where front-end interferes !





# Questions ?



# HTML BASICS

# HTML is the standard markup language

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- Browsers do not display the HTML tags, but use them to render the content of the page

*w3schools*

# A Simple HTML document

```
<!DOCTYPE html>
<html>
<head>
<title>Collegelink</title>
</head>
<body>

<h1>Welcome to Web Development Workshop</h1>
<p>I am a web page!</p>

</body>
</html>
```

# TERMS

Η HTML είναι αρκετά απλή γλώσσα. Αποτελείται από 3 βασικά χαρακτηριστικά:

- Elements
- Tags
- Attributes

Element

Tag

`<a href="http://shayhowe.com/">Shay Howe</a>`

Attribute

# HTML elements & tags

Τα HTML elements αποτελούνται συνήθως από ένα starting (opening) tag, και ένα ending (closing) tag, με το περιεχόμενο να περιλαμβάνεται μεταξύ των 2 tag.

`<tagname> Content goes here ... </tagname>`

Start tag	Element content	End tag
<code>&lt;h1&gt;</code>	My First Heading	<code>&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	My first paragraph.	<code>&lt;/p&gt;</code>
<code>&lt;br&gt;</code>		

# HTML elements & tags

Υπάρχουν και κάποια self-closing tags. Αυτά αποκαλούνται και ως “void - elements” . Τα πιο γνωστά που χρησιμοποιούνται και περισσότερο είναι τα :

- `<br />`
- `<embed />`
- `<hr />`
- `<img />`
- `<input />`
- `<link />`

# Use Lowercase Tags

Τα HTML tags δεν είναι case sensitive! Δηλαδή είτε γράψουμε <p> είτε γράψουμε <P> είναι το ίδιο paragraph element για τον browser.

Παρόλα αυτά συνίσταται απ το W3C η χρησιμοποίηση lowercase tags στην HTML, και απαιτείται lowercase χρήση σε πιο αυστηρά document types όπως XHTML.

\* Το W3C είναι ένας διεθνής οργανισμός όπου καθορίζει τα Web standards.



# HTML attributes

Τα HTML attributes προσδίδουν extra πληροφορία για ένα element.

- όλα τα HTML elements μπορούν να έχουν attributes
- τα attributes πάντα προσδιορίζονται στο opening tag του element
- η συντάξη είναι `<tag_name attribute_name="attribute_value">`

# Most Used Attributes

Τα attributes που χρησιμοποιούνται συχνότερα είναι τα :

- href , για το a tag, `<a href = "http://www.collegelink.gr"> Click to visit Collegelink</a>`
- lang, για το html tag, `<html lang="en-us">` . Δηλώνεται η γλώσσα του document που είναι πολύ σημαντικό για parsers , screen readers, μηχανές αναζήτησης (Google)
- src, για το img tag, ``

# HTML Global Attributes

Τα global attributes εφαρμόζουν σε οποιοδήποτε HTML element!

- class : ορίζει ένα όνομα κλάσης για ένα στοιχείο (CSS)
- id : ορίζει ένα **μοναδικό** id για ένα στοιχείο του DOM (CSS, JS)
- style: ορίζει inline style για κάποιο στοιχείο
- hidden: υποστηρίζεται στην HTML5, προσδίδει invisibility, συνήθως μπαίνει σε φόρμες.

# HTML Tags

## <!DOCTYPE>

Είναι η πρώτη γραμμή που πρέπει να υπάρχει στο html αρχείο μας.

Είναι το declaration για τον browser, ουσιαστικά δεν είναι HTML tag, παρά η οδηγία για τον browser για να αναγνωρίσει την έκδοση HTML που χρησιμοποιεί η σελίδα.

Στην HTML5 το declaration γίνεται με έναν τρόπο:

```
<!DOCTYPE html>
```

# HTML Tags

Στην HTML 4.01 υπάρχουν 3 διαφορετικοί τρόποι declaration:

- HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

- HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- HTML 4.01 Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

# HTML <html> Tag

- Το <html> tag “λέει” στον browser ότι έχει να κάνει με ένα HTML document.
- Είναι το 1<sup>ο</sup> tag(root tag) ενός HTML document.
- Περιέχει όλα τα elements της σελίδας πλην του <!DOCTYPE>

# HTML <head> Tag

Το head tag περιέχει όλα τα head elements της σελίδας όπως:

- <title>
- <style>
- <script>
- <noscript>
- <link>

\* Στην HTML 4.01 <head> είναι υποχρεωτικό, ενώ στη HTML5 μπορούμε να το αγνοήσουμε!

# HTML <script> Tag

Το <script> tag ορίζει ένα script(συνήθως JavaScript) που μπορεί να τρέξει στη μεριά του client. Στην HTML 4.01 είναι απαραίτητο να συνοδεύεται απ το attribute **type =“text/javascript”** ενώ στην HTML5 μπορούμε να το αγνοήσουμε.

```
<script type="text/javascript">  
for (i=0;i<10; i++) {  
    console.log(i)  
}  
</script>
```



# HTML `<script>` Tag

Σαν best practice συνηθίζεται να μη γράφουμε “χύμα” κώδικα javascript μέσα στα html αρχεία μας, παρά να τα κάνουμε include στο `<head>` με το attribute `src` όπως φαίνεται παρακάτω.

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

# HTML <link> Tag

Το <link> χρησιμοποιείται για να ορίσει ένα link μεταξύ του document και ενός εξωτερικού συνδέσμου( external resource )

Η κύρια του χρήση είναι να “δείχνει” σε εξωτερικά css αρχεία.

Είναι ένα empty element, περιλαμβάνει μόνο attributes όπως και το <script> .

```
<head>  
  <link rel="stylesheet" type="text/css" href="theme.css">  
</head>
```

# HTML <body> Tag

Το <body> tag περιέχει όλα τα HTML στοιχεία του HTML document:

- κείμενο
- links
- εικόνες
- lists
- tables

# Document Structure

Ένα αρχείο HTML είναι ένα απλό αρχείο txt, το οποίο το αποθηκεύουμε με κατάληξη .htm ή .html .

Example : web\_page.html

# Document Structure

Το πιο απλό HTML document, με τα ελάχιστα required tags.

```
<html>
```

```
<head>
```

```
<title>Site title</title>
```

```
</head>
```

```
<body>
```

Site Content

```
</body>
```

```
</html>
```

Example: simple\_page.html

# HTML <!-- --> Tag

Το <!-- --> είναι το tag όπου μπορούμε να γράφουμε σχόλια στην HTML. Ότι εσωκλείεται σε αυτά τα tags δεν τυπώνεται μπροστά στη σελίδα, αλλά υπάρχει στο source code της σελίδας. (Ctrl + U).

```
<!DOCTYPE html>
```

```
<!-- Hello I am a comment -->
```

```
<html>
```

```
<head>
```

```
<title>Collegelink</title>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to the Web Development Workshop</h1>
```

```
<p>I am a web page.</p>
```

```
</body>
```

```
</html>
```

```
example: comment.html
```

# HTML <a> Tag

Το <a> tag ορίζει ένα hyperlink (σύνδεσμο), που χρησιμοποιείται για να οδηγεί σε μία άλλη σελίδα.

Το href είναι το πιο σημαντικό attribute όπου περνάμε το link που θέλουμε να μας οδηγεί.

Example <a href="collegelink.gr">Collegelink</a>

# HTML <h1> - <h6> Tags

Τα <h1> - <h6> tags χρησιμοποιούνται για να προσδιορίζουν τίτλους περιεχομένων στη σελίδα μας (headings)

```
<!DOCTYPE html>
<html>
<head>
<title>Collegelink</title>
</head>
<body>
<h1>Collegelink Workshop</h1>
<h2>Collegelink Workshop</h2>
<h3>Collegelink Workshop</h3>
<h4>Collegelink Workshop</h4>
<h5>Collegelink Workshop</h5>
<h6>Collegelink Workshop</h6>
</body>
</html> example: headings.html
```



# HTML Block and Inline Elements

Όλα τα στοιχεία καταλαμβάνουν χώρο σε μια HTML σελίδα.

Διαχωρίζονται σε **block elements & inline elements**.

Τα block elements by default ορίζουν μια νέα γραμμή και καταλαμβάνουν όλο τον χώρο που μπορούν. Είναι **full width** δηλαδή.

Αυτά που χρησιμοποιούνται περισσότερο είναι τα:

- <div>
- <h1> - <h6>
- <p>
- <form>

# HTML Block and Inline Elements

Τα inline elements δεν ορίζουν by default νέα γραμμή και καταλαμβάνουν όσο χώρο χρειάζονται στη σελίδα.

Ποια πολύ χρησιμοποιούνται τα :

- `<span>`
- `<a>`
- `<img>`

# HTML <div> Tag

Το <div> tag χρησιμοποιείται συνηθώς σαν container, δηλαδή για να ομαδοποιήσει άλλα elements στην HTML σελίδα μας ώστε να μπορούμε να δώσουμε ένα συγκεκριμένο style στην ομάδα αυτή.  
example: div.html

# HTML `<span>` Tag

Το `<span>` tag έχει παρόμοια χρήση με το `<div>` tag, χρησιμοποιείται και αυτό σαν container, για inline elements όμως.

example : span.html

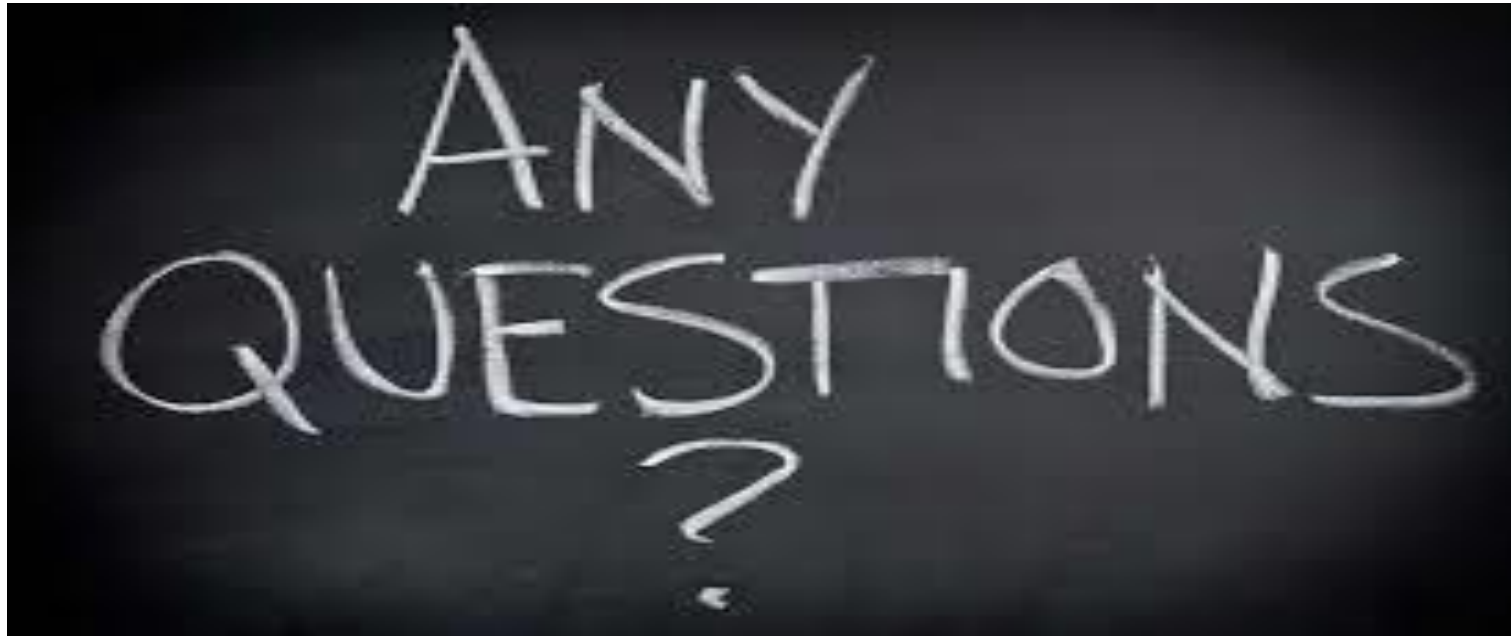
# HTML <img> Tag

Το <img> tag είναι ένα empty tag. Δεν έχει closing tag και περιλαμβάνει attributes όπως src, alt, height, width ...

Το src ορίζει το path της εικόνας ενώ το alt καλό είναι να υπάρχει καθώς δείχνει μια περιγραφή σε περίπτωση που δεν μπορεί να φορτωθεί η εικόνα.

example: image.html

# Questions ?



ANY  
QUESTIONS  
?

# HTML

## Tables & Lists

# HTML Tables

Οι πίνακες (tables) στην HTML ορίζονται με το <table> tag. Κάθε γραμμή(row) ορίζεται με το <tr> , στήλη(data-cell) με το <td> , ενώ ο τίτλος της στήλης με το <th>.

Γενικότερα τα tables θεωρούνται πολύ συμπαγή elements, και χρησιμοποιούνται ιδιαίτερα για την υλοποίηση newsletter, το αποτέλεσμα των οποίων πρέπει να είναι το ίδιο στους διάφορους mail clients. example: tables.html , [http://www.contactpigeon.com/apps/en/stem/viewmail.html?gm\\_pid=401412&gm\\_nID=&crd=7180907&rdme=49255370](http://www.contactpigeon.com/apps/en/stem/viewmail.html?gm_pid=401412&gm_nID=&crd=7180907&rdme=49255370)



# HTML Tables

Υπάρχουν attributes όπως το colspan που αλλάζει τον αρχικό χώρο που θα έπιανε το element. example tables\_1.html

## Συνοψίζοντας:

- HTML **<table>** “ανοίγουμε” το table
- HTML **<tr>** ορίζουμε ένα table row
- HTML **<td>** ορίζουμε τον χώρο για τα δεδομένα του του κελιού
- HTML **<th>** επικεφαλίδα td
- HTML **<caption>** τίτλος πίνακα (μπαίνει μετά το άνοιγμα του table)
- HTML **<tbody>** ομαδοποιεί το content ενός πίνακα (μπαίνει πριν το tr)
- Το **colspan** attribute σε ένα κελί φτιάχνει πολλαπλές στήλες
- Το **rowspan** attribute σε ένα κελί φτιάχνει πολλαπλά rows

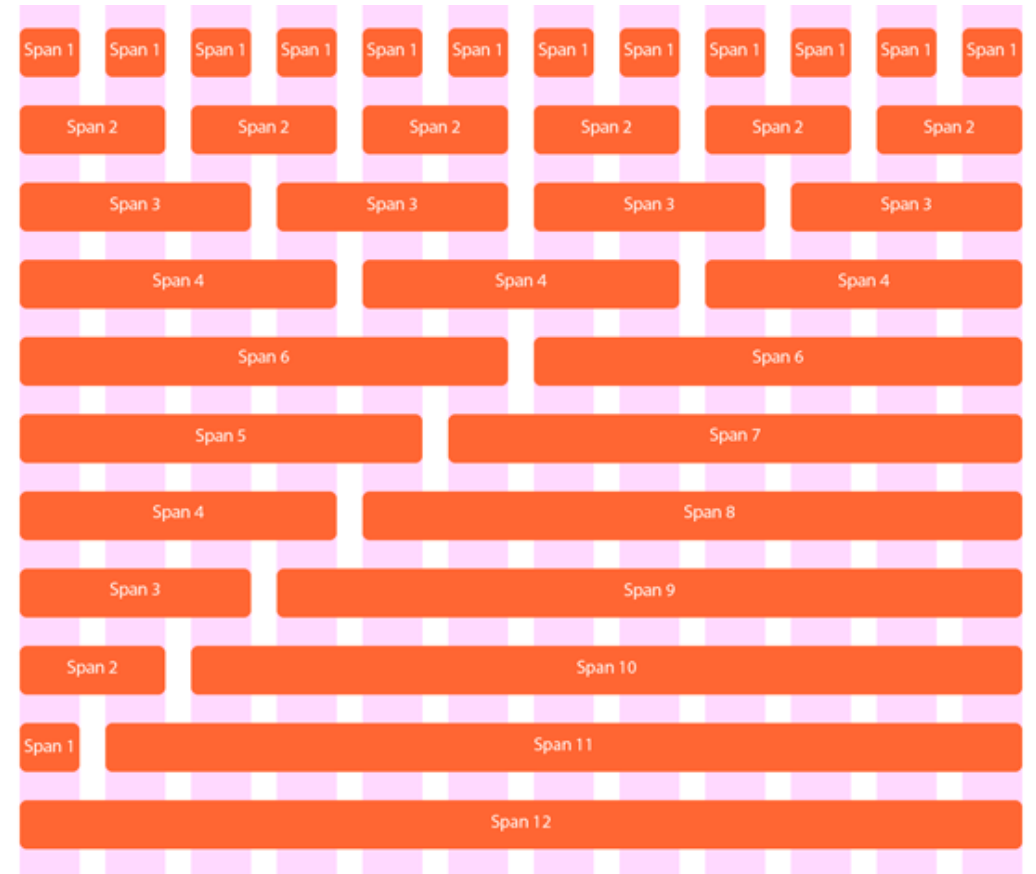
w3schools

# HTML Tables

Product	Barcode	Revenue	Quantity
Korres Pomegranate Mattifying Primer 30ml		€650	20
Korres Ελιξήριο Νεότητας Χρυσός Κροκος 30ml	5203069065163	€639	19
Frezyderm Sun Screen Color Velvet Face Cream SPF50 50ml	5202888222351	€486	37
La Roche Posay Effaclar Duo [+] Cream 40ml	3337872414084	€339	45
Frezyderm Ac-norm Peel-off Mask For Oily / Acne Prone Skin 50ml	5202888228360	€290	31
Medisei Panthenol Extra Face & Eye Cream 24h 50ml & Face & Eye Serum Αντιρυτιδικός Ορός 30ml & Face Cleansing Gel Τζελ Καθαρισμού 150ml & Extra Cream Κρέμα για Ερεθισμένα Δέρματα 100ml	5200312242104	€282	24

# HTML Tables

Χρησιμοποιούνταν για την  
δόμηση των ιστοσελίδων  
καθώς δεν υπήρχαν τότε οι  
τωρινές δυνατότητες.



# HTML Tables

```
<table>
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

# HTML Tables

Για να ορίσουμε επικεφαλίδα είτε σε επίπεδο γραμμής είτε σε επίπεδο στήλης χρησιμοποιούμε το `<th>`

```
<table>
  <tr>
    <th>Name</th>
    <th>Stock</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  ...
</table>
```

Name	Stock	Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

# HTML Tables

Μπορούμε να προσθέσουμε μια περιγραφή για τον πίνακα χρησιμοποιώντας το `<caption>`

```
<table>
  <caption>
    Our products
  </caption>
  ...
</table>
```

---

Name	Our products Stock	Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

# HTML Tables

Μπορούμε να χωρίσουμε έναν πίνακα δομικά σε 3 κομμάτια:

- <thead>: Table head
- <tbody>: Table body
- <tfoot>: Table footer

Our products		
Name	Stock	Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23
Subtotal		\$105.19

```
<table>
  <caption>Our products</caption>
  <thead>
    <th>Name</th>
    <th>Stock</th>
    <th>Price</th>
  </thead>
  <tbody>
    ...
  </tbody>
  <tfoot>
    <tr>
      <td>Subtotal</td><td></td>
      <td>$105.19</td>
    </tr>
  </tfoot>
</table>
```

# HTML Tables

Όπως και στο excel μπορούμε και στην HTML να ενώσουμε 2 κελιά χρησιμοποιώντας τα attributes `colspan` και `rowspan`.

`colspan`: πόσες στήλες καταλαμβάνει το κελί

`rowspan`: πόσες γραμμές καταλαμβάνει το κελί



# HTML Tables

```
<table>
  <caption>Our products</caption>
  <thead>
    <th colspan="2">Name & Stock</th>
    <th>Price</th>
  </thead>
  <tbody>
    ...
  </tbody>
  <tfoot>
    <tr>
      <td>Subtotal</td><td></td><td>$105.19</td>
    </tr>
  </tfoot>
</table>
```

	Our products	
	Name & Stock	Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23
Subtotal		\$105.19

# HTML Tables

Εάν θέλουμε κάθε κελί να έχει το δικό του border μπορούμε να δώσουμε στο `table-collapse` την τιμή `separate`.

Χρησιμοποιώντας παράλληλα και το `border-spacing` μπορούμε να καθορίσουμε το κενό μεταξύ των border ώστε να μην είναι το ένα ακριβώς δίπλα με το άλλο.

# HTML Tables border spacing

```
table {  
  border-collapse: separate;  
  border-spacing: 4px;  
}  
table,  
th,  
td {  
  border: 1px solid #cecf55;  
}  
th,  
td {  
  padding: 10px 15px;  
}
```

Name & Stock		Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23
Subtotal		\$105.19

# HTML Tables Rows only border

```
table {  
  border-collapse: collapse;  
}  
th,  
td {  
  border-bottom: 1px solid #cecf5;  
  padding: 10px 15px;  
}  
tfoot tr:last-child td {  
  border-bottom: 0;  
}
```

Name & Stock		Price
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23
Subtotal		\$105.19

# HTML Lists

Οι λίστες στην HTML χρησιμοποιούνται συστηματικά. Υπάρχουν αρκετοί τύποι λιστών :

- Unordered Lists (ul tag)
- Ordered Lists (ol tag)
- Description Lists (dl tag)

# HTML Lists

Μπορούμε να ορίσουμε την αρχή της αρίθμησης με το **start** attribute:

```
<ol start="30">  
  <li>Head north on N Halsted St</li>  
  <li>Turn right on W Diversey Pkwy</li>  
  <li>Turn left on N Orchard St</li>  
</ol>
```

30. Head north on N Halsted St
31. Turn right on W Diversey Pkwy
32. Turn left on N Orchard St

# HTML Lists

Μπορούμε να αντιστρέψουμε την αρίθμηση με το **reversed** attribute:

```
<ol reversed>  
  <li>Head north on N Halsted St</li>  
  <li>Turn right on W Diversey Pkwy</li>  
  <li>Turn left on N Orchard St</li>  
</ol>
```

3. Head north on N Halsted St
2. Turn right on W Diversey Pkwy
1. Turn left on N Orchard St

# HTML Lists

Μπορούμε να ορίσουμε τον αριθμό ενός στοιχείου ορίζοντας το **value** attribute στο **li**:

```
<ol>  
  <li>Head north on N Halsted St</li>  
  <li value="9">Turn right on W Diversey  
Pkwy</li>  
  <li>Turn left on N Orchard St</li>  
</ol>
```

1. Head north on N Halsted St
9. Turn right on W Diversey Pkwy
10. Turn left on N Orchard St



# HTML Lists

```
<dl>
  <dt>design</dt>
  <dd>A plan or drawing produced to show the look and function or workings of a building, garment, or
other object before it is built or made</dd>
  <dd>Purpose, planning, or intention that exists or is thought to exist behind an action, fact, or material
object</dd>
  <dt>work</dt>
  <dd>A person's regular occupation, profession, or trade</dd>
</dl>
```

design

A plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is built or made

Purpose, planning, or intention that exists or is thought to exist behind an action, fact, or material object

work

A person's regular occupation, profession, or trade

# HTML Horizontal Lists

Αρκεί να ορίσουμε το display CSS attribute του list item (`<li>`) ως `inline` ή `inline-block`

```
li {  
  display: inline-block;  
}
```

Orange      Green      Blue

! αυτός ο τρόπος αφαιρεί την εμφάνιση του δείκτη

# HTML Horizontal Menu with lists

```
#header nav {  
  letter-spacing: 0.075em;  
  position: absolute;  
  right: 2.5em;  
  top: 1.5em;  
  text-transform: uppercase;  
}  
#header nav ul li {  
  display: inline-block;  
  margin-left: 1.5em;  
}
```

```
#header nav ul li a {  
  border: solid 1px transparent;  
  color: #FFFFFF;  
  display: inline-block;  
  line-height: 1em;  
  padding: 0.8em 1.2em;  
  text-decoration: none;  
  font-size: 14px;  
}  
#header nav ul li.active > a, #header nav ul li:hover  
> a {  
  background: rgba(188, 202, 206, 0.15);  
}
```

# Questions ?



# Introduction to CSS

# CSS

We will talk about:

- Basics
- Terms
- Cascading
- Specificity
- Combining Selectors
- Common Values

# CSS – What is CSS

**CSS, or Cascading Style Sheets, is a presentation language created to style the appearance of content—using, for example, fonts or colors.**

# CSS – Without CSS

## Account Details

### Personal Details

FIRST NAME

LAST NAME

I AM

BIRTH DATE

BIRTH DATE

### Contact Details

PHONE NUMBER

EMAIL

### ADDRESS

STREET NAME & NUMBER

POST CODE

CITY

COUNTRY

Build up a perfect profile!

Make sure your profile ranks higher than others on employers searches, by answering your personality profiling questions.



# CSS – With CSS

## Account Details

### Personal Details

FIRST NAME

Ioannis

LAST NAME

Imelos

I AM

Male

BIRTH DATE

5

July

1992

### Contact Details

PHONE NUMBER

6932971440

EMAIL

imelos@collegelink.gr

# CSS Selectors

Καθορίζουν σε ποιο σημείο της html εφαρμόζονται τα styles μας.  
Συνήθως στοχεύουν type of element, class ή id.

```
p { ... }
```

```
.this_class { ... }
```

```
#that_id { ... }
```

# CSS Properties

Καθορίζουν τα styles που θα εφαρμοστούν σε κάθε element.

Αναφέρουμε μερικά properties ενδεικτικά απ τα πολλά που υπάρχουν:

- background
- border
- column
- padding
- text-align

# CSS Properties Values

Έχουμε επιλέξει τα elements που θα επηρεάσουμε καθώς και τα χαρακτηριστικά τους που θα επηρεάσουμε.

Με τις τιμές που δίνουμε ορίζουμε με ποιον τρόπο θα επηρεαστούν.

# CSS Properties Values

```
p {
```

```
  color: orange;
```

```
  font-size: 16px;
```

```
}
```

selector

```
p {
```

property

```
  color: orange;
```

```
  font-size: 16px;
```

value

```
}
```

# CSS - Link CSS

Με αυτόν τον τρόπο, “καλούμε” κάποιο αρχείο CSS και δίνουμε styling στην HTML μας!

```
<head>  
    <link rel="stylesheet"  
href="style.css">  
</head>
```

# CSS – C for Cascading

Η CSS είναι **σειριακή επικαλυπτόμενη** γλώσσα εκτός αν το ορίσουμε εμείς διαφορετικά . Για παράδειγμα

```
p {color: blue !important}
```

```
p {color: black}
```

\*Γενικά αποφεύγουμε τη χρήση του important εκτός σπάνιων περιπτώσεων!

# CSS – C for Cascading

```
p {  
    color: orange;  
    font-size: 16px;  
}
```

```
p {  
    color: blue;  
}
```



# CSS - Specificity

Όπως είπαμε τα selectors μπορούν να είναι διάφορων τύπων.

Από type of element (το μικρότερο specificity) σε class μέχρι και id (το μεγαλύτερο specificity)

**type of element : 0-0-1**

**class: 0-1-0**

**id: 1-0-0**

# CSS – Specificity

HTML

```
<p id="food"> ... </p>
```

CSS

```
#food{  
    color: orange;  
}  
p {  
    color: blue;  
}
```

HTML

```
<p id="food"> ...  
</p>
```

CSS

```
#food{  
    color: orange;  
}  
p {  
    color: blue;  
}
```

# CSS – Specificity

HTML

```
<div class="hotdog">  
  <p>... </p>  
  <p>... </p>  
  <p class="mustard">...</p>  
</div>
```

CSS

```
.hotdog p {  
  color: orange;  
}  
.hotdog p.mustard {  
  color: blue;  
}
```

# CSS – Specificity

HTML

```
<div class="hotdog">  
  <p>... </p>  
  <p>... </p>  
  <p class="mustard">...</p>  
</div>
```

CSS

```
.hotdog p {  
  color: orange;  
}  
.hotdog p.mustard {  
  color: blue;  
}
```

# CSS – Specificity

HTML

```
<div class="hotdog">  
  <p>... </p>  
  <p>... </p>  
  <p class="mustard">...</p>  
</div>
```

CSS

```
.hotdog p.mustard {  
  color: blue;  
}  
.hotdog p {  
  color: orange;  
}
```

# CSS – Specificity

HTML

```
<div class="hotdog">  
  <p>... </p>  
  <p>... </p>  
  <p class="mustard">...</p>  
</div>
```

CSS

```
.hotdog p.mustard {  
  color: blue;  
}  
.hotdog p {  
  color: orange;  
}
```

# CSS – Styling Multiple Classes

Αρκετές φορές θέλουμε κάποια styles να τα δώσουμε σε πολλαπλά elements. Για αυτό χρησιμοποιούμε πολλαπλές classes σε ένα element – κάθε class για διαφορετικό specificity και styling.

```
<a class="btn btn-danger">...</a>  
<a class="btn btn-success">...</a>
```

# CSS – Styling Multiple Classes

```
<a class="btn btn-danger">...</a>  
<a class="btn btn-success">...</a>
```

HTML

```
.btn {  
    font-size: 14px;  
}  
.btn-danger{  
    background: red;  
}  
.btn-success{  
    background : green;  
}
```

CSS



# CSS – Common Values

Η πλειοψηφία του CSS που θα γράψετε στην καριέρα σας θα αφορά **χρώματα** και **μεγέθη**.

# CSS – Colors

Τα χρώματα στη CSS ορίζονται με 4 τρόπους:

1. Keywords
2. Εξαδικά
3. RGB
4. HSL: Hue(0-R, 120-G, 240-B), Saturation, Lightness

# CSS – Colors

```
.btn {  
    background: red;  
}  
.btn{  
    background : #ff0000;  
}  
.btn{  
    background : rgb(255, 0, 0);  
}  
.btn{  
    background : hsl(0, 100%, 50%);  
}
```

# CSS – Length

Τα μεγέθη στη CSS ορίζονται με 3 τρόπους:

1. Pixels
2. Percentages
3. em – και πάλι ποσοστά

# CSS – Length

```
.btn {  
    font-size: 14px;  
}  
.btn {  
    width: 50%;  
}  
.btn {  
    font-size: 5em;  
}
```

```
.btn {  
    font-size: 14px;  
}  
.btn-success {  
    font-size: 2em; -> ??px  
}
```

# Positioning With CSS

# CSS Positioning

We will talk about:

- Box Model
- Display
- Unique Positions
- Float
- inline-block

# CSS – Box Model

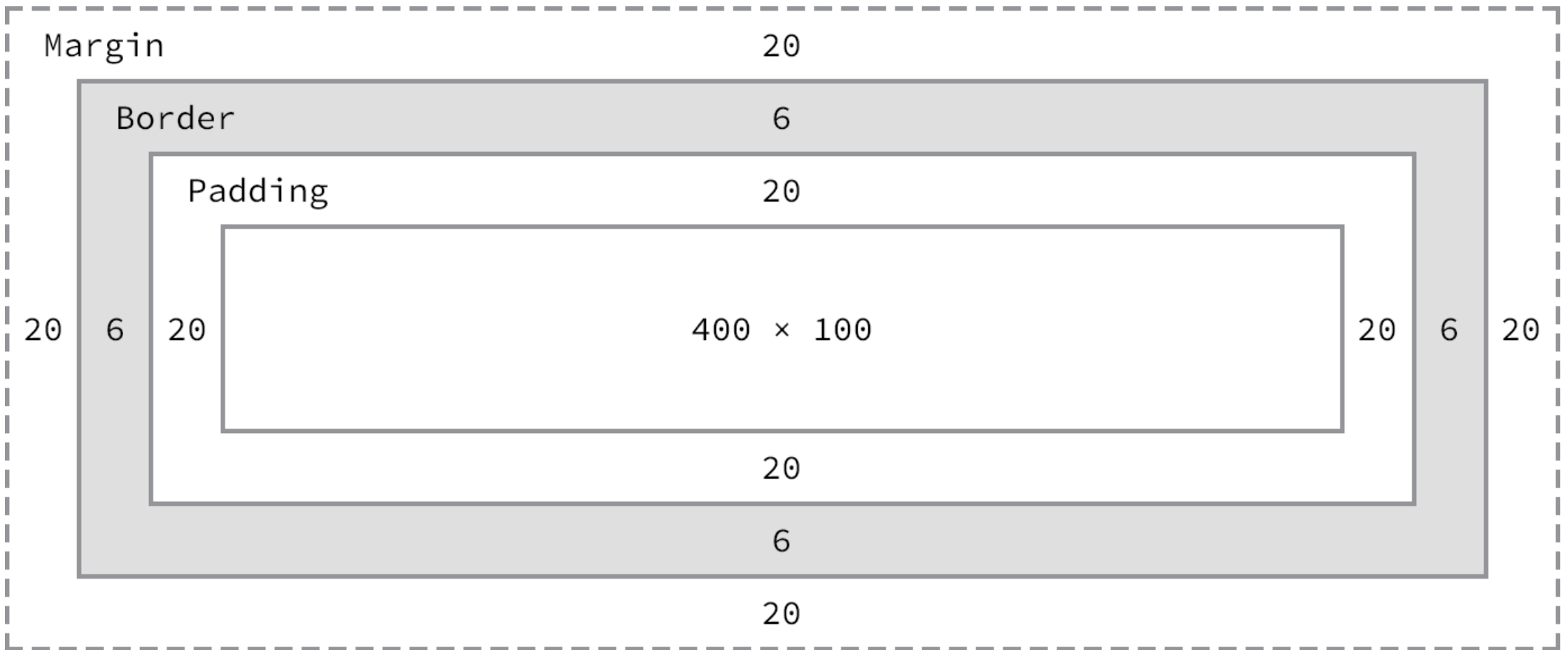
Κάθε html element σε μία σελίδα είναι ένα **παραλληλόγραμμο κουτί**. Όλα είναι παραλληλόγραμμα και εμείς επηρεάζουμε το μέγεθος τους, το που εμφανίζονται και τον τρόπο με τον οποίο εμφανίζονται.

Υπάρχουν πολλά στοιχεία τα οποία επηρεάζουν το μέγεθος του κάθε παραλληλογράμμου

**height, width, border, padding & margin.**



# CSS – Box Model



**Width:**  $20\text{px} + 6\text{px} + 20\text{px} + 400\text{px} + 20\text{px} + 6\text{px} + 20\text{px} = 492\text{px}$

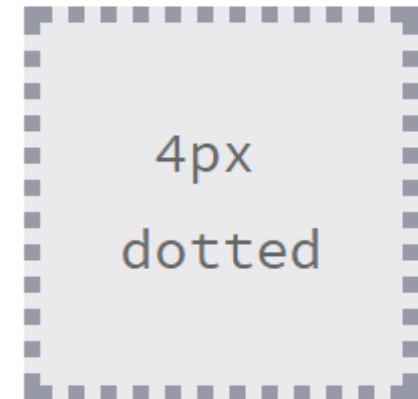
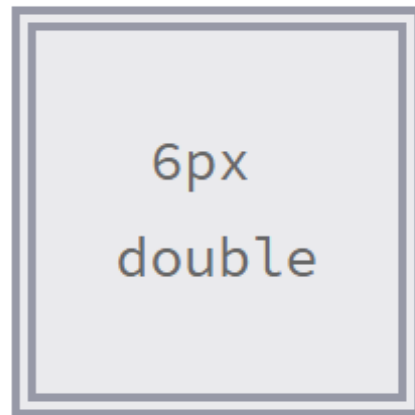
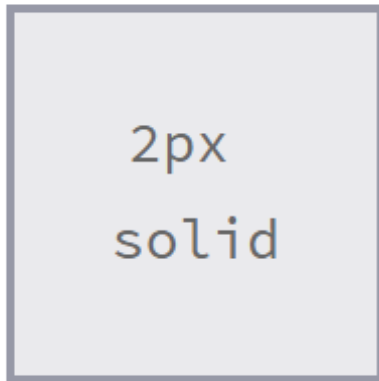
**Height:**  $20\text{px} + 6\text{px} + 20\text{px} + 100\text{px} + 20\text{px} + 6\text{px} + 20\text{px} = 192\text{px}$

# CSS – Box Model

```
div{  
    width: 400px;  
    height: 100px;  
    margin: 20px;  
    border: 6px;  
    padding: 20px;  
}
```

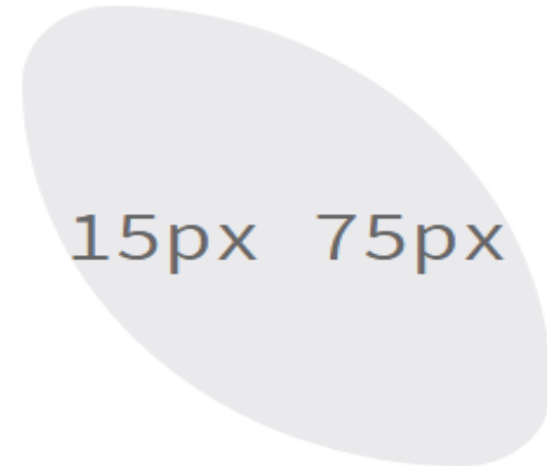
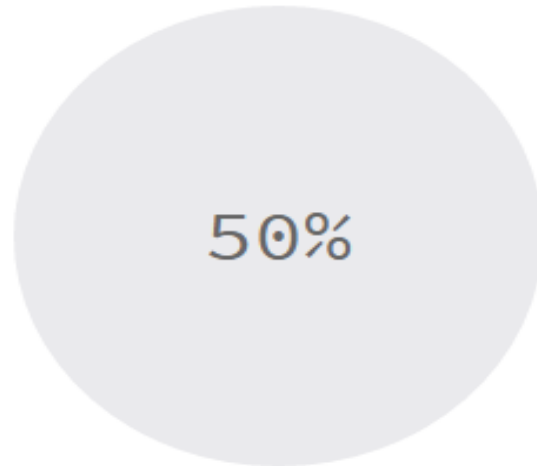
# CSS – Box Model

```
div{  
    border: 1px solid #000;  
}
```



# CSS – Box Model

```
div{  
    border-radius: 50%;  
}
```



# CSS – Box Sizing

```
div{  
    box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
}
```

# CSS – Browser Specific CSS

Υπάρχουν κάποια CSS properties & values που λειτουργούν σε συγκεκριμένους browsers.

**Chrome: -webkit-**

**Mozilla: -moz-**

**Internet Explorer: -ms-**

# CSS – Browser Developer Tools

× Elements Resources Network Sources Timeline Profiles Audits Console

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <!-- Header -->
    <header class="container">...</header>
    <!-- Hero -->
    <section class="container">
      <h2>Dedicated to the Craft of Building Websites</h2>
      <p>...</p>
      <a class="btn btn-alt" href="register.html">Register Now</a>
    </section>
    <!-- Teasers -->
    <section class="container">...</section>
    <!-- Footer -->
    <footer class="container">...</footer>
  </body>
</html>
```

Styles Computed Event Listeners DOM Breakpoints Properties »

☐ Show inherited

margin -

border 1

padding 10

88 x 16

10

1

1 30 30 1

▶ border-bottom-color:  rgb(223, 226, 229);

▶ border-bottom-left-radius: 5px;

▶ border-bottom-right-radius: 5px;

□ ≡ 🔍 html body section.container a.btn.btn-alt ⚙️

# CSS – Display Property

Ένα από τα βασικότερα properties της CSS είναι το display. Βασικές τιμές είναι το **inline**, **block**, **inline-block** & **none**



# CSS - Positioning

Το καλύτερο χαρακτηριστικό της CSS είναι ότι μας δίνει τη δυνατότητα να τοποθετούμε τα διάφορα elements με κάθε πιθανό και απίθανο τρόπο που μπορούμε να φανταστούμε!

Υπάρχουν 2 βασικοί τρόποι με τους οποίους κάνουμε το positioning στις σελίδες μας:

**float & inline-block**

# CSS – Float Positioning

Ο βασικός τρόπος για να χρησιμοποιήσουμε float είναι για ορίσουμε το που θα “κάτσει” κάποιο element.

# CSS – Float Positioning

```
div{  
    float: left;  
    width: 70%;  
}  
sidebar{  
    float: right;  
    width: 30%;  
}
```

<div>  
float: left;

<sidebar>  
float: right;

# CSS – Float Positioning

```
div{  
    float: left;  
    width: 30%;  
}
```

```
<div>  
float: left;
```

```
<div>  
float: left;
```

```
<div>  
float: left;
```

# CSS – Inline block positioning

To **display: inline-block;** Είναι μία εντολή που μας αλλάζει τη ζωή.  
Ξαφνικά τα elements μπορούν να βρίσκονται μαζί στην ίδια σειρά.

# CSS – Inline block positioning

```
div{  
    display: inline-block;  
    width: 30%;  
}
```

```
<div>  
display:  
inline-block;
```

```
<div>  
display:  
inline-block;
```

```
<div>  
display:  
inline-block;
```

# CSS – Unique positioning

Καμιά φορά χρειάζεται να αλλάξουμε κατά πολύ τη δομή εμφάνισης των elements σε μία σελίδα.

Για να το κάνουμε αυτό υπάρχει μία εντολή cheat:

**position: static to relative / absolute**

**+ offset positions: top, right, bottom & left**

# CSS – Unique positioning

```
.offset{  
    position: relative;  
    left: 20px;  
    top: 20px;  
}
```





# CSS – Unique Positioning

```
div{  
    position: relative;  
    width: 100%;  
}  
  
.offset{  
    position: absolute;  
    right: 20px;  
    top: 20px;  
}
```

# CSS – Unique Positioning

```
<section>  
position: relative;  
width: 100%;
```

```
<div  
class="offset">  
position: absolute;  
right: 20px;  
top: 20px;
```

## Συνοψίζοντας :

- relative = θέση σχετικά με την κανονική θέση του object
- absolute = θέση σχετικά με το κοντινότερο relative πατέρα

ένα object με relative και top 5px θα μετακινηθεί 5 pixel από την θέση που ήταν,

ένα object με absolute και top 5px, το top του θα πάει 5px πιο κάτω από το top του πρώτου κοντινότερου relative πατέρα. αν δεν υπάρχει θα πάει 5px από την οθονή

# CSS – Bonus

Αν θέλετε να στυλάρετε ένα element στο κέντρο της σελίδας χωρίς `text-align:center`; Στο parent element χρησιμοποιούμε:

**`margin-left: auto;`**

**`Margin-right: auto;`**

# CSS – Bonus

Αν θέλετε να στυλάρετε ένα element στο κέντρο της σελίδας χωρίς `text-align:center`; Στο parent element χρησιμοποιούμε:

**`margin-left: auto;`**

**`Margin-right: auto;`**

# Bootstrap



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

# OVERVIEW

# Requirements

HTML5 Doctype

```
<!DOCTYPE html>
```



# Requirements

## Mobile First

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
```

# Requirements

Normalize.css

<https://github.com/necolas/normalize.css/blob/master/normalize.css>

# Containers

container



container-fluid



# Containers

container

```
<div class="container">  
  
...  
</div>
```

container-fluid

```
<div class="container-fluid">  
  
...  
</div>
```

# Grid System

Container > Rows > Columns > Content



# Grid System

- Extra small devices
  - Less than 768px
- Small devices
  - More than 768px
- Medium devices
  - 992px and up
- Large devices
  - 1200px and up

# Grid System

```
<div class="row">  
  <div class="col-md-8">.col-md-8</div>  
  <div class="col-md-4">.col-md-4</div>  
</div>  
<div class="row">  
  <div class="col-md-4">.col-md-4</div>  
  <div class="col-md-4">.col-md-4</div>  
  <div class="col-md-4">.col-md-4</div>  
</div>  
<div class="row">  
  <div class="col-md-6">.col-md-6</div>  
  <div class="col-md-6">.col-md-6</div>  
</div>
```

# Grid System

.col-md-8

.col-md-4

.col-md-4

.col-md-4

.col-md-4

.col-md-6

.col-md-6



# Grid System - Mobile

## desktop

.col-md-8		.col-md-4	
.col-md-4		.col-md-4	
.col-md-6		.col-md-6	

## mobile

.col-xs-12 .col-md-8	
.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4
.col-xs-6 .col-md-4	
.col-xs-6	.col-xs-6

# Grid System - Offset

.col-md-4

.col-md-4 .col-md-offset-4

.col-md-3 .col-md-offset-3

.col-md-3 .col-md-offset-3

.col-md-6 .col-md-offset-3

```
<div class="row">  
  <div class="col-md-4">.col-md-4</div>  
  <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>  
</div>
```

## Grid System - Nesting columns

```
<div class="row">  
  <div class="col-sm-9">  
    Level 1: .col-sm-9  
    <div class="row">  
      <div class="col-xs-8 col-sm-6">Level 2: .col-xs-8 .col-sm-6</div>  
      <div class="col-xs-4 col-sm-6">Level 2: .col-xs-4 .col-sm-6</div>  
    </div>  
  </div>  
</div>
```

# Grid System - Nesting columns

Level 1: .col-sm-9

Level 2: .col-xs-8 .col-sm-6

Level 2: .col-xs-4 .col-sm-6

## Grid System - Ordering columns

```
<div class="row">  
  <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>  
  <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>  
</div>
```

.col-md-3 .col-md-pull-9

.col-md-9 .col-md-push-3

**TYPOGRAPHY**

# Font

## Helvetica Neue

**h1. Bootstrap heading**

Semibold 36px

**h2. Bootstrap heading**

Semibold 30px

**h3. Bootstrap heading**

Semibold 24px

**h4. Bootstrap heading**

Semibold 18px

**h5. Bootstrap heading**

Semibold 14px

**h6. Bootstrap heading**

Semibold 12px

## Paragraph

<p>

*Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.*

</p>

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.



## Lead Paragraph

```
<p class="lead">
```

*Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.*

```
</p>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

## Mark tag

<p>

*Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. <mark>Duis mollis, est non commodo luctus.</mark>*

</p>

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

## Deleted text - Strikethrough

*<p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. **<del>Duis mollis, est non commodo luctus.</del></p>***

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. ~~Duis mollis, est non commodo luctus.~~

*<p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. **<s>Duis mollis, est non commodo luctus.</s></p>***

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. ~~Duis mollis, est non commodo luctus.~~

## Underline

*<p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. <u>Duis mollis, est non commodo luctus.</u></p>*

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

## Bold - Italic (emphasized)

*<p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.*

***<strong>Duis mollis, est non commodo luctus.</strong></p>***

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. **Duis mollis, est non commodo luctus.**

*<p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. **<em>Duis mollis, est non commodo luctus.</em></p>***

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. *Duis mollis, est non commodo luctus.*

# Text Alignment

`<p class="text-left">Left aligned text.</p>` `<p class="text-justify">Justified text.</p>`

`<p class="text-center">Center aligned text.</p>` `<p class="text-right">Right aligned text.</p>`

`<p class="text-nowrap">No wrap text.</p>`

Left aligned text.

Center aligned text.

Right aligned text.

Justified text.

No wrap text.

## Change case

```
<p class="text-lowercase">Lowercased text.</p>
```

```
<p class="text-uppercase">Uppercased text.</p>
```

```
<p class="text-capitalize">Capitalized text.</p>
```

lowercased text.

UPPERCASED TEXT.

Capitalized Text.

# Blockquote

```
<blockquote>
```

```
  <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non  
commodo luctus.</p>
```

```
  <footer>Someone famous in <cite title="Source Title">Source Title</cite></footer>
```

```
</blockquote>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

— Someone famous in *Source Title*



## Blockquote reverse

```
<blockquote class="blockquote-reverse">
```

```
  <p>Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non  
commodo luctus.</p>
```

```
  <footer>Someone famous in <cite title="Source Title">Source Title</cite></footer>
```

```
</blockquote>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est non commodo luctus.

Someone famous in *Source Title* —

## Lists - Inline

```
<ul class="list-inline">  
  <li>Lorem ipsum dolor sit amet</li>  
  <li>Consectetur adipiscing elit</li>  
  <li>Integer molestie lorem at massa</li>  
  <li>Facilisis in pretium nisl aliquet</li>  
</ul>
```

Lorem ipsum dolor sit amet   Consectetur adipiscing elit   Integer molestie lorem at massa   Facilisis in pretium nisl aliquet

**TABLES**

# Table

```
<table>
<tr>
  <td>Product 1</td>
  <td>In Stock</td>
  <td>$30.02</td>
</tr>
<tr>
  <td>Product 2</td>
  <td>In Stock</td>
  <td>$52.94</td>
</tr>
<tr>
  <td>Product 3</td>
  <td>Out of Stock</td>
  <td>$22.23</td>
</tr>
</table>
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

## Table - bootstrap

```
<table class="table">
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

# Table - Striped

```
<table class="table table-striped">
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```

```
tbody tr:nth-child(odd) {
  background: #f0f0f2;
}
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

## Table - Bordered

```
<table class="table table-bordered">
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23

## Table - Hover

```
<table class="table table-hover">
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```


Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23



## Table - Condensed

```
<table class="table table-condensed">
  <tr>
    <td>Product 1</td>
    <td>In Stock</td>
    <td>$30.02</td>
  </tr>
  <tr>
    <td>Product 2</td>
    <td>In Stock</td>
    <td>$52.94</td>
  </tr>
  <tr>
    <td>Product 3</td>
    <td>Out of Stock</td>
    <td>$22.23</td>
  </tr>
</table>
```

Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23
Product 1	In Stock	\$30.02
Product 2	In Stock	\$52.94
Product 3	Out of Stock	\$22.23



## Table - Contextual classes

```
<table class="table">
  <tr class="active">
    <td>Product 1</td><td>Active</td>
  </tr>
  <tr class="success">
    <td>Product 2</td><td>Success</td>
  </tr>
  <tr class="info">
    <td>Product 3</td><td>Info</td>
  </tr>
  <tr class="warning">
    <td>Product 4</td><td>Warning</td>
  </tr>
  <tr class="danger">
    <td>Product 5</td><td>Danger</td>
  </tr>
</table>
```

Product 1	Active
Product 2	Success
Product 3	Info
Product 4	Warning
Product 5	Danger











# Table - Responsive

```
<div class="table-responsive">
  <table class="table">
    <tr>
      <td>Product 1</td>
      <td>In Stock</td>
      <td>Brand</td>
      <td>Category</td>
      <td>Supplier</td>
      <td>Store</td>
      <td>Shelf</td>
      <td>Data</td>
    </tr>
  </table>
</div>
```

Product 1	In Stock	Brand	Category	Supplier	Store
-----------	----------	-------	----------	----------	-------

# OTHER ELEMENTS

# Glyphicons

							
glyphicon glyphicon-asterisk	glyphicon glyphicon-plus	glyphicon glyphicon-euro	glyphicon glyphicon-eur	glyphicon glyphicon-minus	glyphicon glyphicon-cloud	glyphicon glyphicon-envelope	glyphicon glyphicon-pencil
							
glyphicon glyphicon-glass	glyphicon glyphicon-music	glyphicon glyphicon-search	glyphicon glyphicon-heart	glyphicon glyphicon-star	glyphicon glyphicon-star-empty	glyphicon glyphicon-user	glyphicon glyphicon-film
							
glyphicon glyphicon-th-large	glyphicon glyphicon-th	glyphicon glyphicon-th-list	glyphicon glyphicon-ok	glyphicon glyphicon-remove	glyphicon glyphicon-zoom-in	glyphicon glyphicon-zoom-out	glyphicon glyphicon-off
							
glyphicon glyphicon-signal	glyphicon glyphicon-cog	glyphicon glyphicon-trash	glyphicon glyphicon-home	glyphicon glyphicon-file	glyphicon glyphicon-time	glyphicon glyphicon-road	glyphicon glyphicon-download-alt

## Glyphicons - Example

```
<i class="glyphicon glyphicon-search" aria-hidden="true"></i>  
<span>search</span>
```



# Breadcrumbs

```
<ol class="breadcrumb">  
  <li><a href="#">Home</a></li>  
  <li><a href="#">Library</a></li>  
  <li class="active">Data</li>  
</ol>
```

[Home](#) / [Library](#) / Data

# Pagination

```
<nav aria-label="Page navigation">
  <ul class="pagination">
    <li><a href="#" aria-label="Previous"><span aria-
hidden="true">&laquo;</span></a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#" aria-label="Next"><span aria-
hidden="true">&raquo;</span></a></li>
  </ul>
</nav>
```





# Labels

```
<span class="label label-default">Default</span>  
<span class="label label-primary">Primary</span>  
<span class="label label-success">Success</span>  
<span class="label label-info">Info</span>  
<span class="label label-warning">Warning</span>  
<span class="label label-danger">Danger</span>
```

**Default**

**Primary**

**Success**

**Info**

**Warning**

**Danger**

# Thumbnails

```
<div class="row">
  <div class="col-sm-6 col-md-4">
    <div class="thumbnail">
      
      <div class="caption">
        <h3>Thumbnail label</h3>
        <p>...</p>
        <p><a href="#" class="btn btn-primary"
role="button">Button</a> <a href="#" class="btn btn-
default" role="button">Button</a></p>
      </div>
    </div>
  </div>
</div>
```



# Alerts

```
<div class="alert alert-success" role="alert">...</div>  
<div class="alert alert-info" role="alert">...</div>  
<div class="alert alert-warning" role="alert">...</div>  
<div class="alert alert-danger" role="alert">...</div>
```

**Well done!** You successfully read this important alert message.

**Heads up!** This alert needs your attention, but it's not super important.

**Warning!** Better check yourself, you're not looking too good.

**Oh snap!** Change a few things up and try submitting again.

# Panels

```
<div class="panel panel-default">  
  <div class="panel-heading">  
    <h3 class="panel-title">Panel title</h3>  
  </div>  
  <div class="panel-body">  
    Panel content  
  </div>  
</div>
```

Panel title

Panel content

# Wells

```
<div class="well well-lg">  
  <p>  
    Look, I'm in a large well!  
  </p>  
</div>
```

Look, I'm in a large well!

# CSS – Typography

**We will talk about :**

- **Font Properties**
- **Text Properties**
- **Web Fonts**

# CSS TYPOGRAPHY

# CSS – Typography

Ένα από τα πρώτα πράγματα που κάνουμε όταν σχεδιάζουμε ένα νέο website είναι να αποφασίσουμε τη γραμματοσειρά και το χρώμα της. Μπορεί να σας φαίνεται λεπτομέρεια, αλλά σε όρους design και marketing είναι πάρα πολύ σημαντική απόφαση!



# CSS – Text properties

Μπορούμε να επηρεάσουμε απολύτως τον τρόπο εμφάνισης του κειμένου στη σελίδα μας.

```
.text{  
    color: #000;  
    font-family: "Zona Pro", Helvetica, Arial, sans-serif;  
    font-size: 20px;  
    font-style: italic;  
}
```

# CSS – Text properties

```
.text{  
    font-weight: bold;  
    font-weight: 400;  
    line-height: 22px;  
    height: 22px;  
}
```

```
.text{  
    text-align: center;  
    text-indent: 20px;  
    text-shadow: 5px 10px  
#000;  
}  
.text:hover{  
    text-decoration: underline;  
}
```

# CSS – Text properties

```
.text{  
    font-weight: bold;  
    font-weight: 400;  
    line-height: 22px;  
    height: 22px;  
    text-transform: uppercase;  
    letter-spacing: 2px;  
    word-spacing: 0.5em;  
}
```

```
.text{  
    text-align: center;  
    text-indent: 20px;  
    text-shadow: 5px 10px  
#000;  
}  
.text:hover{  
    text-decoration: underline;  
}
```

# CSS – Typography

Θα συναντήσετε αρκετές φορές τις λέξεις typeface & font.

Αναφέρονται σε κάτι διαφορετικό.

Typeface είναι αυτό που βλέπουμε η γραμματοσειρά – ενώ font είναι το αρχείο που περιέχει το typeface.

# CSS – Typography

Στο web χρησιμοποιούμε **sans serif** γραμματοσειρές όπως Arial, Helvetica, Calibri ή Roboto

Δεν θέλουμε **serif** – είναι δυσανάγνωστες σε οθόνη

# HTML

## Tips & Best Practices

# HTML Best Practices

Η HTML συχνά συγχωρεί λάθη και απροσεξίες και η σελίδα γίνεται render σωστά χωρίς αυτό να σημαίνει ότι κώδικας μας είναι σωστός.

Είναι σημαντικό να ακολουθούμε τις κοινές πρακτικές και να προσέχουμε πως ανοίγουμε και κλείνουμε τα tags.

# HTML Common Errors

## Bad code

```
<p id="intro">New items on the menu today include <strong>caramel apple cider and  
breakfast crepes</p>.</strong>  
<p id="intro">The caramel apple cider is delicious.
```



# HTML Tips

## Good code

```
<p class="intro">New items on the menu today include <strong>caramel apple cider and  
breakfast crepes</strong>.</p>  
<p class="intro">The caramel apple cider is delicious.</p>
```

# Use Semantic Elements

## Bad Code

```
<span class="heading"><strong>Welcome Back</span></strong>  
<br><br>  
It has been a while. What have you been up to lately?  
<br><br>
```

# Use Semantic Elements

## Good Code

```
<h1>Welcome Back</h1>  
<p>It has been a while. What have you been up to lately?</p>
```

# Proper Document Structure

## Bad Code

```
<html>  
  <h1>Hello World</h1>  
  <p>This is a web page.</p>  
</html>
```

Η σελίδα θα εμφανιστεί χωρίς να χρησιμοποιήσουμε τα `<!DOCTYPE html>`, `<html>`, `<head>`, και `<body>` στοιχεία.

# Proper Document Structure

## Good Code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```

Όμως ο μόνος τρόπος να είμαστε σίγουροι ότι θα εμφανίζεται σωστά σε κάθε περίπτωση είναι να προσθέσουμε όλα αυτά τα στοιχεία.

# Organized Syntax

- Χρησιμοποιούμε πάντα πεζά γράμματα
- Στοιχίζουμε τα εμφωλευμένα στοιχεία πιο μέσα σε σχέση με τα γονικά

# Practical ID & Class Values

## Bad Code

```
<p class="red">Error! Please try again.</p>
```

Οι τιμές πρέπει να περιγράφουν το περιεχόμενο του element και όχι το style του.

## Good Code

```
<p class="alert">Error! Please try again.</p>
```

# Image alt text

Πάντα πρέπει να προσθέτουμε alt text στις εικόνες.

Είναι η κειμενική περιγραφή της εικόνας που χρησιμοποιείται:

- Εάν δεν μπορεί να εμφανιστεί η εικόνα
- Από screen readers για ΑΜΕΑ
- SEO reasons



# Image alt text

Πάντα πρέπει να προσθέτουμε alt text στις εικόνες.

Είναι η κειμενική περιγραφή της εικόνας που χρησιμοποιείται:

- Εάν δεν μπορεί να εμφανιστεί η εικόνα
- Από screen readers για ΑΜΕΑ
- SEO reasons

# Don't use style in your HTML

## Bad Code

```
<p style="color: #393; font-size: 24px;">Thank you!</p>
```

## Good Code

```
<p class="alert-success">Thank you!</p>
```

```
//css
.alert-success {
  color: #393p;
  font-size: 24px;
}
```

# Don't use style in your HTML

## Better Code

```
<p class="alert alert-success">Thank you!</p>
```

```
//css  
.alert {  
  font-size: 24px;  
}  
.alert-success {  
  color: #393p;  
}
```

# Don't use divs for everything

## Bad Code

```
<div class="container">  
  <div class="article">  
    <div class="headline">Headlines Across the World</div>  
  </div>  
</div>
```

## Good Code

```
<div class="container">  
  <article>  
    <h1>Headlines Across the World</h1>  
  </article>  
</div>
```

# Use Comments in CSS

## Bad Code

```
header { ... }  
article { ... }  
.btn { ... }
```

## Good Code

```
/* Primary header */  
header { ... }  
  
/* Featured article */  
article { ... }  
  
/* Buttons */  
.btn { ... }
```

# Use Lines & Spaces in CSS

## Bad Code

```
a,.btn{background:#000;color:#fff;font-size:20px;padding:4px;}
```

## Good Code

```
a,.btn {  
  background: #000;  
  color: #fff;  
  font-size: 20px;  
  padding: 4px;  
}
```

# Use shorthand properties

## Bad Code

```
div {  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 5px;  
  padding-left: 10px;  
}
```

## Good Code

```
div {  
  padding: 5px 10px;  
}
```

# Questions ?





# Thank you

[hello@collegelink.gr](mailto:hello@collegelink.gr)