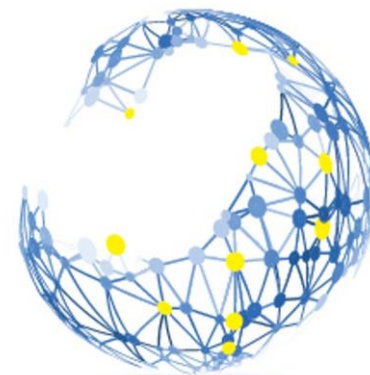


# השתלמות מורים

## תכנות בסביבת האינטרנט

גלעד מרקמן

קריית החינוך פארק המדע,  
נס ציונה



קריית החינוך  
השש שנתית  
פארק המדע  
עתיד של חדשנות



# הקדמה



- עורך דין ובעלים של משרד עורכי הדין מרקמן את טומשין
- בוגר במשפטים בהצטיינות האוניברסיטה העברית, ירושלים
- מוסמך במשפטים אוניברסיטת תל אביב.
- לומד לתואר בוגר במדעי המחשב, האוניברסיטה הפתוחה.
- מורה למדעי המחשב בקריית החינוך "פארק המדע", נס ציונה.
- בעל אתר האינטרנט "תכנות באינטרנט: גלעד מרקמן"

- נייד: 052-2653732; דוא"ל:
- [gilad.markman@parkHamadanz.org.il](mailto:gilad.markman@parkHamadanz.org.il);
- [gilad@markman.co.il](mailto:gilad@markman.co.il)

מס'	תאריך	נושא
1	22/02/2022 ג'	הקדמה + CSS + HTML
2	1/3/2022 ג'	אסינכרוני / מטלה - הכנת אתר הכולל דף ראשי + שני דפי תוכן + דף כניסה + דף הרשמה.
2	8/3/2022 ג'	תכנות צד לקוח JavaScript
3	10/03/2022 ה'	אסינכרוני / מטלה - כתיבת קוד צד הלקוח לאימות טופס כניסה וטופס הרשמה
4	15/03/2022 ג'	תכנות צד שרת ASP.Net בשפת C#
5	23/03/2022 ד'	המשך תכנות צד שרת ASP.Net
6	29/03/2022 ג'	אסינכרוני / מטלה - המרת האתר לדפי aspx; כתיבת קוד לכניסה לאתר (Login) על בסיס שם משתמש השמור במערכת + אבטחת הדפים.
7	6/04/2022 ד'	שפת שאילתות SQL. תכנות בסיסי נתונים בצד השרת בשפת C# - ADO.Net.
8	12/04/2022 ג'	המשך תכנות בסיסי נתונים בצד השרת בשפת C# - ADO.Net
9	28/04/2022 ה'	המשך תכנות בסיסי נתונים בצד השרת בשפת C# - ADO.Net
10	10/05/2022 ג'	מטלה מסכמת / בנוכחות המרצה - סיום האתר, כתיבת קוד לכניסה לאתר (Login) והוספת משתמש חדש (Register)

- הספר מבוא לתכנות בסביבת האינטרנט הוצאת כותר
- אתר האינטרנט w3schools
- אתר "תכנות באינטרנט: גלעד מרקמן"
- קורס קמפוס (ללא עלות): "מרושתים - איך עובד האינטרנט"
- מבוא לתכנות בסביבת האינטרנט הרצאות מצולמות ליחידת הלימוד "תכנות בסביבת האינטרנט"
- חומרים בתיקיה השיתופית של קורס זה.

Visual Studio - תכנות באינטרנט

webprogramming.azurewebsites.net/Pages/HTML/Lesson1.aspx

Classroom Gmail Google Calendar מפות תרגום מרקמן - טומשין האוניברסיטה הפתוחה Smart School Mt-Calendar V4 tab... תכנות באינטרנט | ג... Google Dr... - תשפ"א Reading list

GM Wellcome Visitor

Main HTML & CSS JAVASCRIPT ASP.Net SQL ADO Login Register


HTML & CSS

- Intro
- Basic
- CSS Part I
- CSS Part II
- Tables
- Forms
- Lists and Bars
- HTML to ASP.net web Pages

Home Next >

## HTML & CSS Introduction

בשיעור זה נכיר את שפת HTML - Hyper Text Markup Language, השפה ליצירת דפי אינטרנט. אנו נכתוב את דף האינטרנט הראשון שלנו - Hello World, ונלמד להשתמש בסביבת העבודה של Microsoft Visual Studio.



HTML1 intro


תכנות באינטרנט

גלעד מרקמן  
קריית החינוך פארק המדע,  
נס ציונה

Watch later Share

קריית החינוך  
השש שנתית  
פארק המדע  
עמנו של חנוש

GM  
Internet Programming  
Courses



1 / 20 46%

תכנות באינטרנט

גלעד מרקמן  
קריית החינוך פארק המדע,  
נס ציונה

קריית החינוך  
השש שנתית  
פארק המדע  
עמנו של חנוש

GM  
Internet Programming  
Courses

לינק לוידיאו

לינק ל PDF

Waiting for www.google-analytics.com...

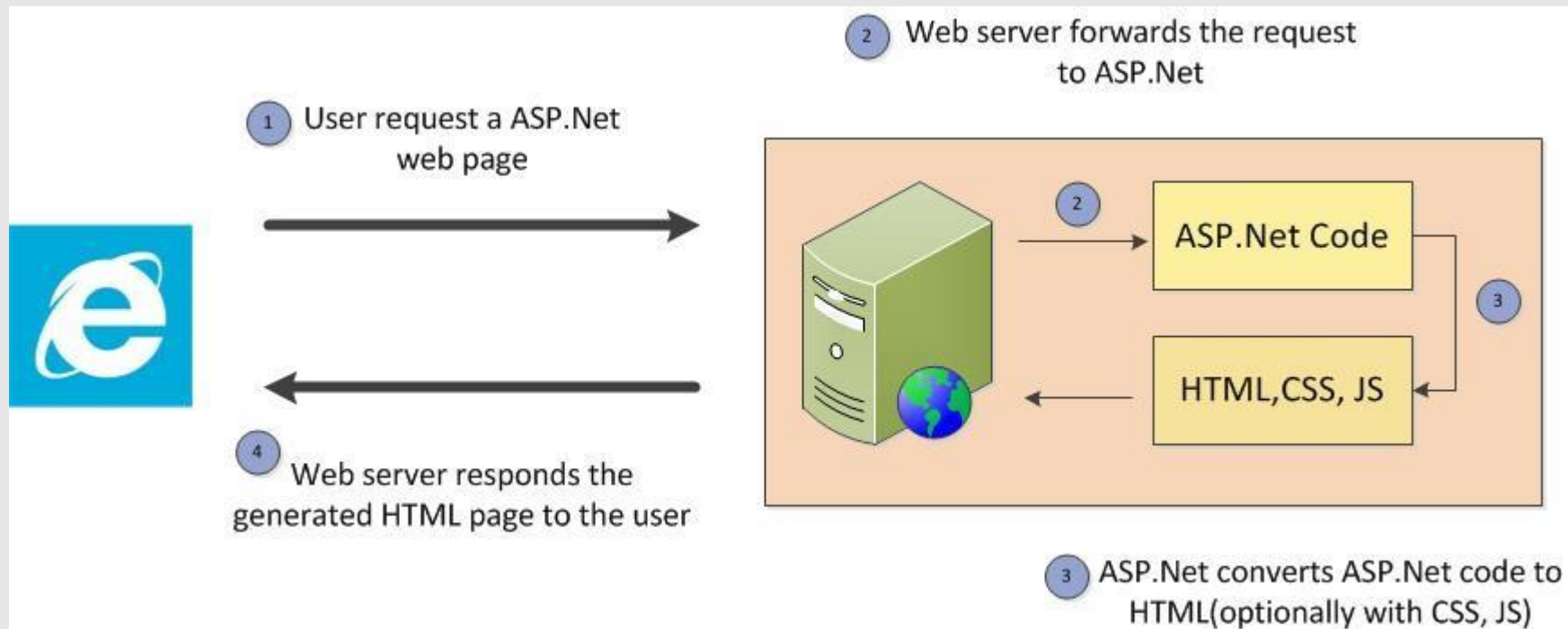
20 22:26 11/10/2021 עבר

...Status - G

Lesson1

הקלד כאן כדי לחפש

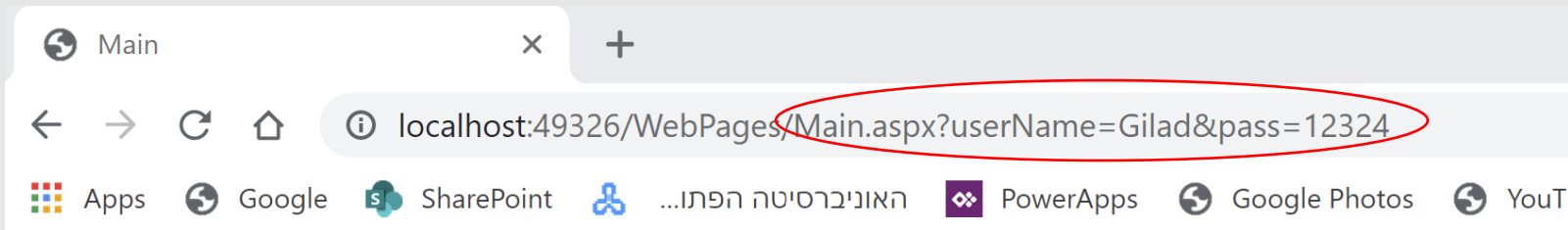
- במודל שרת לקוח הדפדפן של הלקוח שולח בקשה (request) לשרת לקבלת דף אינטרנט. השרת מקבל את הבקשה, יוצר את הדף, ומחזיר אותו ללקוח.



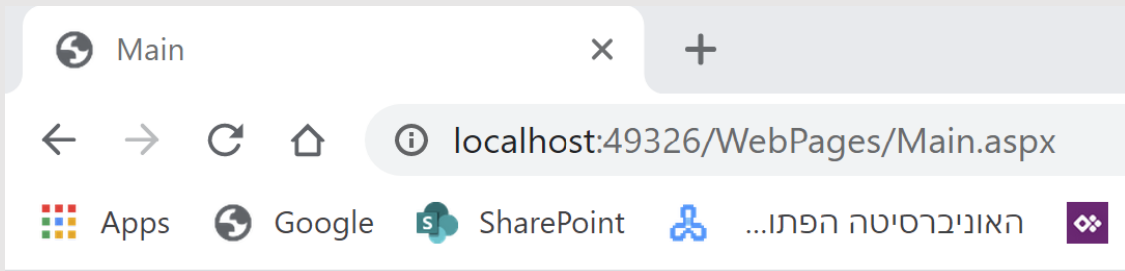
- הפרוטוקול שבו מבוצעת התקשורת בין הלקוח לשרת אינטרנט הינו פרוטוקול HTTP או HTTPS. לכן, הפניה של הלקוח נקראת Http Request, ואילו התגובה של השרת נקראת Http Response.
- קיימים פרוטוקולים נוספים לתקשורת באינטרנט כדוגמת ftp (העברת קבצים) ואחרים.
- כתובת ה url בעת פניה לפי פרוטוקול Http לדוגמה תתחיל ב- http://.
- הפניה של הלקוח Request יכולה לכלול גם מידע שנשלח מן הלקוח אל השרת (כמו שם משתמש וסיסמה).



- לפרוטוקול HTTP קיימות שתי שיטות: get ו- post למשלוח נתונים לשרת:
- השיטה get שולחת את הנתונים כחלק מכתובת ה url. הנתונים גלויים למשתמש.



- השיטה post שולחת את הנתונים בצורה נסתרת.



- נחזור לזה כאשר נדבר על Forms ב HTML

## ASP – Active Server Pages

- ASP.Net היא טכנולוגיית צד שרת שפותחה על ידי חברת מיקרוסופט המאפשרת להציג דפי תוכן דינמיים הנבנים על ידי השרת ונשלחים אל המשתמש.
- שרת ASP.Net מקבל פניה מן הלקוח, יוצר דף HTML, ושולח אותו חזרה אל הלקוח.
- הדף המחוזר מוצג על ידי הדפדפן בצד הלקוח.

- במודל שרת לקוח אנו מחלקים את הקוד "הרץ" בתהליך זה לשני סוגים:
  - קוד צד לקוח.
  - קוד צד שרת.
- קוד צד לקוח – הם תוכניות שרצות על המחשב של הלקוח. הכוונה בדרך כלל לתסריטי גאווה סקריפט אשר נכללים בדף HTML ומופעלים על ידי הדפדפן של הלקוח.
- קוד צד שרת – הוא קוד שרץ על השרת "מאחורי הקלעים". קוד צד שרת בטכנולוגיית ASP.Net יכול להכתב בשפת C# או Visual Basic.
- קוד צד השרת מטפל בדרך כלל בבסיסי הנתונים באמצעות פניה אל שרת ה-DB.

- אנחנו נשתמש בתוכנת Visual Studio Community 2019/2022 של מיקרוסופט.
- זוהי תוכנה חינמית שניתן [להורידה מהאינטרנט](#) ולהתקין בכל מחשב.
- סרטון הדרכה להתקנת סביבת העבודה באתר "[תכנות באינטרנט](#)".
- סביבת העבודה מאפשר לנו לכתוב את הפרויקט בטכנולוגיה של חברת מיקרוסופט - ASP.Net.
- לסביבת העבודה יש גם ממשק לעבודה עם בסיסי נתונים בשפת SQL.

- בסביבת העבודה של Visual Studio 2022 יש להוסיף ידנית את האפשרות ליצור פרויקטים של .NET Framework.
- יש להכנס ל Visual Studio Installer (אפליקציה שמותקנת ביחד עם התקנת סביבת העבודה). ללחוץ על עדכון ולהוסיף:
- .NET Framework project and item templates

Modifying — Visual Studio Community 2022 — 17.0.0

Workloads Individual components Language packs Installation locations

Web & Cloud (4)

- ASP.NET and web development** 
  - Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- Python development 
  - Editing, debugging, interactive development and source control for Python.
- Azure development 
  - Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework...
- Node.js development 
  - Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- Mobile development with .NET 
  - Build cross-platform applications for iOS, Android or Windows using Xamarin.
- .NET desktop development** 
  - Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- Desktop development with C++ 
  - Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- Universal Windows Platform development 
  - Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Installation details

- ASP.NET and web development**
  - Included
    - .NET Framework 4.7.2 development tools
    - ASP.NET and web development prerequisi...
  - Optional
    - .NET Framework 4.8 development tools
    - Cloud tools for web development
    - .NET profiling tools
    - Entity Framework 6 tools
    - Live Share
    - .NET Debugging with WSL
    - IntelliCode
    - .NET Framework project and item templates
    - Windows Communication Foundation
    - .NET Framework 4.6.2-4.7.1 development t...
    - Additional project templates (previous ver...
    - .NET WebAssembly build tools
- .NET desktop development
- Data storage and processing

Location  
C:\Program Files\Microsoft Visual Studio\2022\Community

Total space required 0 B

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Install while downloading

# HTML

- HTML - Hyper Text Markup Language
- שפת HTML היא השפה הבסיסית לבניית דפי אינטרנט.
- שפת HTML אינה שפת תכנות אלא היא שפה שנועדה לבנות את התוכן של דפי אינטרנט.
- דף HTML אינו רץ עצמאית על המחשב אלא הוא מופעל באמצעות דפדפן (Chrom, Edge, iExplorer, Safari, etc.)
- עיצוב התוכן שנבנה על ידי פקודות HTML נעשה באמצעות פקודות CSS אותן נלמד בהמשך.
- ניתן ליצור דף HTML בכל עורך טקסט, ואין צורך בעורך מיוחד.



# דף ראשון – Hello World

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello</title>
  </head>

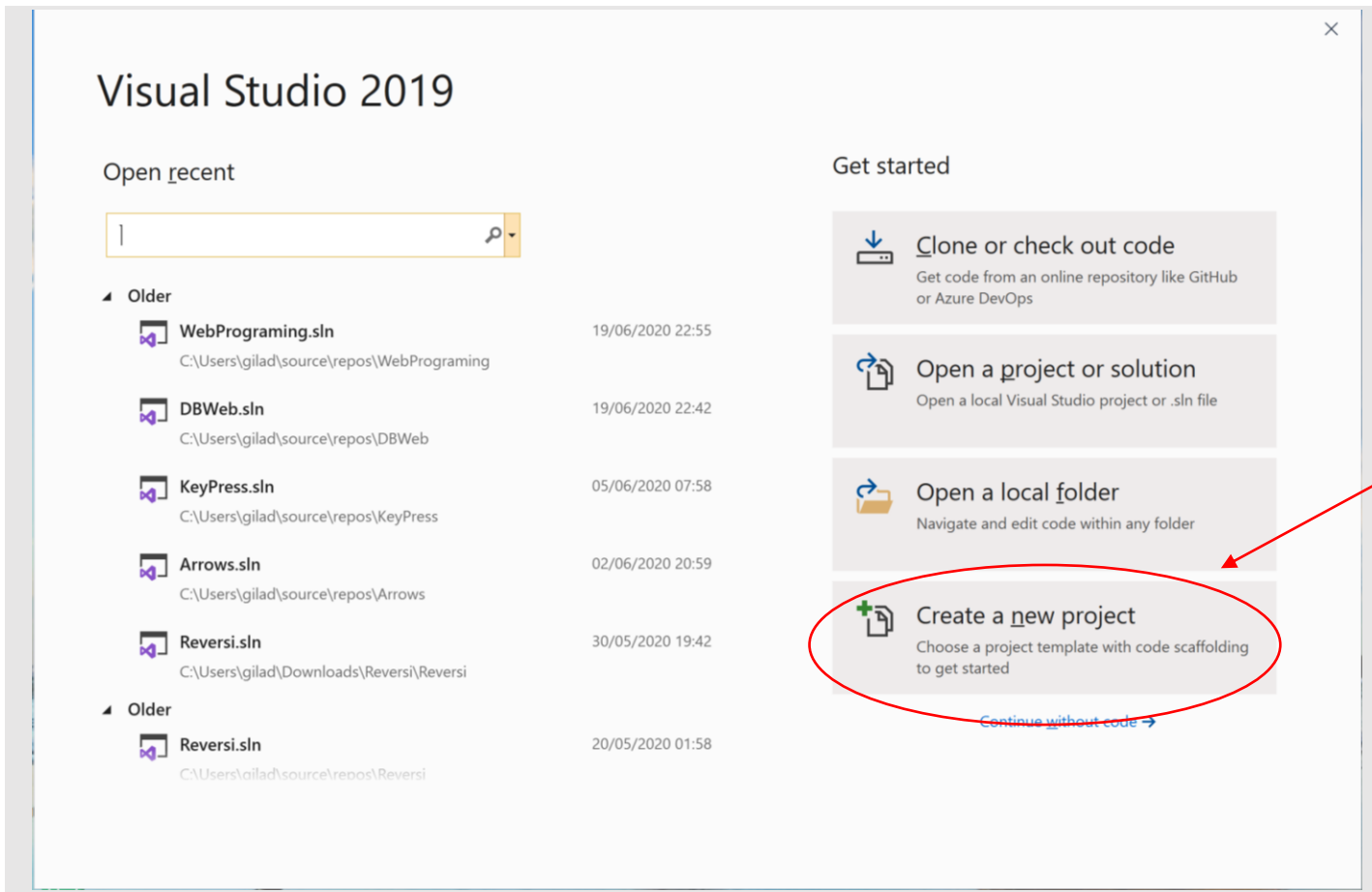
  <body>

    <h1>Hello World</h1>
    <p>My first web page.</p>

  </body>
</html>
```

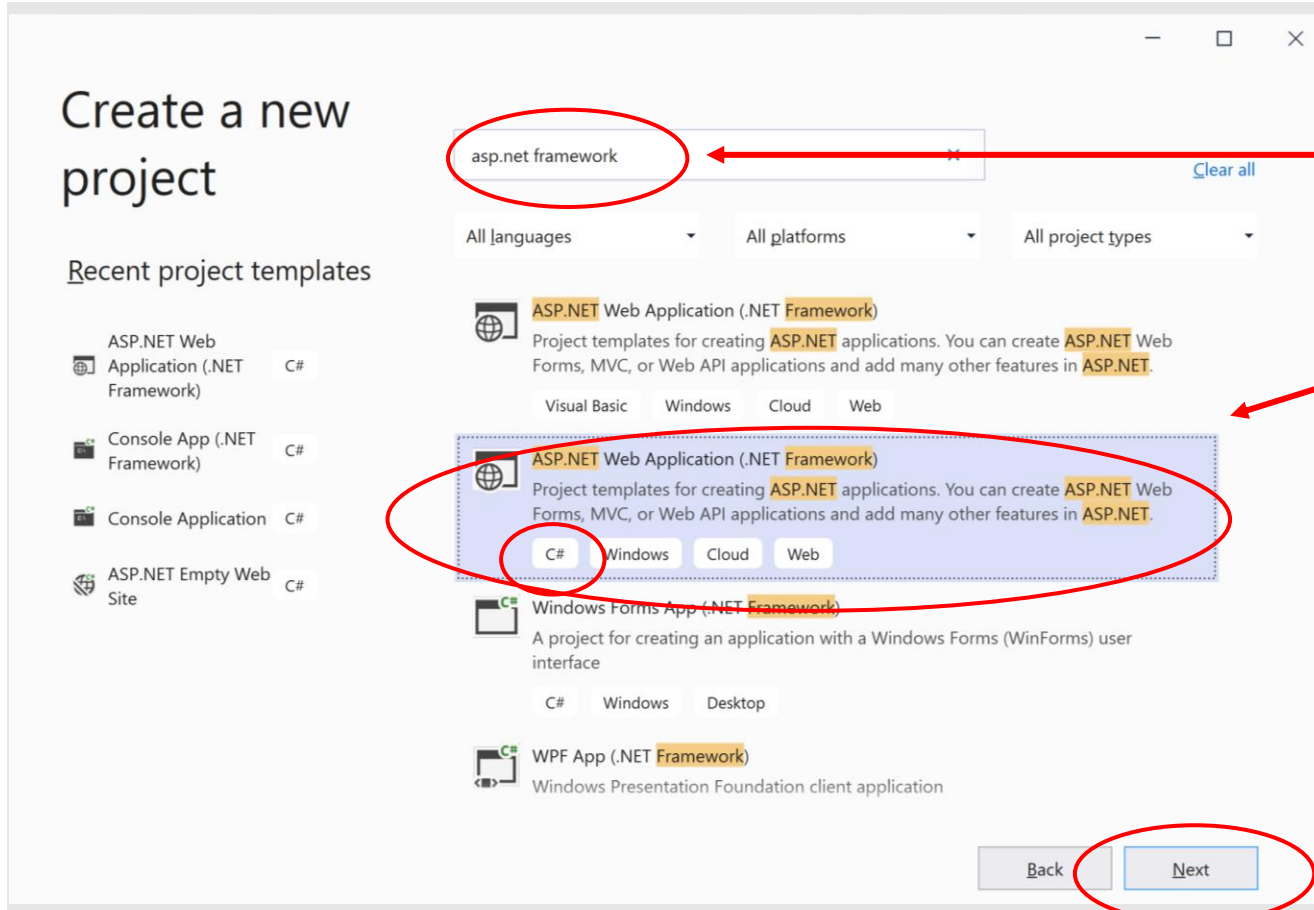
דוגמה לדף HTML.

לפני שנסביר את משמעות התגיות, נראה איך ליצור פרויקט בסביבת העבודה ואיך ליצור דף HTML בפרויקט.



• לאחר הפעלת ה Visual Studio יתקבל מסך פתיחה.

• עלינו ליצור פרויקט חדש



- בתיבת החיפוש יש לרשום:
- ASP.Net Framework
- לבחור את:
- ASP.Net Web Application (.Net Framework)
- לשים לב שבוחרים C#.
- ולוחצים Next.
- הערה: בסרטונים באתר אנו משתמשים בתבנית ישנה.

Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name

WebApplication2

Location

C:\Users\gilad\source\repos

Solution name *i*

WebApplication2

Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Back Create

- בדף הגדרת הפרויקט מספיק לתת שם לפרויקט.
- אין צורך לשנות את שאר השדות.
- וללחוץ על Create.

## Create a new ASP.NET Web Application



### Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.



### Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.



### MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.



### Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.



### Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

#### Authentication

No Authentication

[Change](#)

#### Add folders & core references

- Web Forms
- MVC
- Web API

#### Advanced

- Configure for HTTPS
- Docker support  
(Requires [Docker Desktop](#))
- Also create a project for unit tests

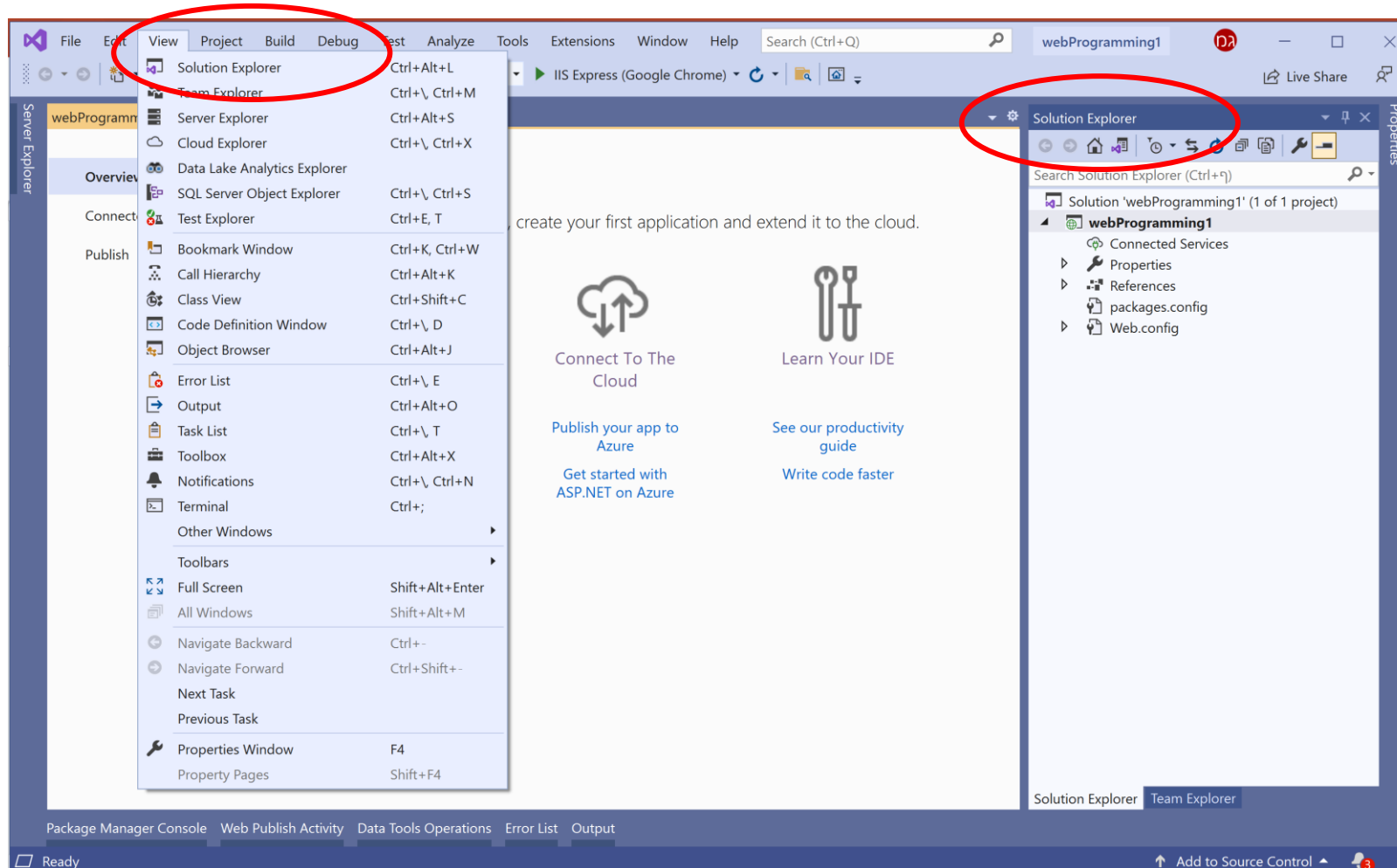
webProgramming1.Tests

Back

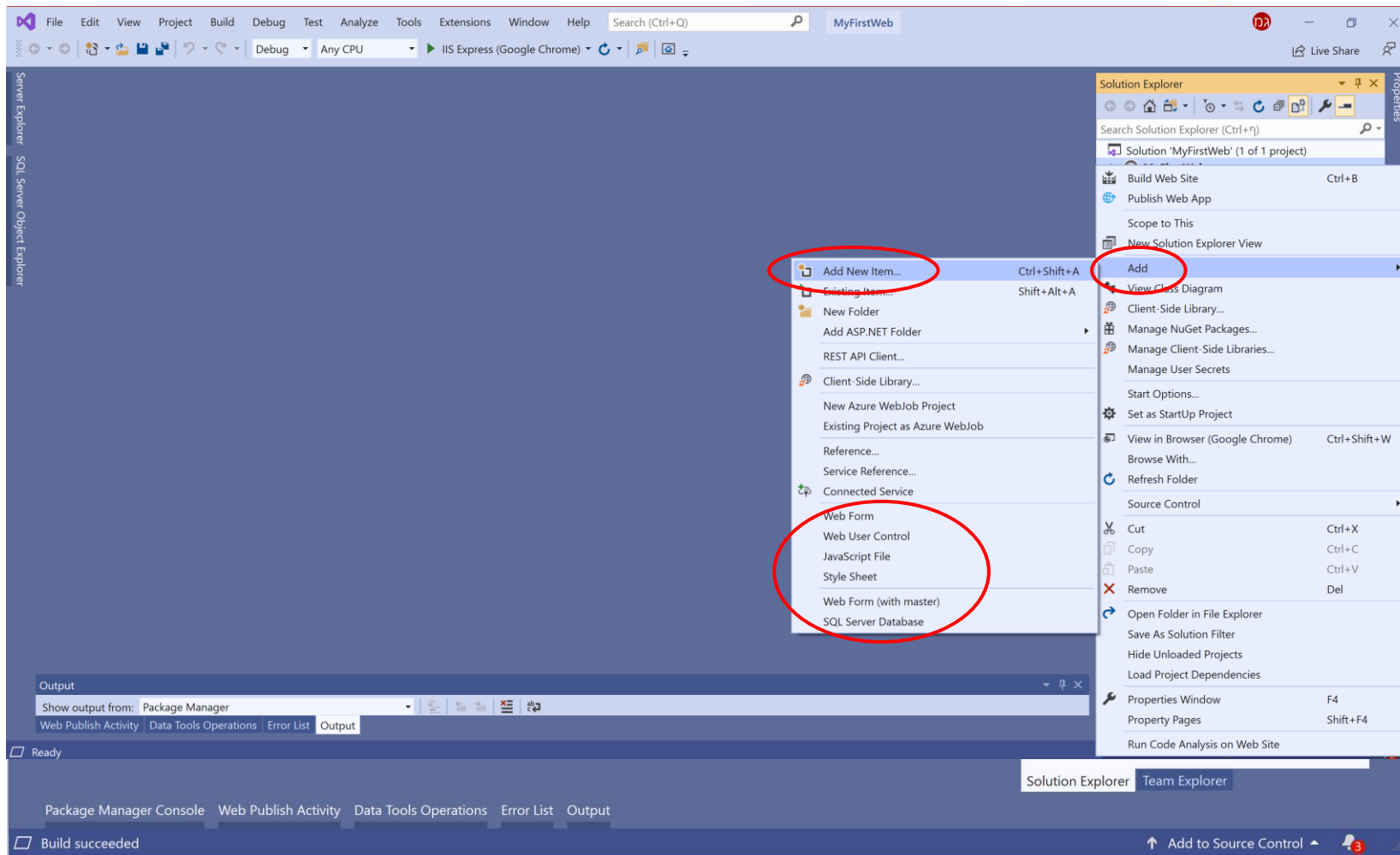
Create

## • לבחור פרויקט ריק

# כניסה ראשונה לפרויקט - Solution Explorer



- בכניסה ראשונה ברירת המחדל היא דף overview. ניתן לסגור אותו.
- אם חלונית Solution Explorer לא נפתחת, יש לפתוח אותה דרך תפריט view.

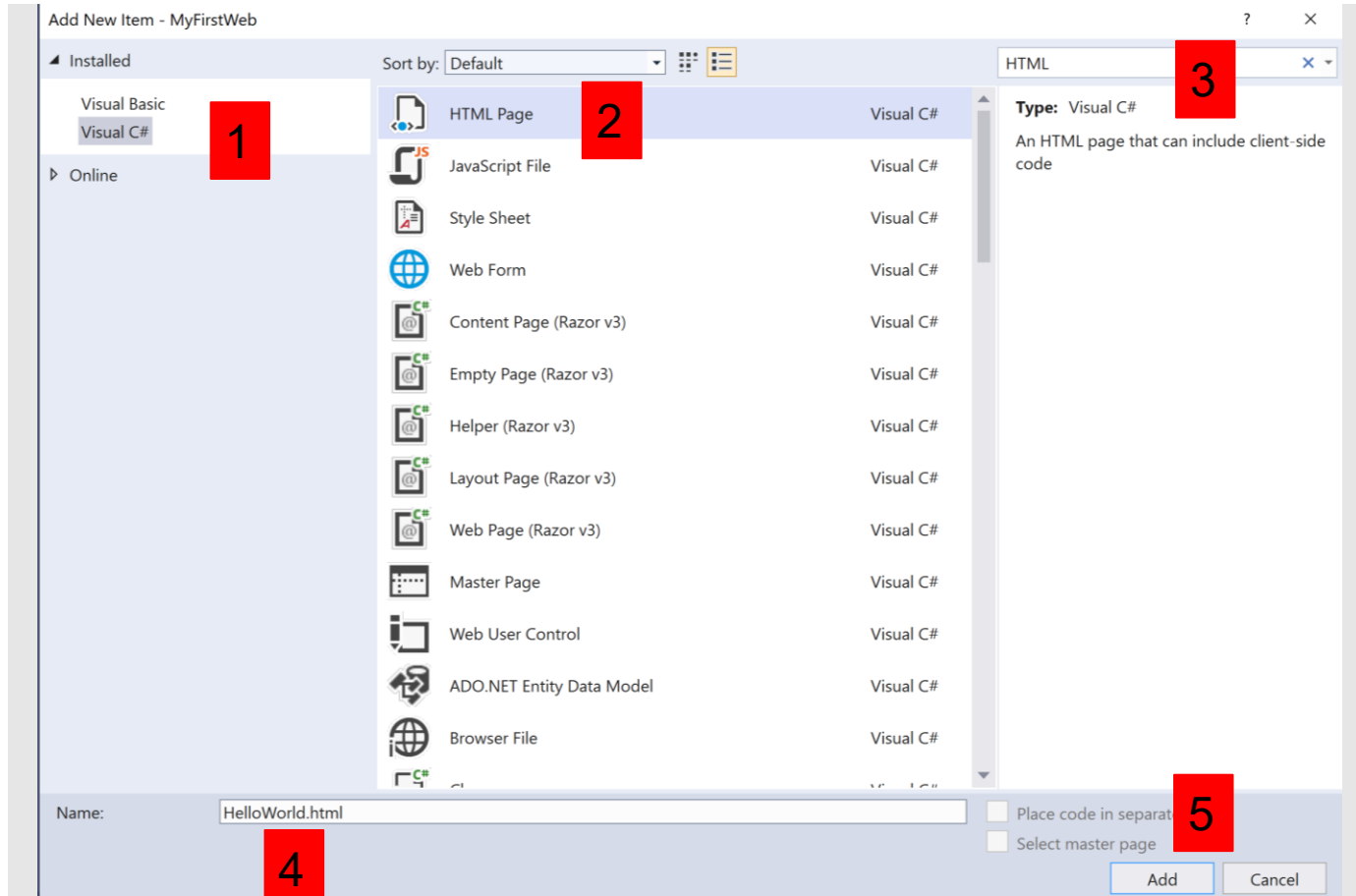


• בסביבת העבודה  
נצביע על שם הפרויקט  
ונלחץ על לחצן ימני  
בעכבר.

• נבחר ב Add

• בתפריט המשנה נבחר  
ב Add New Item...

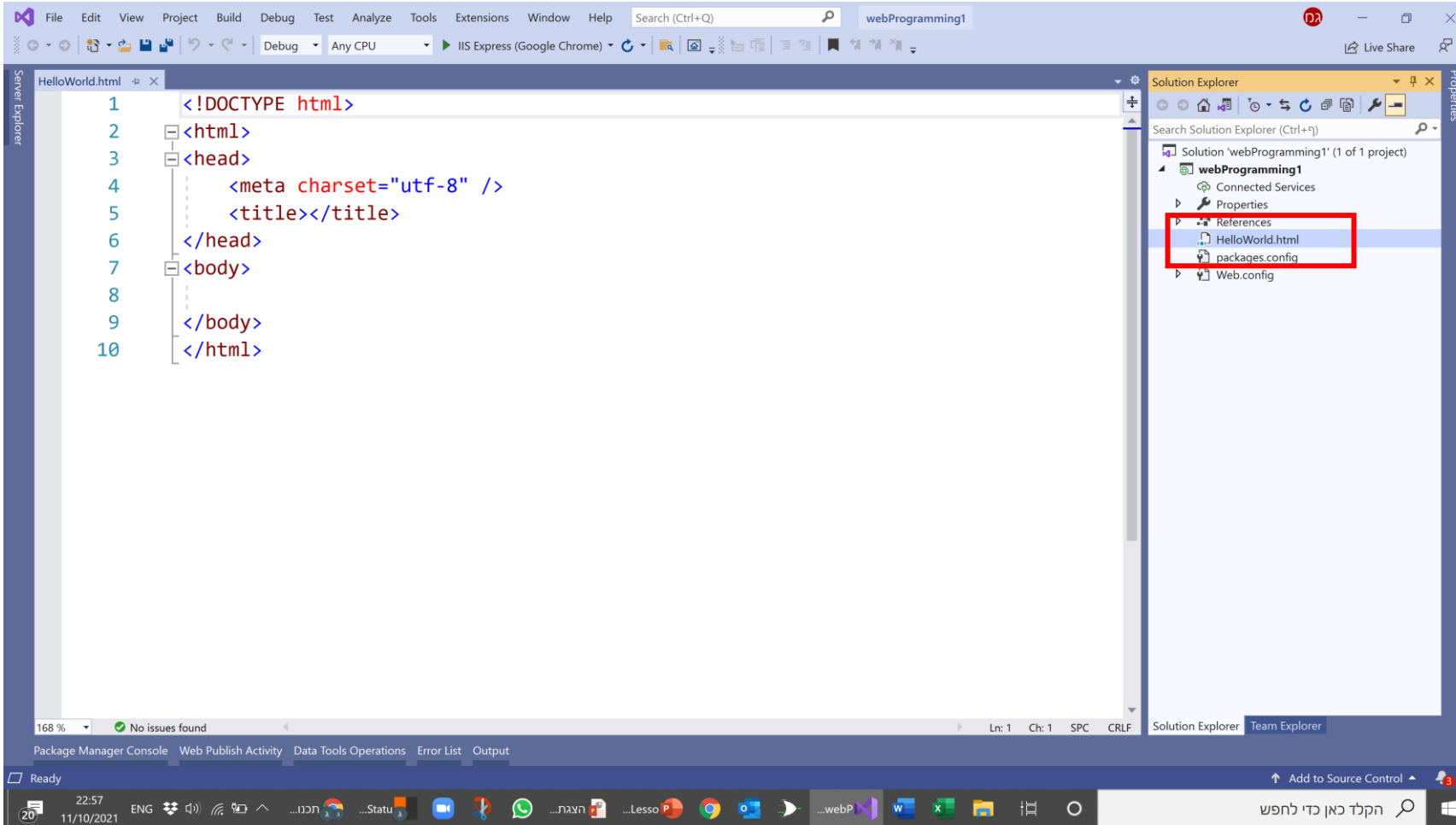
• קיצורי דרך המשתנים  
לפי השימוש בסביבת  
העבודה



יפתח חלון בחירה של סוגי פריטים אפשריים:

1. נודא ש Visual C# מסומן.
2. נבחר ב HTML Page.
3. אם האפשרות לא נמצאת ניתן לחפש בחלון החיפוש.
4. נרשום את שם הדף עם סיומת .html.
5. ונלחץ על Add.

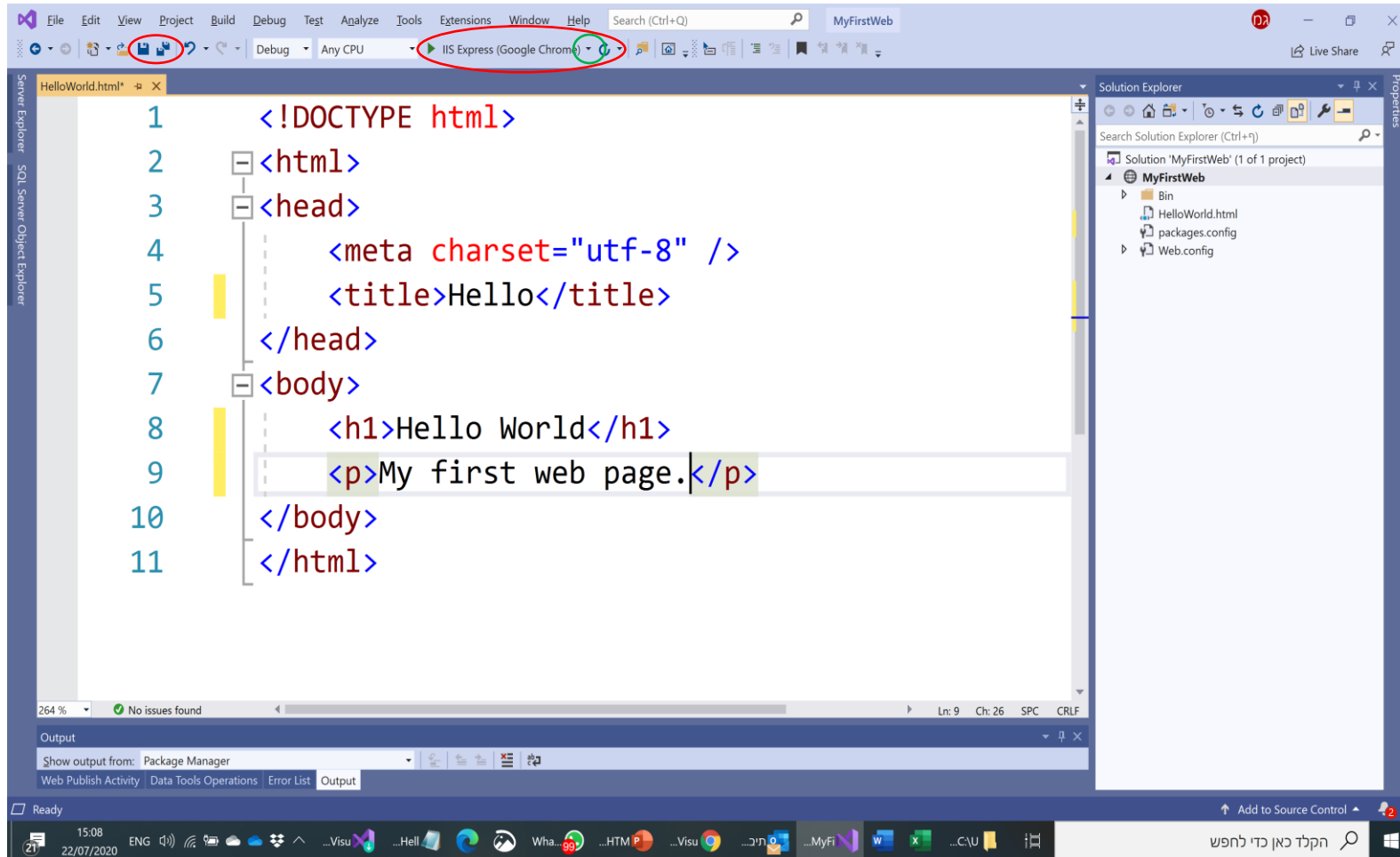




```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title></title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

- קיבלנו תבנית של דף HTML מוכן לעריכה.
- שם הדף החדש מופיע גם ב Solution Explorer

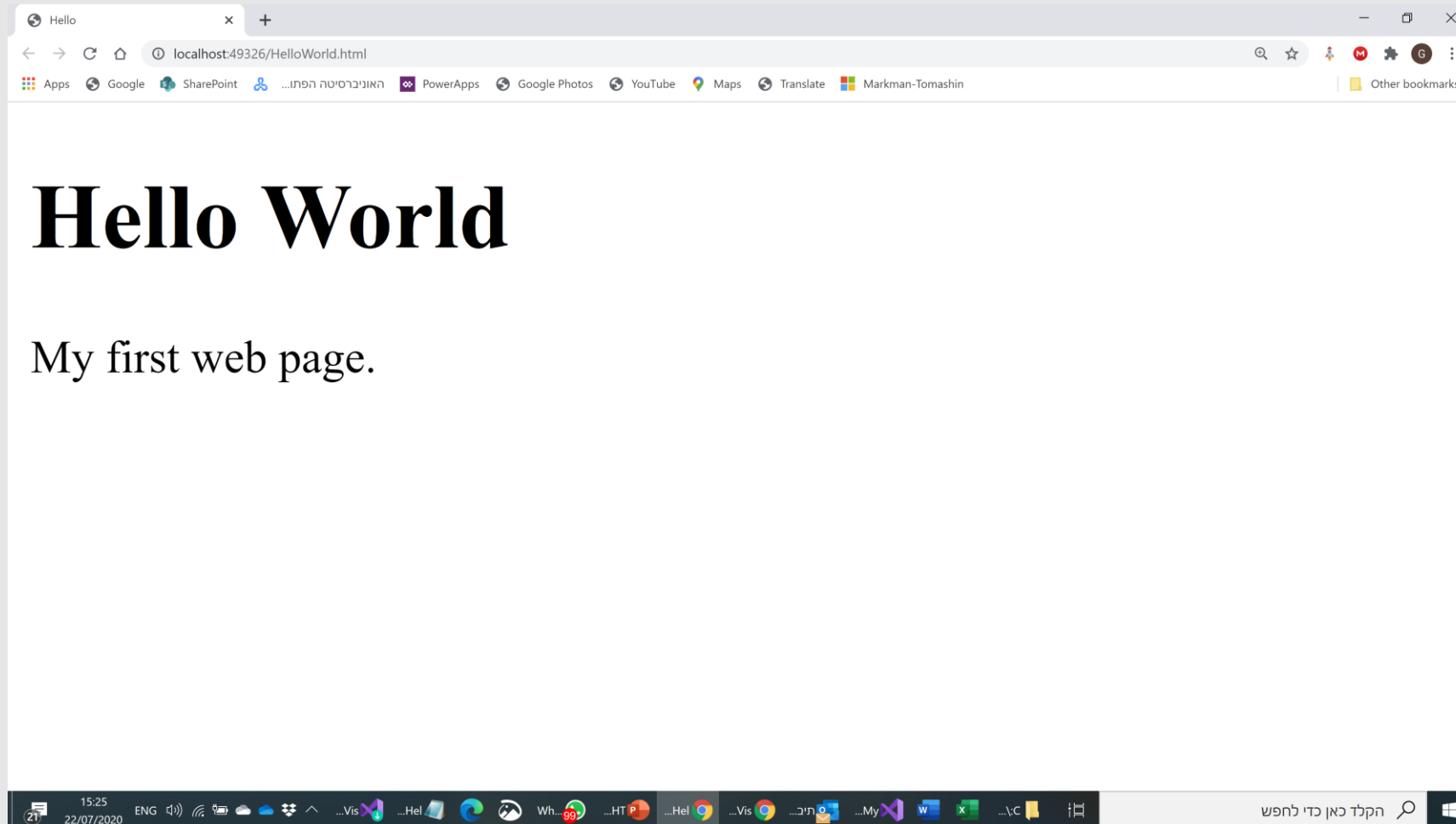
# Hello World



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <title>Hello</title>
6 </head>
7 <body>
8   <h1>Hello World</h1>
9   <p>My first web page.</p>
10 </body>
11 </html>
```

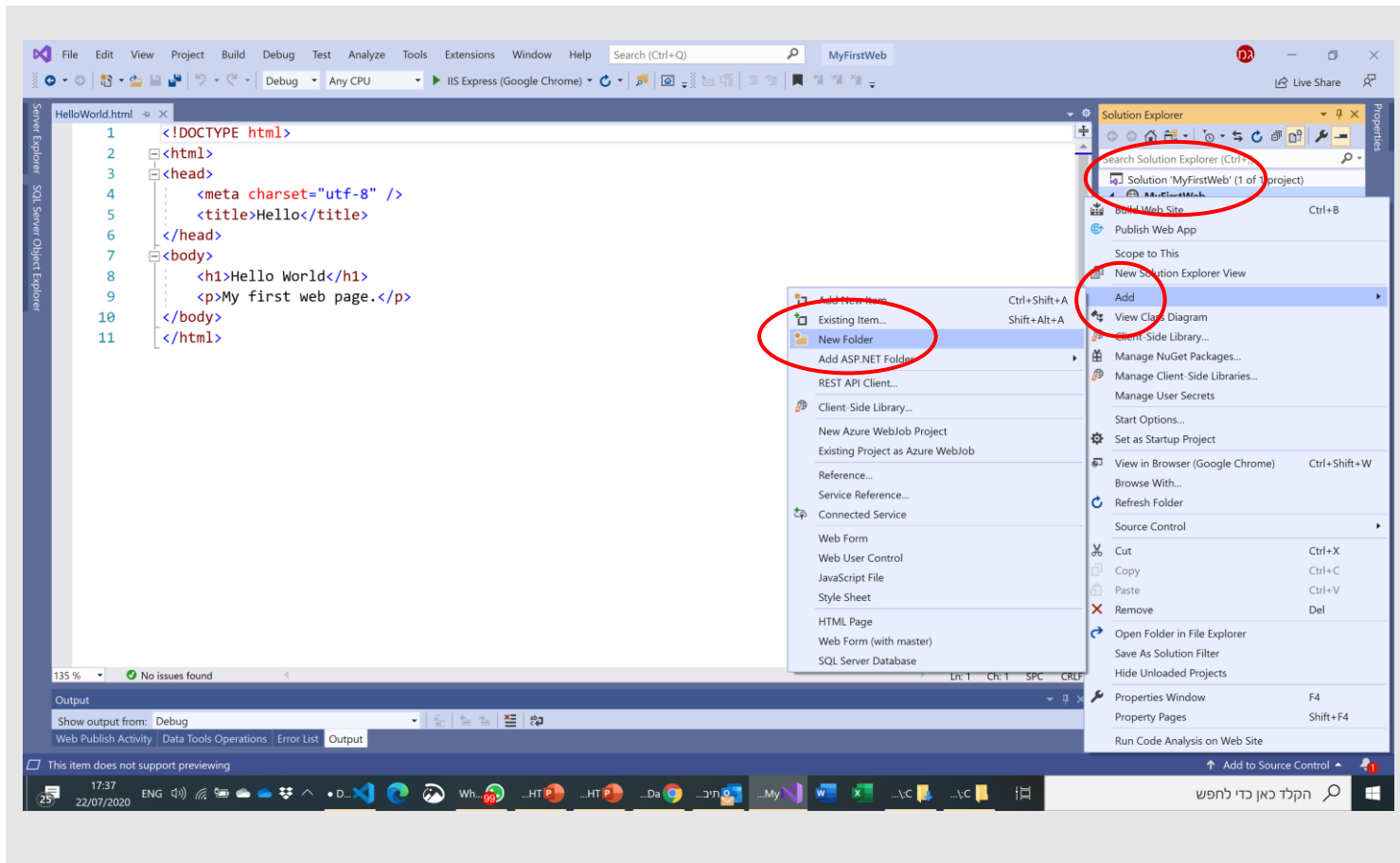
- נקליד את הקוד שרשמנו בהתחלה בתוך הדף החדש.
- בסיום נשמור.
- ונריץ בדפדפן.
- ניתן לבחור בחץ מימין את הדפדפן שיריץ את הדף.

# הדף הראשון Hello World



The screenshot shows a web browser window with a single tab titled 'Hello'. The address bar displays 'localhost:49326/HelloWorld.html'. The browser's bookmark bar includes 'Apps', 'Google', 'SharePoint', 'האוניברסיטה הפתוח...', 'PowerApps', 'Google Photos', 'YouTube', 'Maps', 'Translate', and 'Markman-Tomashin'. The main content area of the browser displays the text 'Hello World' in a large, bold, black serif font, followed by 'My first web page.' in a smaller, black serif font. The Windows taskbar at the bottom shows the time as 15:25 on 22/07/2020, the system language as ENG, and several open applications including Visual Studio, Internet Explorer, and various web browsers.

- אתר אינטרנט כולל קבצים רבים: קבצי HTML, תמונות, סרטים, קובצי עיצוב (נלמד בהמשך), קבצים המכילים קוד ועוד.
- על מנת לשמור על סדר באתר ולאפשר תחזוקה נוחה של האתר יש ליצור תיקיות לפי נושאים, ולשמור את הקבצים בתיקיות אילו.
- מומלץ לבנות את התיקיות הבאות (ניתן לבנות גם תיקיות נוספות לפי הצורך):
  - Pages – לדפי התוכן
  - Img – תמונות
  - Style – לקבצי עיצוב (נלמד בהמשך)
  - JS – לקבצי JavaScript (נלמד בהמשך).



- לחיצה על לחצן ימני על שם האתר "MyFirstWeb".
- לחיצה על Add
- לחיצה על New Folder.

- ניתן לעדכן שם תיקיה לאחר שנוצרה על ידי לחיצה על לחצן ימני של העכבר ובחירת Rename

- ניתן להוסיף קובץ HTML לתיקייה על ידי לחיצה על לחצן ימני על התיקיה ובחירה ב Add (כפי שהדגמנו קודם לכן).
- ניתן לשמור קבצים (תמונות וכד') בתוך תיקיה על ידי גרירת התמונות לתיקיה.
- כך למשל ניתן לגרור את הקובץ HelloWorld.html לתוך התיקיה HTML.
- אזהרה: קבצים של aspx אסור לגרור לתיקיה (נלמד בהמשך איך ליצור אותם).

# מבנה דף HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello</title>
  </head>

  <body>

    <h1>Hello World</h1>
    <p>My first web page.</p>

  </body>
</html>
```

- בראש הדף תופיע גרסת HTML – אנחנו נשתמש תמיד `<!DOCTYPE html>`
- בתחילת הקוד תופיע תגית `<html>` ובסיום תגית סוגרת `</html>`.
- ניתן (אך לא חובה) להוסיף קטע של ראש הדף – Head, המשמש להגדרות.
- לאחר מכן, גוף הדף ובו אנחנו כותבים את תגיות ה HTML.

# תגית - Tag

- שפת HTML בנויה מתגיות Tags. מבנה אופייני של תגית הוא:
  - `<tagName (attributes)> content ... </tagName>`
  - לדוגמה:
    - `<h1 id="MyH1">This is Heading</h1>`
    - המאפיינים (attributes) הם אופצינאליים, נדבר עליהם בהמשך.
    - תגית הסיום `</tagName>` מחוייבת במרבית המקרים, למעט יוצאים מן הכלל.
    - ניתן לרשום תגית בתוך תגית (Nesting), למשל כל התוכן נרשם בתוך תגית `<body>`.
  - שפת HTML אינה מבחינה בין אותיות גדולות לקטנות בתגיות. מקובל לרשום את התגיות באותיות קטנות.



# תגיות בסיסיות

תגית	פעולה
<code>&lt;h1&gt;...&lt;/h1&gt;, &lt;h2&gt;...&lt;/h2&gt;, ...</code>	כותרות
<code>&lt;p&gt;...&lt;/p&gt;</code>	פסקה
<code>&lt;br /&gt;</code>	מעבר שורה – אין תגית סיום.
<code>&lt;hr /&gt;</code>	קו אופקי – אין תגית סיום
<code>&amp;nbsp;</code>	רווח בתוך שורה (לא תגית)
<code>&lt;a href="https://www.park-hamada.co.il/"&gt;This is a link&lt;/a&gt;</code>	קישור לדף אחר – באינטרנט או בפרויקט שלכם
<code>&lt;img src="pic.jpg" width="100" height="150" /&gt;</code>	תמונה – אין תגית סיום
<code>&lt;title&gt; Page title &lt;/title&gt;</code>	הגדרת כותרת העמוד – נכלל ב head
<code>&lt;!-- ..... --&gt;</code>	הערות

## קישור <a>

### • תגית <a> בנויה מהחלקים הבאים:

```
<a href="url" target="value">This is a Link</a>
```

- href – הכתובת אליה עוברים עם לחיצה על הקישור.
- target – האם הדף החדש יטען באותו חלון (“\_self”) או בחלון חדש (“\_target”).
- המלל של הקישור שיראה למשתמש This is a Link.
- לחיצה על המלל תעביר את המשתמש לקישור.

```
<a href="https://en.wikipedia.org/wiki/Game_of_Thrones">Game of throne at Wikipedia</a>
```

- באינטרנט:

```
<a href="Daenerys.html">Daenerys in this site</a>
```

- דף באתר שלנו:

```
<a href="#summary">תמונה אחרונה</a>
```

- פנימי קישור למיקום אחר בדף שלנו:

- יש להוסיף בדף שלנו את המיקום אליו נעבור

```

```

- קיצור דרך ליצירת קישור לדפים בפרויקט שלנו ניתן לבצע על ידי Drag & Drop.

- הרחבה - כתובת מוחלטת ויחסית:

[https://www.w3schools.com/html/html\\_filepaths.asp](https://www.w3schools.com/html/html_filepaths.asp)

## תמונות <img>

```

```

- המאפיין src יכול לכלול קובץ באתר שלנו או תמונה באינטרנט.
- מומלץ להוריד את התמונות לספריה באתר שלכם ולא להסתפק בהפניה לאינטרנט, שכן לא ניתן להבטיח שהתמונה באתר האינטרנט אליו אתם מבפנים לא תשתנה או תמחק.
- רצוי לקבוע לכל תמונה את הגודל שלה באמצעות המאפיינים גובה או רוחב או בעיצוב CSS.
- קיצור דרך ליצירת קישור לתמונה השמורה באתר שלנו ניתן לבצע על ידי Drag & Drop.

## תמונה כקישור

- ניתן לקבוע תמונה כהיפר קישור, באמצעות הכנסת תגית `<img>` בתוך תגית `<a>`.

```
<a href="Main.html">
```

```
  
```

```
</a>
```

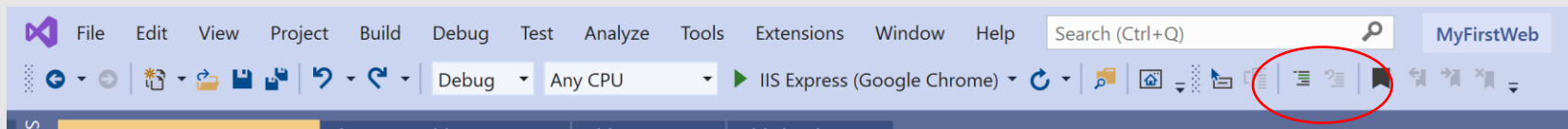
לחיצה על התמונה תעביר אותנו לדף המבוקש.

## הערות ותיעוד הקוד

- ניתן להוסיף בדף HTML הערות שרק המפתח יוכל לראות אותם ואילו המשתמש לא יראה אותם.

```
<!-- This is a comment -->
```

- ההערה יכולה לכלול מספר שורות, ובמהלך פיתוח אנחנו לעיתים "עוטפים" קוד בהערה כדי לא להפעיל אותו במהלך בדיקה.
- קיצור דרך - כפתור ההוספת הערה וביטול הערה ב visual studio





# CSS

- CSS – שפה לעיצוב דף HTML.
- CSS stands for **Cascading Style Sheets**
- המילה cascading (מדורג) מלמדת אותנו כי עיצוב רכיב בדף (רכיב האב) יחול גם על הרכיבים המוכללים בו (רכיבי הצאצאים).



# הדרכים להשתמש בעיצוב CSS

- קיימות שלוש דרכים לעשות שימוש בעיצוב CSS:
  - Inline – שימוש במאפיין style בתוך תגית ה HTML.
  - Internal – שימוש בתגית <style> בכותרת הדף (<head>).
  - External – כתיבת העיצוב בקובץ נפרד המחובר לדף ה HTML.
- השיטה המועדפת הינה השיטה החיצונית External. היא מאפשרת לנו להשתמש באותו עיצוב במספר דפים, ללא צורך בקוד כפול.

- עיצוב Inline נעשה באמצעות המאפיין style בתוך תגית HTML.

• דוגמה:

```
<body>  
  <h1 style="color:red; background-color:azure">Inline CSS Style</h1>  
  <p style="text-align:center">  
    עיצוב של התגית פסקה  
  </p>  
  <p> עוד פסקה </p>  
</body>
```

- העיצוב של ה body ישפיע גם על העיצוב של התגיות שבתוכו.
- העיצוב של הפסקה הראשונה לא חל על הפסקה השניה.

# תגיות <div> <span>

```

<body>
  <div style="background-color:lightblue;">
    <h1>Game of thrines</h1>
    <a href="Daenerys.html">Daenerys</a>
    <div>
      <a href="../GameOfThrones/Arya.html">Arya</a>
    </div>
  </div>
  <div style="background-color:aquamarine">
    
    <p>paragraph <span>another words</span></p>
  </div>
</body>

```

- תגית <div> מגדירה קטע **בדף** אותו ניתן לעצב באופן אחיד, כפסקה אחת.
- תגיד <div> "עוטפת" תגיות אחרות על מנת לאחד אותם ליחידת עיצוב אחת. פרט לכך, התגית אינה מבצעת עיצוב נוסף (מלבד ירידת שורה).
- תגית <div> יכולה להכיל כל אלמנט או כל תגית אחרת של דף ה HTML.
- תגית <span> מגדירה חלק **משורה** כיחידת עיצוב אחת, ונועדה לאפשר לעצב את החלק הזה באמצעות פקודת CSS.

• צבע:

- background-color: blue
- color: red;
- border: 2px solid black
- width, height : 100px / 50% / auto
- text-align: center/right/left
- display: none / block/ inline/flex;

• גבול

• גודל

• יישור טקסט:

• תצוגה – display

## • שוליים

- margin: 10px;
- Margin-left: 5px;
- Margin-right: 5px;

## • רווח

- Padding: 20px;
- Padding-bottom: 10px;
- Padding-top: 15px;

• נראות

• visibility: visible / hidden

• עיצוב טקסט

• font-family: "Times New Roman", Times, serif;

• font-style: normal; italic;

• font-size: 40px;

• יישור טקסט:

• direction: rtl / ltr;

• שקיפות

• opacity: 0.3 [0..1]

• תמונה כרקע

• Background-image: url("pic.jpg");

- מרכז תמונה בודדת:

- display: block;
- margin-left: auto; margin-right: auto;
- width: 40%;

- מרכז שתי תמונות ויותר

- יש לעטוף את התמונות ב `<div>` ואותו לעצב:

- display: flex;
- justify-content: center;
- height: 300px
  
- height: 100%

- בתמונות יש להגדיר את הגובה





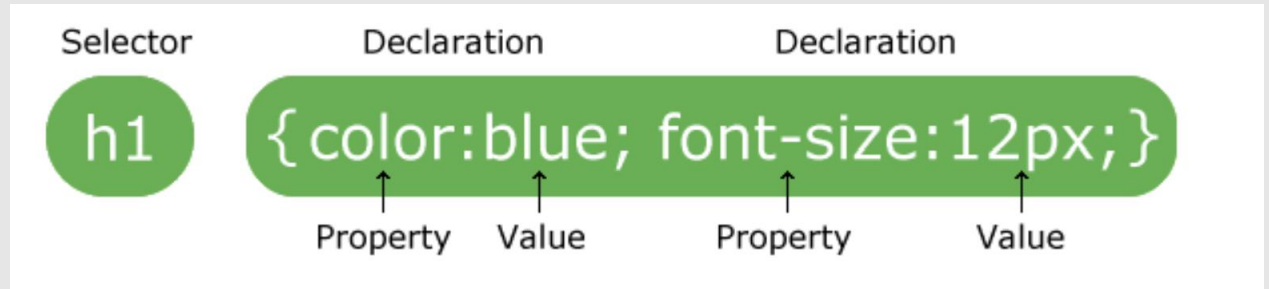
# CSS – Internal Style

```

<head>
  <style>
    body {
      direction: rtl
    }
    h1 {
      color: red;
      background-color: azure
    }
    p {
      text-align: center
    }
  </style>
</head>
<body>
  <h1>Inline CSS Style</h1>
  <p> עיצוב של התגית פיסקה </p>
  <p>עוד פסקה</p>
</body>

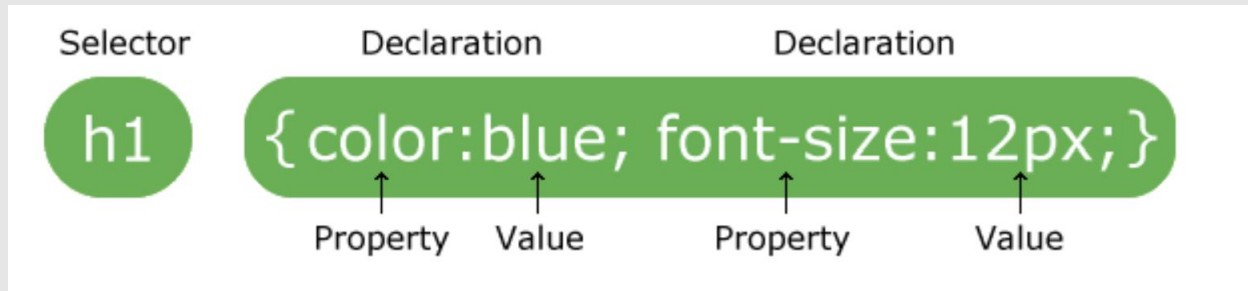
```

- הגדרות העיצוב נכתבות בראש המסמך בתגית <style> וחלות על כל המסמך.
- התחביר של פקודת CSS הוא:



- העיצוב חל על כל הסלקטורים בדף.

• הסלקטור נועד לזהות או לבחור את האובייקט בדף ה HTML שעליו יחולו פקודות העיצוב.



• סוגי הסלקטורים הם:

• תגיות HTML

• #id

• .class

• ריבוי ויחס

• Pseudo-Class

• Pseudo-Element

• [https://www.w3schools.com/css/css\\_selectors.asp](https://www.w3schools.com/css/css_selectors.asp)

```
div {  
  border:solid 2px red;  
  color: blue;  
  font-size: 20px;  
}  
  
h1 {  
  color: red;  
  background-color: azure  
}  
  
p {  
  text-align: center;  
  font-size: 12px;  
}
```

- עיצוב של תגית יחול על כל התגיות באותו דף.
- לא ניתן לעצב אותו סוג של תגיות בצורה שונה. על מנת לעצב שתי תגיות זהות בדרך אחרת נשתמש בסלקטור אחר.
- העיצוב של תגית `<div>` יחול על כל התגיות המוכלות בתוך ה `<div>`.

# סלקטור #id

- ניתן להוסיף לתגית מזהה ייחודי id ולהשתמש במזהה כדי לעצב את התגית לפי ה-id.

```
<div id="myId">  
  <p>פסקה שלישית</p>  
</div>
```

- הוספת ה-id נעשית במאפיין בתגית HTML כך:

- עיצוב ה-id בפקודת ה-CSS יהיה כך (#):

```
#myId {  
  color: red;  
  background-color: deepskyblue  
}
```

- Id הינו מזהה ייחודי. לא ניתן להגדיר לשתי תגיות אותו id.

# סלקטור class.

- באמצעות class ניתן להגדיר קבוצה של תגיות שיחול עליהם עיצוב אחיד. הקבוצה יכולה לכלול מספר תגיות מאותו סוג או תגיות מסוגים שונים.
- הוספת class נעשית במאפיין בתגית HTML כך:

```
<div class="myClass">  
  div with class = myClass & id = div2  
</div>
```

```
.myClass {  
  color:blue;  
  background-color:orangered;  
  border:solid 2px red;  
  text-align:center;  
}
```

- עיצוב ה class בפקודת ה CSS יעשה כך (.):

# ריבוי סלקטורים

- ריבוי - הפרדה של סלקטורים בפסיק (,)

- ניתן לעצב מספר סלקטורים בעת ובעונה אחת, כדי לקצר את הכתיבה:

```
h1, p1, div, #myId, .myClass {  
  background-color: blueviolet;  
  text-align:center;  
}
```

- התוצאה זהה לכתיבה העיצוב לכל סלקטור בנפרד.

```
/* unvisited link */  
a:link {  
  color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
  color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {  
  color: #FF00FF;  
}  
  
/* selected link */  
a:active {  
  color: #0000FF;  
}
```

- הוספת נקודותיים (: ) לסלקטור ולאחריהם "מצב".
- השימוש הוא לשם עיצוב של אלמנט במצב מסויים, כגון:
  - העכבר מצביע עליו (hover).
  - המשתמש לחץ / לא לחץ על הקישור (visited)
  - האלמנט נמצא בפוקוס (focus)
  - ועוד...

• להרחבה ראה קישור :

• [https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp)



```
<style>
  h1 {
    text-align: center;
    background-color: aqua
  }
  p {
    direction: rtl;
    color: blue;
  }
  #logo {
    display: block;
    width: 50%;
    margin-left: auto;
    margin-right: auto;
  }
  #paraDiv {
    border: solid 3px black;
    margin-right: 30px;
    margin-left: 30px;
    padding: 10px;
    background-color: lightgray
  }
  #flexDiv {
    display: flex;
    justify-content: center;
    height: 300px;
    margin: 10px
  }
  .flexImg {
    height: 100%
  }
</style>
```

# דוגמה – עיצוב Internal

בדף HTML יש להחליף את העיצוב (תגית style) ב ID מתאים או ב Class מתאים.

# CSS – External Style

- עיצוב באמצעות קובץ עיצוב חיצוני כולל את השלבים הבאים:
    - קליק ימני בספרייה המתאימה והוספת קובץ מסוג CSS Style Sheet.
    - כתיבת הוראות העיצוב בהתאם לתחביר של פקודת CSS.
    - קישור דף HTML עם דף ה CSS באמצעות תגית Link בכותר <head>.
- ```
<link href=" ../CSS/ExternalStyle.css" rel="stylesheet" />
```
- ניתן לקשר קובץ Style Sheet אחד למספר דפים.

```
ExternalStyle.css  X
1  body {
2      direction: rtl
3  }
4
5  h1 {
6      color: red;
7      background-color: azure
8  }
9
10 p {
11     text-align: center
12 }
13
```

- קובץ העיצוב מכיל אך ורק הוראות CSS.
- על מנת להפעיל את הוראות העיצוב על דף מסויים יש לקשר את דף ה HTML לדף CSS.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title></title>  
    <link href="../CSS/ExternalStyle.css" rel="stylesheet" />  
  </head>  
  <body>  
    <h1>Inline CSS Style</h1>  
    <p> עיצוב של התגית פיסקה </p>  
    <p>עוד פסקה</p>  
  </body>  
</html>
```

- בדף HTML אין צורך בהוראות עיצוב.
- הדף כולל רק קישור לקובץ .Style Sheet

• אם רכיב בדף HTML מעוצב במקביל על ידי מספר הוראות, סדר קדימיות העיצוב הוא כך:

1. העיצוב Inline בעל הקדימות הראשונה. גובר על שאר העיצובים.
2. עיצוב Internal ו External בעלי אותה עדיפות. גובר העיצוב שהוגדר "אחרון". אם הקישור מוגדר לפני העיצוב הפנימי (<style>) – העיצוב הפנימי גובר, ולהפך.
3. ברירת המחדל של הדפדפן.

- ניתן לכתוב הערות comments בדף CSS. הדפדפן מתעלם מהערות אילו.

```
/*comments*/  
/*  
th {  
  background-color:green;  
}  
*/
```

- הערה נועדה גם לתיעוד וגם ל debugging.

- ניתן "לעטוף" קוד בהערה כך שהדפדפן יתעלם ממנו מבלי שמחקנו אותו. מאוד נוח לבדיקות.



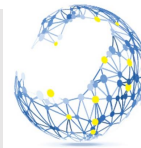
# טבלאות



```
<body>
  <table>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </table>
</body>
```


- טבלה בסיסית כוללת :
- תגית טבלה - `<table>`
- תגית שורה - `<tr>`
- תגית תא - `<td>`
- בדוגמה טבלה עם שתי שורות ושלושה תאים בכל שורה (עמודות)

# יצירת טבלה ב HTML




```
<table>
  <thead>
    <tr>
      <th></th>
      <th></th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>
```



- טבלה יכולה לכלול בנוסף:
- חלוקה לראש טבלה <thead> ולגוף הטבלה <tbody>.
- התאים בראש הטבלה – מוגדרים באמצעות <th>.
- השימוש בחלוקה זו מאפשר לעצב בצורה שונה את ראש הטבלה לעומת גוף הטבלה.

- תוכן הטבלה מוכנס אך ורק בתוך תאי התוכן <td> או <th>.
- תוכן התאים יכול לכלול כל תגית של HTML, לרבות טקסט, קישורים, תמונות וכד'.

```
<tr>
  <td>Arya Stark</td>
  <td>
    <p>
      Arya is the third child and younger daughter of Lord Eddard Stark and his wife Lady Catelyn Stark. She is tomboyish and is often mischievous. She wields a small sword and is trained in the art of assassination.
    </p>
  </td>
  <td>
  <td>
    <a href="https://en.wikipedia.org/wiki/Arya_Stark" data-bbox="198 748 324 768">Arya Stark</a>
  </td>
</tr>
```

Name	Summary	picture	Wiki
Arya Stark	Arya is the third child and younger daughter of Lord Eddard Stark and his wife Lady Catelyn Stark.		<a href="https://en.wikipedia.org/wiki/Arya_Stark">Arya</a>


wife Lady Catelyn Stark.  
male pursuits,

w,

# עיצוב גבולות טבלה



```
table, td, th {
  border: 1px solid blue;
}
```

Name	Summary	picture	Wiki
Arya Stark	Arya is the third child and younger daughter of Lord Eddard Stark and his wife Lady Catelyn Stark.		<a href="#">Arya</a>

- המאפיין border מגדיר את רוחב פס הגבול, סגנון וצבע.
- ניתן להגדיר גבול לחלק מההיקף עם המאפיינים: border-top; border-bottom; border-left; border-right
- הרווח בין הגבולות יכול להקבע על ידי המאפיין: border-spacing: 10px.
- ביטול רווח בין גבולות נעשה על ידי הפקודה: border-collapse: collapse

```
#myTable {  
  width: 60%;  
  height: 300px;  
  margin-right: auto;  
  margin-left: 100px;  
}
```

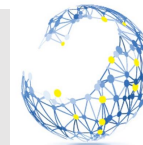
```
#myTable {  
  width: 60%;  
  height: 300px;  
  margin-right: auto;  
  margin-left: auto;  
}
```

- עיצוב התגית `<table>` מאפשר להגדיר:

- רוחב טבלה: בפיקסלים או אחוזים
- גובה טבלה: בפיקסלים או אחוזים.
- מיקום אופקי של הטבלה: כנ"ל.

- מרכז הטבלה יעשה על ידי הפקודות:

# Hover



```
tr:hover {  
    background-color:lightblue;  
}  
  
td:hover {  
    background-color: lightgreen;  
}
```

- ניתן להגדיר שינוי של צבע בעת הצבעה על תא בטבלה באמצעות העכבר על ידי `.td:hover`.

- ניתן להגדיר שינוי של צבע בעת הצבעה על שורה בטבלה על ידי `.tr:hover`.

• במקרים בהם הטבלה שלנו ארוכה או רחבה - ניתן להוסיף פסי גלילה.

```
<div id="divTable">  
  <table class="tbl">...</table>  
</div>
```

```
#divTable {  
  overflow-y: scroll;  
  height: 600px;  
}
```

• נעטוף את הטבלה ב div.

• נקבע את גובה ה-div.

• נוסיף ל Div מאפיין overflow-y: scroll

• ניתן להשתמש במאפיינים:

• Overflow – מטפל בחריגה אופקית ואנכית – יתווספו שני פסי גלילה

• Overflow-x – מטפל בחריגה אופקית בלבד – יתווסף פס גלילה לרוחב בלבד.

• Overflow-y – מטפל בחריגה אנכית בלבד – יתווסף פס גלילה אנכי בלבד.

# טפסים

# Forms



- טופס נועד לקלוט מידע מן המשתמש ולשלוח את המידע לשרת.
- דוגמה לטופס כניסה לאתר Login:

```
<form id="form1" method="get" action="../WebPages/Main.aspx">  
  <label for="userName">UserName:</label>  
  <input type="text" name="userName" id="userName" />  
  <br /> <br />  
  <label for="pass">Password:</label>  
  <input type="password" name="pass" id="pass" />  
  <br /> <br />  
  <input type="submit" />  
</form>
```

UserName: Password:

- התגית `<form>` נועדה להכיל פקדים (elements) לצורך קליטה של נתונים מן המשתמש.
- בדוגמה שלנו הפקדים בהם השתמשנו הם:
  - `<label>` - תווית בלבד שאינה קולטת נתונים.
  - `<input type="text">` - תיבה לקלטית טקסט.
  - `<input type="password">` - תיבה לקליטת סיסמה.
  - `<input type="submit">` - כפתור השולח את הנתונים לשרת.

```
<form id="form1" method="get" action=" ../WebPages/Main.aspx">  
  <label for="userName">UserName:</label>  
  <input type="text" name="userName" id="userName" />  
  <br /> <br />  
  <label for="pass">Password:</label>  
  <input type="password" name="pass" id="pass" />  
  <br /> <br />  
  <input type="submit" />  
</form>
```

UserName:

Password:

- המאפיין method של התגית <form> קובע את הפרוטוקול לפיו נשלחים הנתונים לשרת.

```
<form id="form1" method="post" action=" ../WebPages/Main.aspx" >
```

- קיימות שתי שיטות: get ו- post למשלוח נתונים כפי שהסברנו בהתחלה.
- המאפיין action של התגית <form> קובע את הדף בשרת אליו ישלחו הנתונים.
  - בדוגמה לעיל, דף השרת הוא main.aspx.
  - ברירת המחדל (אם לא מציינים את המאפיין action) הוא הדף הנוכחי.
- ניתן להסתפק בתגית <form id="myform"> מבלי להגדיר את המאפיינים הנוספים.

- המאפיין id כמו בכל תגית נועד לתת שם ייחודי לטופס.
- ה id משמשת אותנו לשלוש מטרות עיקריות:
  - האחת, שימוש בטופס באמצעות קוד בצד המשתמש ב javascript.
  - השניה, שימוש בטופס באמצעות קוד בצד השרת #c.
  - השלישית, עיצוב הטופס באמצעות css.
- שימוש בטפסים באמצעות קוד ילמד בהמשך הקורס.

# פקד <input>

- פקד input הינו האלמנט החשוב ביותר של הטופס. הפקד כולל למעשה מספר סוגים של אלמנטים הנקבעים בהתאם למאפיין ה type.
- לדוגמא תיבת טקסט, סיסמה, כפתור רגיל, כפתור "שלח" (submit) ואפס.

```
<label for="userName">UserName:</label>
<input type="text" name="userName" id="userName" />

<label for="pass">Password:</label>
<input type="password" name="pass" id="pass" />

<input type="button" name="btn" id="btn" value="button" />

<input type="submit" value="send" />
<input type="reset" value="reset" />
```

UserName:

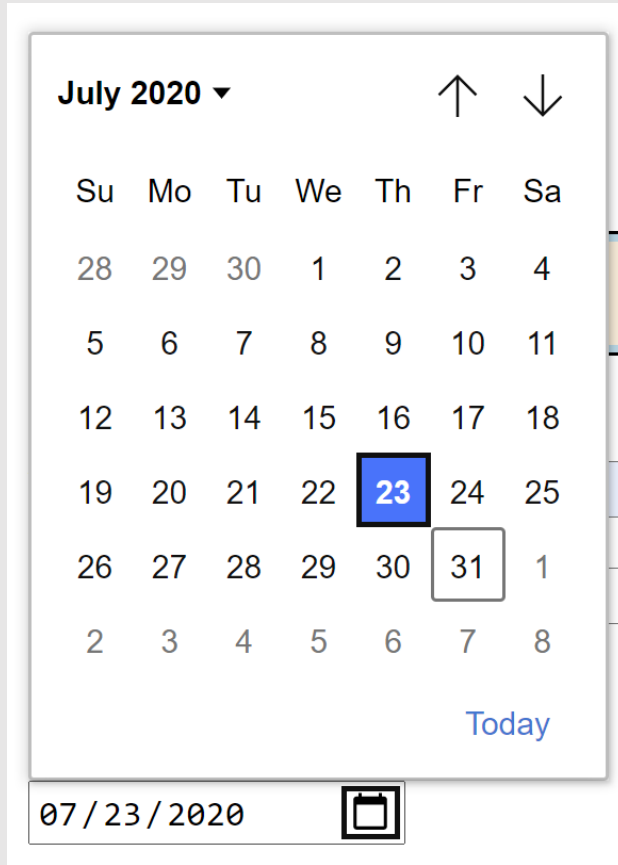
Password:

- המאפיין date מאפשר לקלוט תאריכים.

```
<input type="date" name="date" id="date" />
```

- ניתן גם לקלוט תאריך ושעה או רק שעה:

```
<input type="datetime-local" name="datetime" id="datetime" />  
<input type="time" name="time" id="time" />
```





- תיבת הבחירה מאפשרת בחירה מרובה של אפשרויות (בהבדל מכפתורי רדיו).

```
<label>Chosse your hobby: </label> <br />
<input type="checkbox" id="football" name="football" value="footbal">
<label for="football"> football</label> &nbsp;
<input type="checkbox" id="music" name="music" value="music">
<label for="music"> music</label> &nbsp;
<input type="checkbox" id="basketball" name="basketball" value="basketball">
<label for="basketball"> basketball</label>
```

Chosse your hobby:

football    music    basketball



# רשימה נפתחת - <select>

- רשימה נפתחת מוגדרת באמצעות רכיבים <select> ו- <option>

```
<label for="colors">Choose a color:</label>
<select id="colors" name="colors">
  <option value="null" selected="selected"></option>
  <option value="red">Red</option>
  <option value="blue">Blue</option>
  <option value="green">Green</option>
  <option value="yellow">Yellow</option>
</select>
```

Choose a color:

- Red
- Blue
- Green
- Yellow

- רשימת בחירה מאפשרת לקבוע רשימה של ערכים שניתן להכניס בתיבת `<input>`.
- ניתן לרשום בתיבה גם ערכים שאינם ברשימה.

```
<label for="browsers">Choose a browser:</label>  
<input list="browsers" name="browser">  
<datalist id="browsers">  
  <option value="Internet Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

Choose a browser:  ▼

- Internet Explorer
- Firefox
- Safari

- קיימים פקדים נוספים הניתנים לשימוש בדף HTML.
- לרשימה מלאה והסברים של הפקדים:
- [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
- כמובן שניתן לעצב כל אחד מן הפקדים באמצעות פקודות CSS כפי שלמדנו, וניתן להוסיף לכל תגית של פקד id או class לצורך עיצוב.
- חשוב מאוד – יש להוסיף לכל פקד את המאפיין name=".." אחרת התוכן לא ישלח לשרת.

# רשימות ותפריטים

- בפרק זה ניישם את מה שלמדנו בשיעורים הקודמים:
  - נלמד על תגיות HTML ליצירת רשימות Lists.
  - נלמד כיצד ליצור תפריט ניווט – רשימה של קישורים.
  - נלמד על מודל הקופסה בעיצוב דף HTML.
  - נלמד כיצד לעצב את תפריט הניווט באמצעות CSS, וליצור תפריט מעוצב – Horizontal Bar.
- את העיצוב לקחנו מאתר w3school בכתובת:
  - [https://www.w3schools.com/css/css\\_navbar\\_horizontal.asp](https://www.w3schools.com/css/css_navbar_horizontal.asp)

## • רשימה ממוספרת &lt;ol&gt;

```
<ol>
  <li> Main Page </li>
  <li> Daenerys </li>
  <li> Jon Snow </li>
  <li> Arya </li>
  <li> Ned Stark </li>
</ol>
```

1. Main Page
2. Daenerys
3. Jon Snow
4. Arya
5. Ned Stark

## • התוצאה:

## • רשימה לא ממוספרת &lt;ul&gt;

```
<ul>
  <li> Main Page </li>
  <li> Daenerys </li>
  <li> Jon Snow </li>
  <li> Arya </li>
  <li> Ned Stark </li>
</ul>
```

- Main Page
- Daenerys
- Jon Snow
- Arya
- Ned Stark

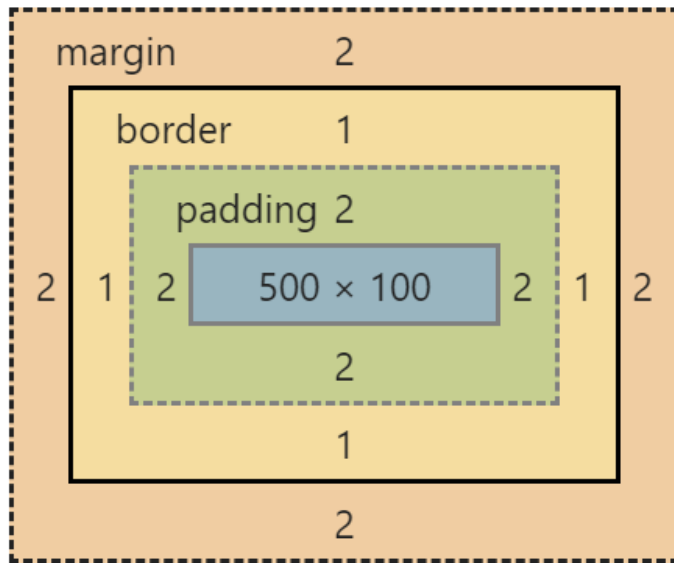
## • התוצאה:

- ניתן בתוך פריט הרשימה – `<li>` list item להוסיף קישורים, כך שלחיצה על הרשימה תעביר אותו לדף המבוקש.

```
<ul>  
  <li> <a href="../GameOfThrones/MainPage.html">Main Page</a> </li>  
  <li> <a href="DaenerysPage.html">Daenerys</a> </li>  
  <li> <a href="JonSnow.html">Jon Snow</a> </li>  
  <li> <a href="../GameOfThrones/Arya.html">Arya</a> </li>  
  <li> <a href="../GameOfThrones/NedStark.html">Ned Stark</a> </li>  
</ul>
```

- [Main Page](#)
- [Daenerys](#)
- [Jon Snow](#)
- [Arya](#)
- [Ned Stark](#)

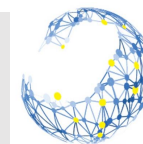
- התוצאה:
- לחיצה על קישור תעביר אותנו לדף המבוקש.



- ניתן לתאר כל רכיב HTML בדף כקופסה:
- התוכן - בדוגמה בגודל 500x100, רקע כחול.
- מרווח padding - המקיף את התוכן בדוגמה ברוחב 2px.
- גבול border - התוחם את התוכן והמרווח. בדוגמה ברוחב 1px, בסגנון רצוף solid, ובצבע שחור.
- שוליים margin – המפרידים בין הרכיבים בדף. בדוגמה שלנו ברוחב 2px.
- באמצעות CSS ניתן לעצב כל אחד מרכיבים אילו.



# עיצוב רכיבי "הקופסה" של הרשימה



## • התוצאה:



- [Main Page](#)
- [Daenerys](#)
- [Jon Snow](#)
- [Arya](#)
- [Ned Stark](#)
- [Wikipedia](#)

Game Of Thrones

## • נעצב את הרשימה כך:

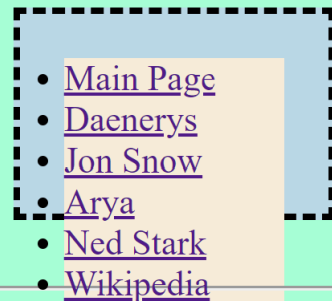
```
ul {  
    background-color: lightblue;  
    width: 500px;  
    padding: 20px;  
    border: dashed 3px black;  
    margin: 30px;  
    overflow: hidden;  
}  
li {  
    background-color: antiquewhite;  
}
```

• אם התוכן חורג מהגודל המוגדר עבורו ההתנהגות שלו נקבעת על ידי המאפיין overflow. הערכים של overflow הם:

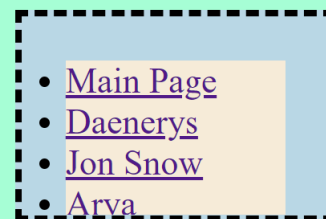
• Scroll – מתווספים פסי גלילה המאפשרים לגלול את התוכן:



• Visible – מציג את התוכן הגולש מתחום האובייקט:



• Hidden – חותך את התוכן הגולש:



• ניצור קובץ עיצוב ייעודי לתפריט – HorizontalBar.css

• נקשר את הקובץ לדף HTML.

• נעצב את הרשימה כך:

```
ul {  
  list-style-type: none; /*Hide the bullets*/  
  background-color: lightblue;  
  padding: 2px;  
  border:solid 1px black;  
  margin: 2px;  
  overflow: hidden;  
}
```

• נקבל:

[Main Page](#)  
[Daenerys](#)  
[Jon Snow](#)  
[Arya](#)  
[Ned Stark](#)  
[Wikipedia](#)

```
li {  
  float:left;  
}
```

```
li a {  
  background-color: antiquewhite;  
  display:block; /*make the box around the link clickable*/  
  padding:10px; /*space around the block*/  
  margin: 1px;  
  width: 100px;  
}
```

• נהפוך את התפריט לאופקי צמוד לשמאל:

• נשנה את העיצוב של הקישורים:

• נקבל:

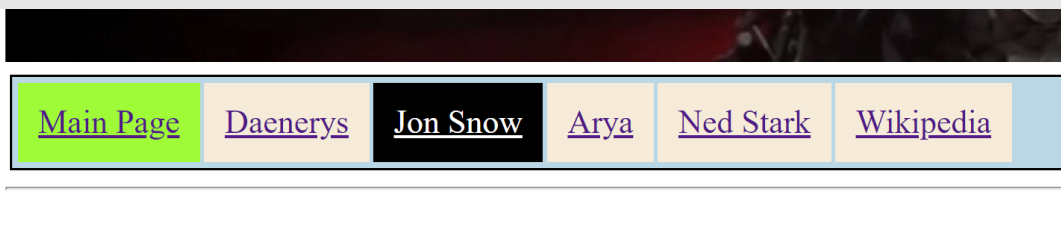
[Main Page](#)[Daenerys](#)[Jon Snow](#)[Arya](#)[Ned Stark](#)[Wikipedia](#)[Main Page](#)[Daenerys](#)[Jon Snow](#)[Arya](#)[Ned Stark](#)[Wikipedia](#)

```
li a:hover {  
    background-color :black;  
    color:white;  
}
```

```
.active {  
    background-color :lawngreen;  
}
```

- נוסף הוראה המשנה את צבע הקישור בעת הצבעה עם העכבר:

- נוסף עיצוב לקישור של הדף הנוכחי ונוסף לקישור הנוכחי class="active"



- נקבל:

- נוסף בתוך אחד או יותר מ `<li>` רשימה נפתחת:

```
<li class="dropDown">  
  <a href="https://en.wikipedia.org/wiki/Game_of_Thrones">Wikipedia </a>  
  <div class="dropDownContent">  
    <a href="https://en.wikipedia.org/wiki/Game_of_Thrones_(season_1)">season 1</a>  
    <a href="https://en.wikipedia.org/wiki/Game_of_Thrones_(season_2)">season 2</a>  
    <a href="https://en.wikipedia.org/wiki/Game_of_Thrones_(season_3)">season 3</a>  
    <a href="https://en.wikipedia.org/wiki/Game_of_Thrones_(season_4)">season 4</a>  
    <a href="https://en.wikipedia.org/wiki/Game_of_Thrones_(season_5)">season 5</a>  
  </div>  
</li>
```

<a href="#">Main Page</a>	<a href="#">Daenerys</a>	<a href="#">Jon Snow</a>	<a href="#">Arya</a>	<a href="#">Ned Stark</a>	<a href="#">Wikipedia</a>
					<a href="#">season 1</a>
					<a href="#">season 2</a>
					<a href="#">season 3</a>
					<a href="#">season 4</a>
					<a href="#">season 5</a>

- קיבלנו:

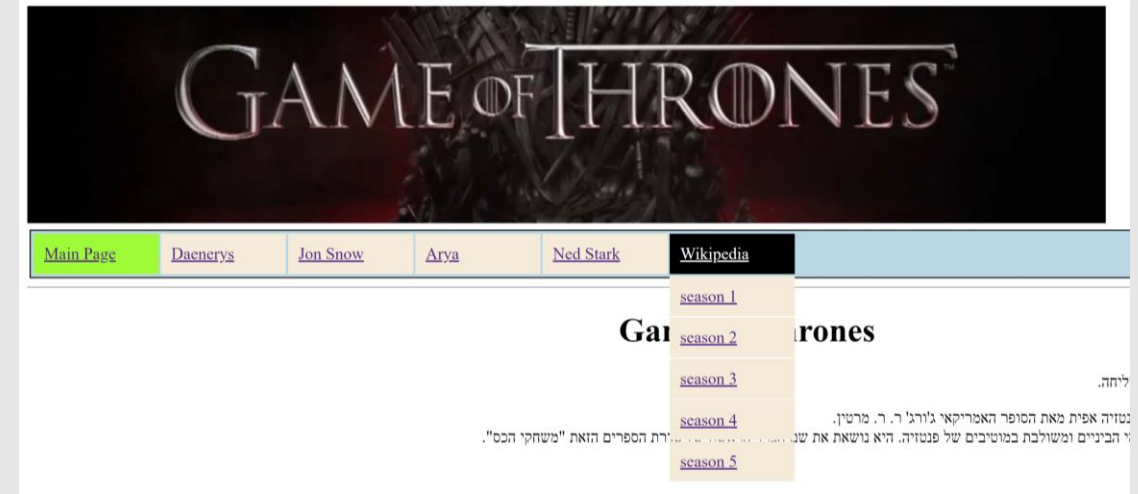
- על מנת להפריד את רשימת המשנה מהרקע של הרשימה הראשית נעצב את רשימת המשנה (הגדרנו `class=dropDown` ו-`class=dropDownContent`)
  - נוסף הוראה: `.position: absolute`.
  - נקבע כי הרשימה הנפתחת היא נסתרת במצב הרגיל: `.display: none`.
  - נקבע כי הרשימה הנפתחת תופיע מעל האלמנטים שמתחתיה (תסתיר אותם):  
`.z-index: 1`

```
.dropDownContent {  
  display: none; /*hide the sub list if not hover*/  
  position: absolute; /*position outside the main list*/  
  width: 100px;  
  z-index: 1; /*put the list in front any element*/  
}
```

- נהפוך את תת הרשימה לגלויה בעת הצבעה עם העכבר:

```
.dropDown: hover .dropDownContent {  
    display: block;  
}
```

- ובכך סיימנו.





# Dev Tools – F12

```
3 <head>
4   <meta charset="utf-8" />
5   <style>
6     h1 {
7       background-color:lightcoral;
8     }
9     #line2 {
10      width: 200px;
11      height: 300px;
12      background-color:lightblue;
13      margin-left: 100px;
14      margin-top: 40px;
15      padding: 20px;
16    }
17  </style>
18 </head>
19 <body>
20   <h1>First Line</h1>
21   <h1 id="line2">Second Line</h1>
22   <h1>Third Line</h1>
23 </body>
24 </html>
```

- נדגים את השימוש בכלי הפיתוח של כרום באמצעות הקוד הבא.

- נריץ את הקוד ובכרום נלחץ על F-12 או באמצעות Ctrl-Shift-i.

- יפתח לנו חלון הפיתוח ובו מידע על העיצוב שעשינו.

- לחיצה על התגית בקוד תציג לנו נתוני עיצוב של אותה תגית.



- הקדמה – מודל שרת לקוח, סביבת העבודה, מקורות ומידע.
- HTML – שפת תגיות, מבנה הדף, פקודות בסיסית: `<p>`, `<h1>`, `<a>`, `<img>`.
- CSS – עיצוב מדורג, שלושת שיטות העיצוב: `Inline`, `Internal`, `External`.
- טבלאות.
- טפסים.
- תפריטים ורשימות – מודל הקופסה.
- שימוש ב `F12` – `Dev Tools`.