**טכנולוגיות אינטרנט – סיכום:**

# Presentation 1 – HTML – Hyper Text Markup Language:

## **נקודות חשובות:**

### HTML:

* אלמנט (Element) – מורכב מתגית פתיחה, תגית סיום ומה שבניהם.
* שפה זו ה לא Case Sensitive וסבלנית לשגיאות
* בשפה זו ישנה התעלמות מרווחים מיותרים.
* לא מתחייב שלכל אלמנט שקיימת עבורו תגית פתיחה תהיה תגית סגירה.
* לכל תגית פתיחה ניתן להצמיד תכונה (Attribute) הפועלת בעת הפעלת התגית.
* לא ניתן להגדיר תכונות (Attributes) בתגית הסגירה – רק בתגית הפתיחה.

### Forms:

* בכל דף HTML ניתן להכניס כמה טפסים שרוצים.
* בכל פעם יכול להישלח אך ורק טופס אחד.
* כל כפתור אשר נמצא בתוך הטופס משמש ככפתור Submit.
* Form הוא הדרך היחידה של הדפדפן לקבל נתונים מהלקוח.

### עיצובים ו-CSS:

* במידה ומגדירים עיצוב בכמה מקומות (למשל כמה קבצי CSS, דרך ה-Head או בתוך התגית עצמה) – הדפדפן תמיד יבחר את העיצוב הספציפי ביותר, למשל:
  + העיצוב של התגית עצמה למרות שהוגדר לקחת את העיצוב מקובץ CSS חיצוני.
  + העיצוב שב-Head למרות שהוגדר לקחת את העיצוב מקובץ CSS חיצוני.
  + העיצוב שבקובץ ה-CSS האחרון שקושר לדף ה-HTML למרות שיש גם הגדרות עיצוב ב-CSS האחרים.
* יש לשים לב לכתיבת הסלקטור המתאים:
  + div – מתייחס לכל האלמנטים שהם div.
  + #div – מתייחס לאלמנט שה-id שלו הוא div.
  + .div – מתייחס למחלקה שהשם שלה הוא div.

### תמונה:

* הפורמטים הנפוצים:
  + GIF (Graphic Interchange Format) – פורמט דחיסה הטוב לתמונות שבהן יש כמות גדולה של פיקסלים באותו הצבע.
  + JPG (Joint Photographer Export Group) – פורמט דחיסה העובד על הגלים של הצבעים, יעיל לדחיסת תמונות שהם צילומים.
  + (ניתן גם להשתמש בפורמט PNG, אך הוא פחות נפוץ כיום).

## **סיכום מצגת:**

* רשת האינטרנט (WWW – World Wide Web) היא רשת של מסמכים מקושרים על כל המחשבים בעולם.
* כל המחשבים הפועלים ברשת האינטרנט משתמשים בפרוטוקול תקשורת HTTP (Hyper Text Transfer Protocol).

### העברת מידע:

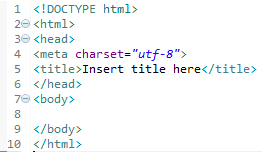
* המידע האינטרנטי (Web Information) שמור במסמכים הנקראים דפי אינטרנט (Web Pages) והם שמורים בשרתי אינטרנט (Web Servers).
* מחשבים הקוראים את דפי האינטרנט נקראים לקוחות אינטרנט (Web Clients) והם עושים זאת באמצעות תכנה הנקראת דפדפן אינטרנט (Web Browser).
* 

### HTML – הגדרה:

* HTML היא הסטנדרט והשפה הבסיסית לדפי אינטרנט.
* זוהי אינה שפת תכנות, אלא שפה המבוססת תגיות – מילים מיוחדות (tags) המציינות היכן טקסטים ואלמנטים אחרים יופיעו בדף האינטרנט המוצג בדפדפן האינטרנט.

### כללים בסיסיים:

#### מבנה תכנית בסיסית:

* 
* בכדי לכתוב בעברית – חובה להחליף את charset ל- utf-8.
* תגית head – מכילה את כל המידע על המידע (Meta Data).
* תגית title – מכילה את השם דף ה-HTML כפי שרואים בהרצת הדף בדפדפן:
  + 
* תגית body – מכילה את כל התגיות הנראות לעין בעת הרצת דף ה-HTML בדפדפן.

#### פקודות בסיסיות:

##### כותרת ראשית:

* 
* התכונה (Attribute) align מאפשרת לקבוע את העימוד של הכותרת (ימין, מרכז, שמאל).

##### קו מפריד:

* 
* התכונה width מאפשרת לקבוע את רוחב הקו.

##### פסקה:

##### ירידת שורה:

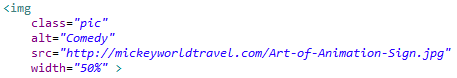
קישור: <a href="*url*">*link text*</a>

* <a href="html\_images.asp">HTML Images</a>
* <a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>

###### קישור בתוך דף ה-HTML:

* + יצירת מקום הגעה:
    - 
  + יצירת הקישור עצמו:
    - 

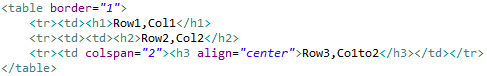
##### תמונה:

* 
* Class (אופציונלי) – המחלקה אליה שייכת התמונה (בדרך כלל משמש לעיצוב זהה למספר תמונות).
* alt (אופציונלי) – מה שיוצג במידה ולא ניתן להציג את התמונה.
* src – מקור התמונה.
  + מקור התמונה יכול להיות:
    - מהאינטרנט.
    - מקובץ בתיקיית הפרויקט.
  + חובה להשתמש במיקום יחסי – כך יהיה ניתן להעביר את הפרויקט ביתר קלות בין המחשבים (כל הרעיון שעומד מאחורי HTML).
* width - רוחב התמונה (אפשרי לרשום את כמות הפיקסלים שאנו רוצים שהתמונה תוצג בהם, אך מומלץ לרשום את כמות הפיקסלים באחוזים בכדי להציג בצורה נכונה על כל סוגי המסכים).

##### רשימה:

* 
* כל שורה אינה מחויבת בתגית סגירה, אך מומלץ לעשות כן.

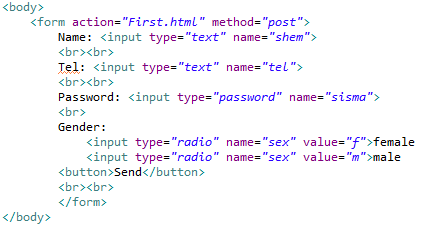
##### טבלה:

* 
* tr - רשומה חדשה בטבלה (לא חייב תגית סגירה, אך מומלץ לעשות כן).
* td - עמודה ברשומה (לא חייב תגית סגירה, אך מומלץ לעשות כן).
* border - בכדי לעשות קו תוחם לטבלה.
* colspan – ממזג בין שדות בטבלה.

##### Frame:

* היכולת להציג מספר דפי HTML בו זמנית על המסך כאשר לכל דף מוקצה חלק מסוים מהמסך.
* שיטה ישנה (HTML 4):
* 
* שיטה עדכנית (HTML 5):
* 

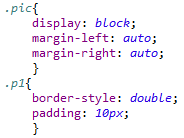
##### Form:

* 
* Action – מהו הדף הבא אליו נעבור לאחר דף זה.
* method="post" – הסתרת הנתונים שהוזנו בטופס מכתובת ה-URL של הדף הבא.
  + כאשר רוצים להציג את הנתונים שהוזנו בטופס בכתובת ה-URL של הדף הבא, משתמשים בשיטת - method="get".
* כאשר רוצה שרק אחד מה-Check Boxes יבחר בעת הזנת הטופס ע"י המשתמש – נותנים להם את אותו שם (דוגמת Gender).

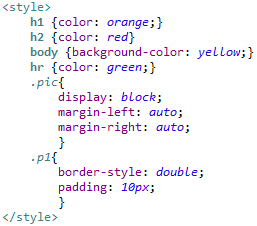
##### ישויות (Entities):

* בדיוק כמו ב-Java, ישנם מספר תווים אשר מציינים ביטויים שמורים (כמו הסימן "<") ובכדי להציגם למסך, יש להשתמש בביטויים אחרים.
  + &nbsp; – התעלמות מרווחים.
  + &lt; – קטן מ...
  + &gt; – גדול מ...
  + &amp; – סימן חיבור (אמפרסנד).
  + &quot; – מרכאות.

##### מחלקות:

* יצירת כמה מאפיינים אחידים לעיצוב אחיד.
* 

##### עיצוב:

* ישנם ארבעה מקומות בהם ניתן להגדיר עיצוב:
  + בתוך התגית עצמה – שימוש בתכונת style.
    - <p style="border-left-style: double; padding-left: 10px;">
  + בין התגית הפותחת לתגית הסוגרת של head בתוך תגית style.
    - 
    - עיצוב אחיד לכל הכותרות, פסקאות, תמונות.
  + CSS (Cascading Style Sheet):
    - קובץ חיצוני המאפשר למפתחים לעצב מספר רב של דפי HTML בפעם אחת.
    - בתוך תגית head חייבת להופיע שורת הקישור לקובץ:
      * <link rel="stylesheet" type="text/css" href="definition.css">
  + עיצוב ברירת מחדל (By Default):
    - לא מגדירים שום עיצוב ספציפי.

# Presentation 2 – Java Script Basics:

## **נקודות חשובות:**

### כללי:

* JS הינה שפת מונחית עצמים (Object Oriented):
  + Java היא שפה מונחית עצמים המבוססת על מחלקות (Class) – לא ניתן לעבוד ללא מחלקות ורק בניהן יש ירושה.
  + JS היא שפה מונחית עצמים המבוססת על אב טיפוס (Prototype) – לא ניתן להגדיר מחלקות, הירושה היא אך ורק בין אובייקטים.
* ניתן להגדיר מחלקות – שימוש ב-Constructor.
* JS היא אינה תת שפה של Java.
* שפה זו רגישה להבדל בין אותיות קטנות לגדולות (כן Case Sensitive).
* בשפה זו אין התעלמות מרווחים מיותרים – יכול להיות שקוד מושלם לא יעבוד בגלל רווחים מיותרים.
* בשפה זו לכל משתנה יש תמיד סוג טיפוס, אך הוא לא מוגדר מראש וניתן לשנות אותו במהלך ריצת התכנית.
* דרך JS ניתן להגיע לכל אלמנט של HTML וזאת בתנאי שהוגדרה לו תכונת id.

### פונקציות:

* הערכים לפונקציות ב-JS מועברים באמצעות ערך (by value) ולא באמצעות הקשר (by reference) כמו ב-Java.

### מערכים:

* יכול להיווצר מצב שבו יהיו תאים ריקים ("חורים") במערך (ראה הגדרת מערך 3) – תאים אלה יהיו מלאים ב-null.
* טיפוסי המשתנים בכל תא במערך יכולים להיות שונים זה מזה.
* בניגוד ל-Java, התאים במערך אינם יושבים בזיכרון בצורה רציפה, אלא בתור רשימה מקושרת.
* ניתן לשנות את גודל המערך באופן דינמי.

### DOM:

* DOM מחלק את דף ה-HTML לאובייקטים עליהם אפשר לערוך מניפולציות ובכך מקל על עבודת המפתח.

## **סיכום מצגת:**

### Java Script – הגדרה:

* שפת תכנות השולטת על סביבת הדפדפן והתוכן.
* מאפשרת מניפולציות על אובייקטים בתוך דף ה-HTML.
* מאפשרת שינוי/ הכללה באופן דינמי של דפי HTML.
* הקוד בדרך כלל נכתב ישירות בתוך דף ה-HTML.

### חיבור בין HTML לבין JS:

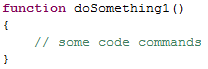
* כאשר רוצים להתחיל לכתוב ב-JS, ניתן לכתוב בכל מקום בקובץ ה-HTML את התגית <script> ובתוכה לכתוב את הקוד ב-JS.
  + הבהרה:
    - בתוך התגית <script> לא ניתן לרשום פקודות ב-HTML – בתוך התגית רשומות רק פקודות ב-JS.
* Onclick – מילה שמורה המקשרת בין קוד ב-HTML לבין קוד ב-JS.

### טיפוסי נתונים:

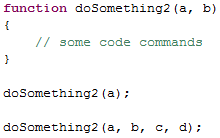
* Numbers – כולל את כל סוגי המספרים, מספרים שלמים ומספרים ממשיים.
* Boolean – בוליאני – True/ False.
* String – מחרוזת.
* Null – תא ריק.
* Undefined – נוצר רק המצביע למשתנה, לא התבצעה הגדרה.
  + דגשים:
    - אין צורך להגדיר את טיפוס המשתנה – הוא מוגדר באופן אוטומטי בעת השמת ערך לתוך המשתנה.
    - במהלך הרצת הקוד – יתכן מצב ובו משתנה מסוים יכיל (לא בו זמנית) ערכים שונים מטיפוסים שונים (משום שטיפוס המשתנה מוגדר ע"י הערך השמור בו).

### פונקציות:

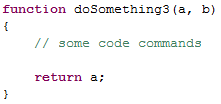
#### מבנה כללי של פונקציה ב-JS:

* 
* השוואה ל-Java:
  + אין צורך להגדיר כימוס – public, private.
  + אין צורך להצהיר על הערך המוחזר (בניגוד ל-Java) – void, int, boolean.

#### קריאה לפונקציה:

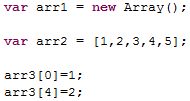
* 
* לא תתקבל תוצאת שגיאה עבור שתי קריאות אלה:
  + קריאה ראשונה – ישלים את ערך הפרמטר החסר ע"י ערך ברירת מחדל של המערכת.
  + קריאה שנייה – יתעלם מהפרמטרים העודפים.

#### החזרה מפונקציה:

* 
* גם ב-JS ניתן להחזיר ערך לקורא לפונקציה בסיום או לסיים את הפונקציה בצורה מסודרת.

### מערכים:

#### דרכים להגדרת מערך:

* 

#### פונקציות מובנות של מערכים:

* Join – משרשר את כל התאים במערך למחרוזת אחת.
* Reverse – הופך את סדר התאים במערך כך שהתא הראשון הופך לאחרון וכן הלאה.
* Sort – מיון המערך (עפ"י רצון המשתמש).

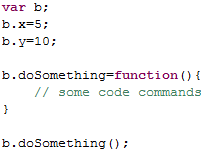
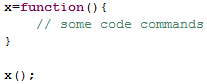
### תמונות:

#### גישה לתמונות ב-JS:

* בכדי לשנות תצוגת תמונה על המסך – חייבים להגדיר תכונת id בתוך תגית <img>
* 

### אובייקטים:

#### יצירת אובייקטים ומתודות:

* JS היא שפה חופשית וניתן להגדיר אובייקטים ומתודות במספר דרכים:
  + פשוט ליצור אובייקט באמצעות שימוש ב-**new**:
    - a=**new** Object();
  + יצירת אובייקט ריק, יצירת תכונות ומתודות ע"י השמת ערך בתוכן:
    - 
  + יצירת פונקציות והפעלתן ע"י השמתן בתוך משתנה:
    - 

### DOM (Domain Object Model) – הגדרה:

* DOM מתאר מסמך המגדיר את הגישה לאובייקטים הבאים מתוך JS:
  + Window.
  + Document.
  + Location.
  + History.

# Presentation 3 – DHTML Basics:

## **נקודות חשובות:**

* + הכוח של div – שינוי התוכן והעיצוב של ה-div באמצעות innerHTML.
  + הכוח של span – שינוי התוכן והעיצוב של שורה או תו.
  + DHTML היא יכולת, לא שפה חדשה.

## **סיכום מצגת:**

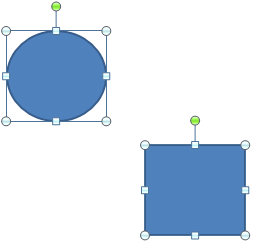
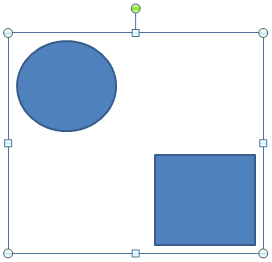
### DHTML (Dynamic HTML) – הגדרה:

* DHTML היא שיטה ליצירת דפי HTML אינטראקטיביים ע"י שילוב של:
  + Static Markup Language – HTML.
  + Client Side Scripting Language – Java Script.
  + Presentation Definition Language - Cascading Style Sheets (CSS).
  + Document Object Model – DOM.

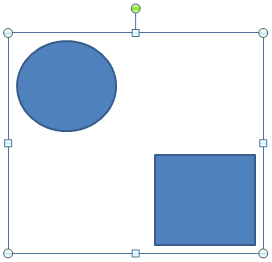
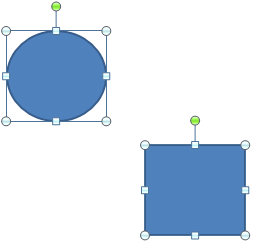
### שיטות שימוש עיקריות:

שימוש בשתי תגיות:

#### <div> – מקבץ:

* לוקח את כל מה שנמצא בין תגית הפתיחה לתגית הסגירה ויוצר מהם מקשה אחת.
* לפני הגדרת ה-div:
  + 
* אחרי הגדרת ה-div:
  + 

#### <span> – מפזר:

* לוקח את השלם ומוציא ממנו חתיכה.
* לפני הגדרת ה-span:
  + 
* אחרי הגדרת ה-span:
  + 

לאחר היצירה של התגיות הללו – ניתן להגיע אל המקומות הללו דרך JS ולשנות את הפריטים והעיצובים בדף ה-HTML.

# Presentation 4 – jQuery Essentials:

### jQuery – הגדרה:

* ספרייה של HTML המכילה פונקציות מובנות שבאות לעזור למפתח לבצע את עבודתו.

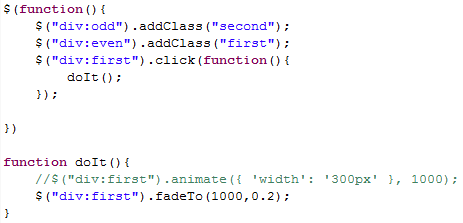
### קישור לספריית jQuery:

* <script src="jquery-2.2.2.min.js"></script>
* ניתן גם לקשר לינק לאתר האינטרנט וממנו לשאוב ספרייה זו.

### שימוש בסיסי ב-jQuery בתוך JS:

* $("div").html("Hello").addClass("good");
* $("div") – בתוך הסוגריים של סימן הדולר יש לרשום סלקטור (selector) אליו אנו רוצים להתייחס.
  + כלומר, בקטע קוד זה נלקחים כל האלמנטים אשר מתאימים לסלקטור הרשום במחרוזת, מושמים במערך ומוחזר אובייקט המייצג את כל תאי המערך וכל שינוי עליו משנה את כל התאים במערך כאשר כל תכונות האלמנט מומרים לתכונות האובייקט.
* .html("Hello") – אחר כך רושמים נקודה, את שם הפונקציה ואת הפרמטרים שלה.
* .addClass("good"); - ניתן לשרשר כמה פונקציות שרק רוצים משום שכל תוצאת פונקציה היא אובייקט חדש (בדיוק כמו ב-Java).

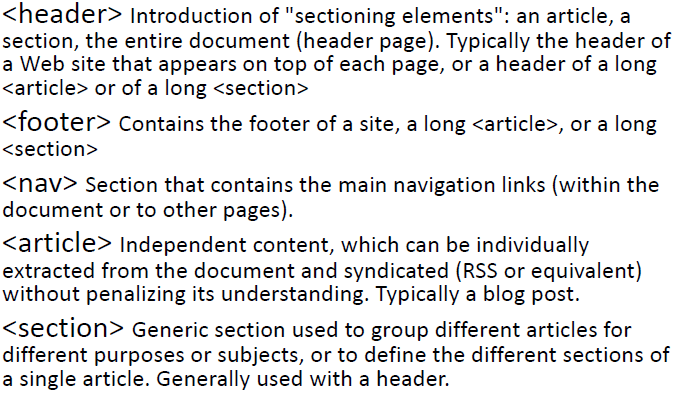
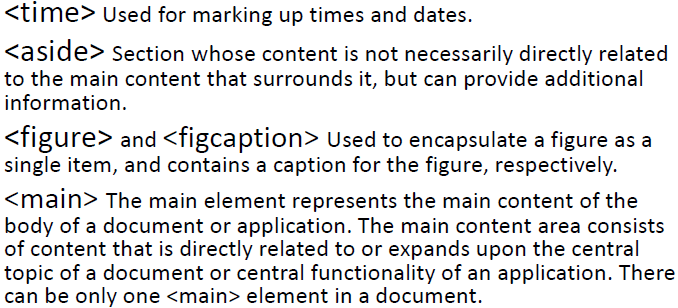
### שימוש קצת יותר מתקדם ב-jQuery בתוך JS:

* ניתן לכתוב פונקציה ב-JS אשר תהווה פרמטר לפונקציה ב-jQuery.
* 

### הבדלי Set ו-Get:

* ההבדל היחידי בין מתודות Set ו-Get הוא בפרמטרים שהם מקבלים בקריאה לפונקציה – חשוב מאוד לשים לב מה שולחים.
* 

# Presentation 5 – HTML5:

* נשאר סלחני לשגיאות – הפכו שגיאות נפוצות מ-HTML 4 לתקניות ב-HTML 5.
  + כמות גדולה של אלמנטים כבר לא מחויבים בתגית סגירה:
    - </li>, </dt>, </dd>,</tr>, </th>, </td>, </thead>, </tfoot>, </tbody>, </option>,</optgroup>, </p> (in most cases), </head>, </body> and </html>
* הוסיפו כמות גדולה של אלמנטים – גישה ל-GPS, למצלמה, למיקרופון.
* שינוי סמנטי של מבנה דף ה-HTML:
  + 
  + 
  + 
  + 

# Presentation 6 – Ajax-JSON:

## **נקודות חשובות:**

* Ajax מעלה את רמת האינטראקטיביות של דף האינטרנט מול משתמש הקצה.
* בניגוד ל-DHTML הלקוח יכול לתקשר עם השרת בלי צורך לטעון מחדש את כל דף האינטרנט.
* Ajax מאפשר לנהל תהליכי עיבוד על מחשב הלקוח (ב-JS) עם מידע שנלקח מתוך השרת.
* Ajax חוסך ברוחב הפס – נשלחים ומתקבלים רק פריטים שחשובים להמשך הפעילות בלי נתונים מיותרים.
* נקרא מודל א-סינכרוני – הלקוח יכול להמשיך לעבוד לפני שהוא מקבל תשובה מהשרת.

## **סיכום מהמצגת:**

### Ajax (Asynchronous JavaScript and XML) – הגדרה:

* Ajax היא בעצם דרך לשלוח/ לקבל מידע מהשרת באמצעות קוד JS.
* Ajax היא לא טכנולוגיה בעצמה.
* Ajax היא "שיטה לפיתוח אתרים" (Web Development Technique) המשלבת מספר טכנולוגיות קיימות בדרך מאוד ספציפית:
  + HTML.
  + JS.
  + XMLHttpRequest.
  + XML (אופציונלי).

### דוגמאות מחיי היומיום:

* הצעות חיפושים נפוצים ב-Google כאשר מקלידים אותיות.
* ריענון מפה ב-Google Maps כאשר רוצים לראות את המפה בדרך מפורטת יותר/ פחות.

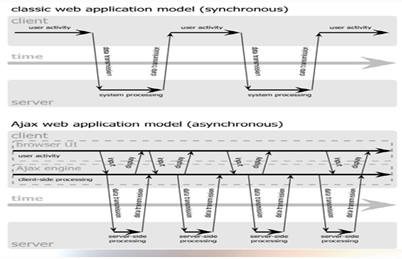
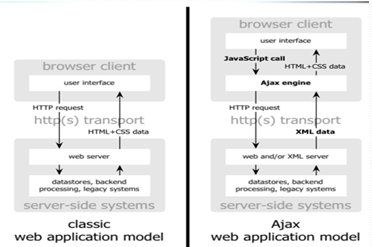
### יתרונות של Ajax:

* אפליקציות יותר מגיבות למשתמש הקצה.
* הורדת עומס מהשרת – רק החלק שברצוננו לשנות משתנה וזאת בלי לרענן את כל הדף.

### חסרונות של Ajax:

* אין תמיכה בכפתור back – כי הדף לא נטען מחדש.
* אין אפשרות להגיע למידע דרך מנועי החיפוש – לא נחשב חלק מדף האינטרנט.
* כתובת ה-URL אינה משתנה כאשר חל שינוי בדף – לכן אי אפשר לאתרו ע"י מנוע חיפוש.
* במצב א-סינכרוני – יכול להיות delay בגלל תהליך העיבוד בשרת – יכול קצת להפריע למשתמש הקצה.

### איך זה עובד:

* תוכן הדף משתנה ע"י DOM InnerHTML.
* מידע המתקבל מהשרת משתמש בפונקציית XMLHttpRequest של JS.
  + השוואת ביצועים:
    - 
  + ארכיטקטורת המערכת:
    - 

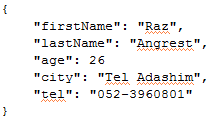
**יש עוד דוגמאות קוד וסתם חפירות, אני מדלג על זה ולא שם בסיכום.**

### JSON (JavaScript Object Notation) – הגדרה:

* JSON הוא פורמט העברת נתונים פשוט ונוח המקל על אנשים לקרוא ולכתוב.
* JSON מבוסס על תת קבוצה של JS.
* JSON הוא פורמט טקסט שהוא לגמרי תלוי שפת תכנות, אבל משתמש במוסכמות של המתכנתים בכל השפות ממשפחת שפת C.
* התכונות הללו הופכות את JSON לכלי היעיל ביותר להחלפת מידע בין האפליקציות.

### Syntax of JSON:

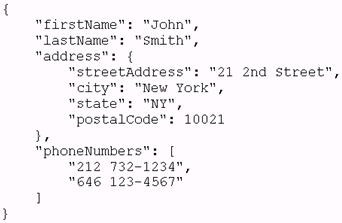
#### משתמשים ב-{} להגדרת אובייקט:

* 

#### משתמשים ב-[] להגדרת מערך:

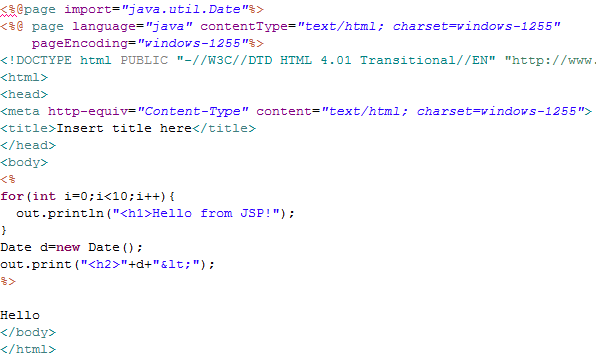
* 

#### ואפשר גם לשלב בניהם:

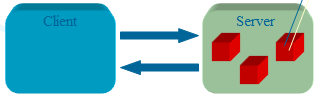
* 

# Presentation 7 – JSP Basics:

### JSP (Java Server Pages – הגדרה:

* JSP מדבר כבר על צד שרת, זהו בעצם תכנות של השרת.
* JSP נוצרים על בסיס תחביר של HTML (או XML).
* בקוד ה-HTML משולבים גם קטעי קוד ב-Java.
* לדוגמה:
* 

### Servlet – הגדרה:

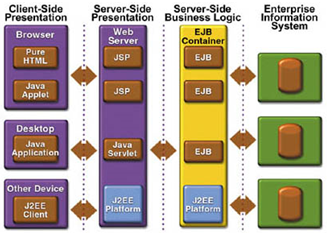
* Servlet הינו מחלקה ב-Java המיישמת ממשק (Interface) ספציפי.
* Servlet בעצם ממיר את הקוד שנכתב ב-Java בשרת לקוד HTML לצורך שליחתו למחשב הלקוח.
* 

### יתרונות ה-Servlet:

* יעילות – כל ה-Servlets מופעלים באותו תהליך (בשרשורים שונים כמובן).
* נוחיות – API פשוט הקל לשימוש לכל מתכנתי Java.
* עוצמה – גישה לשרתי אינטרנט חזקים ולטכנולוגיות מתקדמות.
* ניידות – השלמת ניידות בפלטפורמות ושרתים.
* אבטחה – מנגנון האבטחה של Java עם אימות מיוחד ומתודות אישור.
* מחיר נמוך – אפשרי להשתמש בשרתים חינמיים ושרותי אחסון Servlet.

# Presentation 9 – Web Applications:

### J2EE (Java 2 Enterprise Edition) – הגדרה:

* מגדיר קבוצה של API בתכנות צד שרת המעוצבים כדי לספק תשתית תכנתית הניתנת לשינוי והמתאימה לארגונים בינוניים וגדולים.
* Servlets ו-Java Server Pages הם חלק מ-J2EE.
* המרכיבים הללו בדרך כלל משמשים את ה-Presentation Layer של J2EE Web Application.
* המרכיבים יכולים לשמש גם את ה-Business Logic Layer.
* 

### Components, Containers and Deployment:

#### Components:

* יחידת תכנה בשכבת ה-Application.
* כולל:
  + Applets.
  + Application Clients.
  + Enterprise JavaBeans Components (EJC).
  + Web Components (Servlets and JSPs).

#### Containers:

* הם סביבות ריצה סטנדרטיות המספקות שירותי Component ספציפיים.
* ה-Components יכולים לספק את השירותים ולהיות זמינים בכל פלטפורמה של J2EE.
* הפעולה של התקנת J2EE Components בתוך ה-Container נקראת "Deployment".

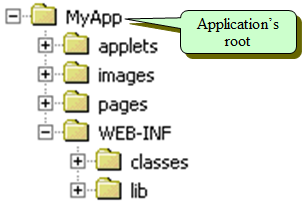
#### Deployment:

* תהליך העברת האפליקציה מסביבת פיתוח לסביבת שימוש.
* Containers מספקים מנגנון לבחירת התנהגות Application בזמן ההתקנה או הפריסה.
* שימוש בקבצי XML מיוחדים הנקראים Deployment Descriptors (DD), Components יכולים להיות מוגדרים עבור סביבה ספציפית של Containers בהעברה לסביבת השימוש.
* יכול להימצא בארבע מקומות:
  + Web Application – web.xml.
  + EJB Models – ejb-jar.xml.
  + J2EE Clients – application-client.xml.
  + J2EE Application – application.xml.

### Web Application – הגדרה:

* אוסף של Servlets, JSPs, דפי HTML, Classes ומקורות אחרים המריצים את האפליקציה על Web Server.
* ה-Web Application יכול להיות מארז ורץ על Containers רבים מהרבה ספקים.
* שני היבטים חשובים של J2EE Web Application:
  + תמיכה בהרבה Containers (כלומר Servers).
  + פלטפורמות מרובות.
* מרכיבים:
  + Servlets.
  + Java Server Pages (JSP).
  + Utility and Helper Classes.
  + Static resources (HTML pages, images, multimedia files, etc.).
  + Client side components (Applets, Beans, other classes).
  + Descriptive meta-information files (DD, TLD, etc.).

### מבנה התיקיות:

* Web Application נמצא במבנה היררכי של תיקיות, כאשר כל מיקום של אלמנט מוגדר היטב.
* לכל Web Application יש Root Directory משלה.
* התיקייה הזאת מהווה בסיס לנתיב היחסי של כל ה-Web Application.
* דוגמה:
  + 
* ניתן להשתמש בכל תיקייה שנמצאת מתחת ל-Root Directory.

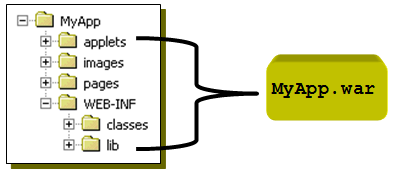
#### WEB-INF:

* משתמשי ה-Web Application לא מורשים לגשת לתיקייה זו.
* מכילה שתי תתי תיקיות:
  + Classes – מכילה מחלקות שאינן מאוגדות, זמינה ל-Class Loader של ה-Web Application.
  + Lib – מכילה קובצי jar מאוגדים, הכוללים מחלקות שיכולים להיטען ע"י Servlets Container's Class Loader.

### Deployment Web Application:

* Deployment יכול להיעשות בשתי דרכים:
  + העתקת ה-Root Directory לתיקייה המוכרת ל-Container.
  + שימוש מיוחד בקובץ Archive.

### Web Application Archive (WAR):

* ה-Web Application יכול להיארז לתוך קובץ Web Archive (WAR) תוך שימוש ב-Java Archive סטנדרטי.
* קובצי WAR יכולים להיות מיוצאים ע"י שימוש בכלי למציאת Deployment.
* כאשר מאגדים אפליקציה, מבנה התיקיות מתחת ל-Root Directory מאוחסן.
  + 
* ה-Deployment של קובץ WAR אחד (ולא של עץ תיקיות שלם) הוא ישיר וקל.
* אחרי Deployment, ASP Container מסיר את הקובץ או מבצע אותו בפורמט הדחוס.

### The web.xml File:

* מתאר ה-Deployment של ה-Web Application הוא קובץ XML הנקרא web.xml.
* הקובץ ה-web.xml חייב להיות ממוקם תחת תיקיית WEB-INF.
* קובץ ה-web.xml הוא קובץ XML תקין המקבל מידע תקין.
* המידע שבקובץ כולל:
  + Servlets / JSP definitions.
  + Servlets / JSP mapping.
  + Session configuration.
  + Application Lifecyle Listener classes.
  + Filter Definitions and Filter Mappings.
  + Error Pages.
  + Security settings.
  + And more.

## **סיכום:**

* J2EE Web Application תומך בגישה של "Write once – Run on any server and platform".
* מרכיבי ה-Web Application שמורים במבנה תיקיות מוגדר.
* Web Application יכולים להיות מאוגדים לתוך קובץ war יחיד לצורך Deployment.
* Web.xml הוא מתאר ה-Deployment של ה-Web Application ומכיל בתוכו את כל הגדרות האפליקציה הנדרשות.

# Presentation 10 – Cookies and Session Management:

* **בהרצאת ההשלמה אמר שלא יהיו שאלות על Cookies.**

## **נקודות חשובות:**

* <% %> – מתורגם לקוד לוקאלי, לא ניתן להגדיר כימוס למשתנים (public, private).
* <%! %> – מתורגם לקוד גלובלי, ניתן להגדיר כימוס למשתנים (public, private).
* אובייקט Request – מאפשר לנהל את כל המידע שנשלח ע"י הלקוח.
* אובייקט Response – מאפשר לנהל את כל המידע שנשלח כתגובה ע"י השרת.
* ב-JSP – ניתן להפעיל מחלקות מובנות של Java (צריך כמובן לייבא אותן קודם).
* Session מתגבר על כל הבעיות של Cookies באמצעות שימוש באובייקטים של Java.

## **סיכום מצגת:**

### Session – הגדרה:

* Session הינו הנוכחות של משתמש יחיד בתוך האפליקציה.
* כל משתמש שנכנס לאפליקציה פותח Session חדש, אשר תלוי במשך העבודה של המשתמש במערכת.

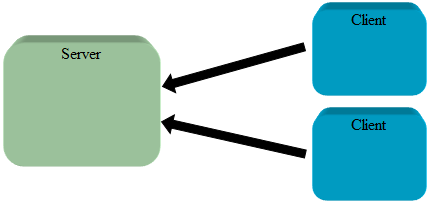
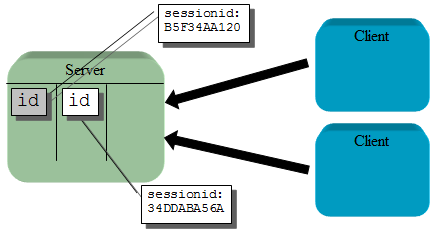
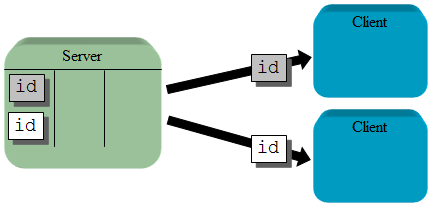
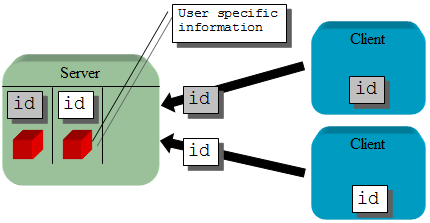
### שימושים נפוצים:

* כל אפליקציה הדורשת תהליך התחברות והזדהות.
* אפליקציות המשרתות את הלקוחות בצורה שונה, למשל: בהתחברות לאתר הבנק רשום שם בעל החשבון בחלקו העליון של המסך.
* אפליקציות לקניית שירות.

### יצירת Session:

* Session נוצר לראשונה עבור לקוח בפנייתו הראשונה לשרת.
* השרת מניח שמדובר בלקוח חדש אם הבקשה אינה כוללת מידע על Session ID ויוצר Session חדש עבור אותו לקוח.
* החל מהבקשה השנייה והלאה, הלקוח נהיה חלק מה-Session הנוכחי וממשיך כל עוד יש לו Session ID חוקי.

### Session Management:

* הלקוח שולח בקשה ראשית לשרת:
  + 
* כל חיבור מקבל מזהה ייחודי (בדרך כלל נקרא Session ID) כשה-Session נוצר (לכל לקוח יש Session ID משלו שהוא מספר אקראי ארוך):
  + 
* ה-Session ID נשלח ללקוח ומוחזר עם כל הקשות העוקבות:
  + 
* השרת יכול להשתמש ב-Session ID כדי לעקוב אחר הבקשות ולקשר אחסון מידע למשתמש ספציפי על השרת:
  + 

### שיטות מעקב אחר Session:

* שימוש ב-Cookies.
* כתיבה מחדש של URL.
* שימוש ב-Secure Socket Layer (SSL).
* **לא נראה לי שחשוב לפרט על כל שיטת מעקב (לא זוכר שדיברנו על זה בכלל בהרצאות).**

### שיטות לסגירת Session:

#### שימוש ב-Time Out:

* השרת או מנהל האפליקציה קובעים פרק זמן מסוים שלאחריו נסגר ה-Session באופן אוטומטי.
* לרוב הזמן מוגדר ל-15-20 דקות.

#### שימוש ב-Log Out:

* הלקוח מבצע סגירה יזומה של ה-Session ע"י פונקציית Log Out.

# Presentation 11 – JSP and JavaBeans:

## **נקודות חשובות:**

* JavaBeans הוא ממשק (Interface) לשימוש במרכיבי Java בשרתים שונים.
* JavaBeans הם רכיבי תכנה הניתנים לשימוש חוזר ותפקידם לכמס הרבה אובייקטים בתוך אובייקט אחד (Bean) כך שיהיה ניתן להעבירם בפעם אחת ממקום למקום בפעם אחת ולא להעביר אובייקטים בודדים.

## **סיכום מצגת:**

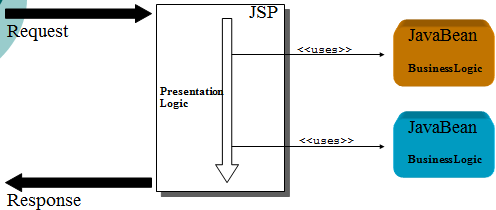
### JavaBeans – הגדרה:

* טכניקה לבניית רכיבי Java לשימוש חוזר.
* עוצבו בהתחלה לשימוש בתוך סביבת פיתוח ויזואלית.
* נפוצים בשימוש עבור כימוס של Business Logic.
* JavaBeans היא מחלקה ב-Java עם שלושה מרכיבים:
  + נתונים (פרטים של אובייקט).
  + מתודות לשימוש מחוץ למחלקה הספציפית.
  + כפיפות למרכיבי תכנות מסוימים.

### שימוש ב-JavaBeans:

* מימוש java.io.Seiralizalbe.
* מתן Constructor ברירת מחדל (ללא שום פרמטרים).
* שימוש במוסכמות השמיות למתודות Get ו-Set בשימוש עם מאפיינים.

### JavaBeans ו-JSP:

* אחד מתפקידי ה-JSP הוא להפריד בין ה-Presentation לבין ה-Business Logic.
* JavaBeans מאפשר להעביר את ה-Business Logic אליו מה-JSP ובכך לאחד בין כל האובייקטים המרכיבים את ה-Business Logic.
* 

### מתי להשתמש ב-JavaBeans:

* בכדי להעריך את קוד ה-Java שב-JSP ולהסתיר את ה-Business Logic.
* שימוש בלוגיקה מסוימת עבור מספר רב של דפי HTML.
* שיתוף מידע בין מרכיבים רבים במערכת.

# שאלות לדוגמה מההרצאות שעלולות להיות במבחן:

### האם HTML היא שפת טקסט?

* כן.

### האם HTML היא שפת תכנות?

* לא, היא שפה מבוססת תגיות (tags).

### האם תצוגת מסמך HTML מכילה רק טקסט?

* לא, באמצעות פקודות HTML ניתן גם להציג תמונות, וידאו.

### מי המפרש (Compiler) של שפת HTML?

* הדפדפן (Browser).

### לכל משתנה ב-JS תמיד קיים טיפוס נתונים?

* נכון, אבל הוא לא מתקבע עד שמוכנס לו ערך.

### האם הפקודה $("div") מחזירה מערך של אובייקטים?

* לא נכון, הפקודה מחזירה אובייקט עוטף המייצג את כל האובייקטים במערך וכל שינוי בו משנה את כל האובייקטים במערך.

### האם ניתן לכתוב Ajax בלי jQuery?

* כן.

### האם יכול להיות שרת שתומך ב-JSP ולא תומך ב-Servlet?

* לא, כי Servlet יותר חשוב מ-JSP.

### האם ניתן להגדיר משתנה מקומי בתור private?

* לא, הוא מוגדר רק ב-Servlet.

### האם ניתן להגדיר משתנה גלובלי בתור private?

* כן.

### האם מי שלא מתחבר לשרת לא פותח Session?

* לא נכון, Session נפתח בפנייה הראשונית לשרת.

# שאלות מתרגול חזרה למבחן:

איזה תגית משמשת להכנסת JavaScript לתוך דף html ?

1.< script=’java’>

2.< javascript>

**3.< script>**

4.< js>

**נכון** או לא נכון : ג'אווה סקריפט יכולה לקרוא ולעשות שימוש בתגיות HTML

**נכון** או לא נכון : javaScript הינה שפה רגישה לאותיות קטנות/גדולות

מה התחביר הנכון להתייחס לסקריפט חיצוני שנקרא formValidation.js ?

script href = “formValidation.js">.1>

2.< script source = “formValidation.js">

3.< "script name = “formValidation.js>

**4.< script src = “formValidation.js">**

נכון או **לא נכון** : ג'אווה סקריפט מתעלמת מרווחים מיותרים

הסבר:

* יכול להיות שקוד לא יעבוד בגלל רווחים מיותרים.

איזה שפה מגדירה את ההתנהגות של ה- web page ?

HTML.1

CSS.2

XML.3

**Java Script.4**

הסבר:

* HTML היא שפת תגיות סטטיות.

מהי הדרך הנכונה להגדיר מערך בג'אווה סקריפט?

1.var salaries = new Array( 1:39438, 2:39839 3:83729)

2.var salaries = new (Array:1=39438, Array:2=39839, Array:3=83729)

**3.var salaries = new Array(39438,39839,83729)**

4.var salaries = new Array() values=39438,39839,83729

מי מהמונחים הבאים מתאר את הפיכת דף האינטרנט לדף דינמי ואינטראקטיבי?

**DHTML.1**

HTML.2

XHTML.3

XML.4

נכון או **לא נכון** :לתגית <!DOCOTYPE> יש תגית סוגרת

DHTML היא שפת תגיות נכון/ **לא נכון**

DHTML הוא שילוב של: שינוי של תוכן בדף כמו שעשינו עם getelmentbyid

1)HTML

2)javascript

3)HTML Dom

4)CSS

**5) כולם**

מה הפירוש של DOM ? – ביצוע מניפולציות על דפי HTML

**Document object model.1**

Data object model.2

Document Oriented model.3

Data oriented model.4

**נכון** או לא נכון : CSS מגדיר כיצד להציג רכיבי HTML , מאפשרת למפתחים לשלוט בסגנון ובפריסה של דפי אינטרנט מרובים בבת אחת.

נכון או **לא נכון** : ג'אווה סקריפט היא תת קבוצה של ג'אווה

נכון או **לא נכון** : DHTML היא שפת טכנולוגיה מיוחדת

הסבר:

* DHTML היא יכולת.
* HTML היא בכלל לא שפת תכנות.

**נכון** או לא נכון : בג'אווה סקריפט , אפשר להעביר אובייקטים כארגומנטים לפונקציות, והוא יכול להחזיר על ידי פונקציות אובייקטים, אובייקטים לא מועברים לפי ערך.

הסבר:

* ב-JS בונים פונקציות ועושים איתם כל מיני פעולות.
* האובייקטים לא מועברים לפי ערך, אלא לפי אובייקטים.

**נכון** או לא נכון : • AJAX אינה טכנולוגיה בפני עצמה. זוהי טכניקה פיתוח אינטרנט, שילוב טכנולוגיות קיימות מספר באופן ספציפי: HTML, JavaScript, ובמיוחד XMLHttpRequest, ו (אופציונלי) XML.

הסבר:

* AJAX היא יכולת שעושים בה שימוש.

**נכון** או לא נכון : • יישומי AJAX מספקים רמה גבוהה יותר של אינטראקטיביות עם משתמשי הקצה

מדוע כדאי לעשות שימוש ב- AJAX ?

1.כדי לבנות אתר אינטרנט מהיר, דינמי

2.בכדי לחסוך במשאבים

AJAX.3 מאפשרת לבצע עיבוד במחשב הלקוח (ב-JavaScript) עם נתונים שנלקחו מהשרת.

**4.כל התשובות נכונות**

**נכון** או לא נכון : • HTML הוא הסטנדרט בדפי האינטרנט , בנוי על TAGS עם אלמנטים בתוכם

**נכון** או לא נכון : • JavaScript נכתב באופן רופף אין צורך להצהיר על סוג של משתנה כשהוא מוגדר לראשונה

נכון או **לא נכון**: ב- JavaScript המשתנה לא יכול להכיל סוגים שונים בזמנים שונים

הסבר:

* סוג המשתנה נקבע ע"י הערך שבתוכו.

האם הקוד הבא כתוב בצורה הנכונה ? **נכון**/ לא נכון

<table>

<tr>

<td> one </td>

<td> two </td>

<td> three </td>

</tr>

<tr>

<td> morning </td>

<td> noon </td>

<td> evening </td>

</tr>

</table>

**נכון** או לא נכון : AJAX מגביר את השימושיות של יישומי אינטרנט, מה שהופך אותם כביכול למהירי תגובה יעיל יותר אינטראקטיבי, ובכך משפר את חוויית המשתמש.

נכון או **לא נכון** : JavaScript היא שפה מונחית עצמים

הסבר:

* לא חייבים לבנות אובייקטים כדי לעבוד בשפה.
* שפה מונחית עצמים: על מנת שהעבודה תהיה נכונה – חייבים לבנות אובייקטים כדי לעבוד.

שאלות פתוחות:

1. מה ההבדל בין קוד HTML לקוד CSS ?
   1. HTML שפת תגיות המגדירה דפי אינטרנט ו-CSS שפת תגיות המגדירה עיצובים.
2. מי מפעיל קוד JSP ?
   1. שרת.
3. מי מפעיל קוד javascript ?
   1. דפדפן – צד לקוח.
4. כיצד ניתן לשמור נתון מסוים כך שיוצג בכל אחד מדפי הפרויקט ?
   1. Cookie או Session.
5. מה רואים כששולחים קובץ JSP ?
   1. דף HTML.

מה המבנה הנכון לעיצוב ב- CSS?

1. {body;color:black}
2.  {body:color=black(body)}
3.  body:color=black
4. **body {color: black}**

# שאלות משחזורים:

### מה נכון לגביי HTML?

* **תג <a> משמש ליצירת קשר לדף HTML אחר.**
* **לתווי רווח רבים יש השפעה ויזואלית זהה לתו רווח אחד.**
* **בתג <a> אפשר להשתמש ליצירת קשר לחלקים שונים במסמך HTML.**
* **<br> בונה שורה ריקה גם אם יש שורת טקסט אחת.**
* **לכל תג יכולה להיות תכונה (Attribute).**
* **ע"י תג נקבעת התצורה בדפדפן.**
* **מסמך HTML הוא מסמך שמכיל טקסט בלבד.**
* **הדפדפן לא מבצע פעולות לאלמנטים שהוא לא מכיר.**
* **יכול להיות אלמנט ללא מאפיינים.**
* HTML היא סוג של שפת תכנות.
* כל תג פתיחה חייב בתג סגירה מתאים.
* HTML מבוסס על XML.
* שם האלמנט מסביר את משמעות הנתונים שבתוכו.
* שם אלמנט יכול להיות כל מחרוזת שהיא.
* הדפדפן אינו מציג תוך אלמנטים שהוא לא מכיר.

### מה נכון לגבי אלמנט div?

* **אלמנט div בונה פסקה.**
* **אלמנט div אינו כולל עיצוב מיוחד כברירת מחדל.**
* **תוכן של אלמנט div אפשר לשנות בצורה דינאמית ע"י שינוי מאפיין innerHTML.**
* לאלמנט DIV יש תפקיד שונה מתפקידו של אלמנט div.
* במסמך HTML יכול להיות אלמנט div אחד בלבד.

### מה נכון לגבי אלמנט form במסמך HTML?

* **באלמנט form יכולים להיות מספר כפתורים מסוג submit.**
* **קליק על כפתור submit גורם לשליחת נתוני הטופס לשרת והעברה לדף HTML אחר.**
* **אלמנטים מסוג input מיועדים להיות בתוך אלמנט form.**
* **לכל אלמנט input נותנים שם לפיו מעבדים נתונים המתקבלים מטופס בשרת.**
* במסמך HTML אחד יכול להיות אלמנט form אחד.

### מה נכון לגביי משתנים ב-JS?

* **אם המשתנה מוגדר מחוץ לפונקציה, תחום הכרתו הוא כל המסמך כולל חלק שבתוך אלמנט body.**
* **למשתנה שמוגדר בתוך הפונקציה ללא שימוש בהגדרה var יש גישה גם מחוץ לפונקציה.**
* **למשתנה שקיבל ערך מספרי אפשר בהמשך לתת ערך של מחרוזת.**
* משתנים אפשר להגדיר רק בתוך חלק של הקוד שנמצא באלמנט head של מסמך HTML.
* למשתנים אין סוג נתונים.

### מה נכון בנוגע לתכנות מונחה עצמים ב-JS?

* **לבניית אובייקטים משתמשים באובייקט new.**
* **להגדרת מחלקה לא משתמשים במילה השמורה class.**
* **שימוש ב-json מאפשר דרך להגדרת קבוצת נתונים בצורת אובייקט.**
* **גישה לנתונים חברי מחלקה אפשר לקבל גם באמצעות אופרטור [].**
* אי אפשר להגדיר בנאי.

### מה נכון לגביי JS?

* **אפשר להגדיר משתנים מקומיים שמוכרים בתוך הפונקציה בלבד.**
* **אפשר להגדיר פונקציות בקובץ נפרד עם סיומת js.**
* **פונקציות יכולות לקבל פרמטרים.**
* פונקציות אינן יכולות להחזיר ערך.
* בתוך פונקציה חייב להיות return.

### מה נכון לגביי מערכים ב-JS?

* **משתמשים בסוגריים מרובעים [] לגישה לאלמנטים של המערך.**
* **להתחלת מערך משתמשים ב-new Array().**
* למערך יש תכונה בשם length.
* במערכים אינדקס מתחיל מ-0 וחייב להשתמש עד למספר ברצף מקסימלי.
* כל הערכים במערך חייבים להיות באותו הסוג.

### מה נכון בנוגע ל-Ajax?

* **מנגנון Ajax מאפשר תצוגה בדפדפן נתונים נוספים משרת תוך המשך עבודה במסמך המוצג.**
* **Ajax מאפשר לקבל נתונים מהשרת ללא קבלת מסמך HTML כולו.**
* **נתונים שמקבלים דרך Ajax יכולים להיות בצורת מסמך JSON.**
* מפעילים פונקציית תגובה שמנגנון Ajax הפעיל אותה רק פעם אחת בעת קבלת נתונים מהשרת.
* כפתור back של הדפדפן מאפשר לחזור לנתונים שהיות מופעלים לפני הפעלת מנגנון Ajax.
* נתונים שמקבלים דרך Ajax חייבים להיות בצורת מסמך XML.
* מגדירים פונקציית תגובה שמנגנון Ajax הפעיל אותה רק פעם אחת בעת קבלת נתונים מהשרת.
* כפתור back של הדפדפן מאפשר לחזור לנתונים שהיו מוצגים לפני הפעלת מנגנון Ajax.

### מה נכון לגביי שורת הקוד הבאה:

### Document.getElementById(a).style.backgroundColor = "red"

* **השורה תשנה צבע רקע של אלמנט אחד בלבד.**
* **תכונה style קיימת לכל אלמנט ויזואלי של מסמך HTML.**
* **Document זה אובייקט הקיים בכל מסמך HTML.**
* **a חייב להיות משתנה שערכו מוגדר לפני שורת הקוד.**
* במקום הפונקציה getElementById(a) אפשר לכתוב getelementbyId(a).

### מה נכון לגבי DHTML ו-DOM?

* **במאפיין id של האלמנט משתמשים לקבלת גישה לאלמנט כאובייקט ב-JS.**
* **לכל אלמנט של HTML אפשר לקבל גישה כאובייקט ב-JS.**
* **כל אלמנט של HTML הוא אובייקט ב-DOM, אפילו אם לא מוגדר לו id.**
* שתי מילים של טקסט במסמך של HTML הם שני אובייקטים ב-DOM.
* DHTML זו שפה נוספת ל-HTML.

### מה נכון לגבי המשפט הבא בקובץ JSP?

### <%=a.toString() %>

* **משתנה a חייב להיות מוגדר היטב לפני המשפט.**
* **המשפט יוצר מחרוזת במסמך שנשלח למשתמש הקצה.**
* המשפט שגוי מפני שאין ; בסוף.
* אפשר להחליף את המשפט בביטוי זהה ב-JS.
* המשפט יכול להיות ממקום בקובץ רק אחרי תווית <body>.

### מה נכון בנוגע לשימוש של JSP ו-Servlet?

* **אפשר לפתח יישום Web שמשתמש ב-Servlet ללא שימוש ב-JSP.**
* **שרת התומך ב-JSP חייב לתמוך ב-Servlet.**
* שרת התומך ב-Servlet חייב לתמוך ב-JSP.
* אפשר לפתח יישום Web שמשתמש ב-JSP ללא תרגום קבצי JSP.

### מה נכון לגביי שורת הקוד הבאה?

### <a href=<http://....>> link here</a>

* **אלמנט a משמש ליצירת קשר עם מסמכים אחרים.**
* **ערך href יכול להיות שם קובץ**
* **באלמנט a ניתן להשתמש כדי לקבוע קשר בין חלקים באותו מסמך.**
* ניתן להגדיר href ב-Absolute Path.
* בשורת קוד יש שגיאה כי ב-URL אין שם קובץ.

### מה נכון לגביי CSS?

* **קיימים סגנונות שיש להם מראה שונה בדפדפנים שונים.**
* **אפשר להגדיר CSS בקובץ נפרד.**
* **אפשר לקבוע מיקום בדפדפן עפ"י CSS.**
* Color מוגדר ב-style וב-attribute שניהם תקפים (כל סימן).
* אין דרך להגדיר סגנון לאות בודדת במילה גם בשימוש עם אלמנטים נוספים ב-HTML.

### מה נכון לגביי שורת הקוד הבאה?

### <IMG SRC...onclick="do.this;do.that"

* **אפשר לגשת לאלמנט דרך JS.**
* **Onclick חייב לקבל שם של פונקציה.**
* **פונקציות בשם that.do/ this.do חייבות להיות מוגדרות ב-JS.**
* Onclick חייב להיות באותיות קטנות.
* אי אפשר לשנות תמונה דרך JS.

### מה נכון לגביי jQuery?

* **בנויה כך שאין צורך בקוד שונה לדפדפן שונה.**
* **לשימוש ב-jQuery חייב להגדיר קשר לקובץ JS שמכיל קוד של jQuery.**
* **מטרת jQuery היא לאפשר כתיבת מינימום קוד לביצוע איגודים טיפוסיים עבור קבוצות אלמנטים של מסמכי HTML.**
* נוצר כדי לקשר בין JS ל-Java.
* פונקציה דולר מחזירה מערך של אובייקטים.

### מה נכון לגביי JSON?

* **להגדרת אובייקט משתמשים ב-{}.**
* **ביטוי בצורת JSON יכול להיות בתוך קוד של JS.**
* בביטוי אחד של JSON אי אפשר להגדיר מערך אובייקטים.
* שם הנתון חייב להיות ללא גרשיים.
* JSON הוא סוג של שפת תכנות.

# שאלות ותשובות ממבחנים מרוכז:

## **Presentation 1 - HTML fundamentals CSS:**

### מה נכון בנוגע ל HTML?

נכון:

* **מסמך HTML הוא מסמך שמכיל טקסט בלבד.**
* **תווית (tag) <br> בונה שורה חדשה בתצוגה, אפילו אם בקוד של המסמך יש שורת טקסט אחת בלבד.**
* **ע"י תוויות (tags) נקבע תצוגה של מסמך HTML בדפדפן (browser).**
* **HTML היא שפת תגיות.**
* **ע"י תווית <a> ניתן ליצור לינק לקובץ HTML אחר.**
* **ע"י תווית <a> ניתן ליצור לינק לחלק אחר במסמך.**
* **לסמלים רבים ריקים יש את אותה המשמעות כמו לסמל אחד יחיד (למשל לעשות הרבה רווחים ייתן את אותה התוצאה כמו רווח אחד).**
* **מנגנון טבלאות (tables) של HTML נהוג להשתמש בקביעת מיקום יחסי של אלמנטים גרפיים על המסך.**
* **במקרה של דפדפן שאינו מכיר תווית (tag) שיש במסמך HTML הוא מתעלם מהתווית.**
* **שפה סטנדרטית ובסיסית של דפי web.**
* **סט של תגיות (tags) שמגדירים איך יציג הדפדפן את הטקסט והאלמנטים של דף ה-web.**
* **לא case sensitive.**
* **HTML לא מבוסס על XML.**
* **לכל תווית (tag) פתיחה או סגירה לא יכולים להיות מאפיינים (attributes).**
* **לתוויות (tags) באותיות גדולות וקטנות אין משמעות שונה.**
* **בפיתוח מסמך HTML אין להתייחס ליכולת הצגת המסמך במסמכים עם רזולוציה שונה.**
* **\*\*תוויות (tags) לא מיועדות לסמן טקסטים בהתאם לסוג ומשמעות הנתונים של המסמך\*\***
* **לא סוג שפת תכנות (זו שפת תגיות).**
* **\*\*לא חייבים לסגור כל תגית\*\***

לא נכון:

* טקסט של שורה אחת בקוד מקור של מסמך html אינה יכול להיות מוצגת במספר שורות.
* HTML מבוסס על XML.
* לכל תווית (tag) פתיחה או סגירה יכולים להיות מאפיינים (attributes).
* HTML היא סוג של שפת תכנות.
* יש לסגור כל תגית.

### מה נכון לגבי אלמנט form במסמך HTML?

נכון:

* **טופס מיועד במיוחד על מנת לשלוח מידע מ- html client ל –server.**
* **יכול להיות שיהיה רק אלמנט אחד של FORM במסמך HTML (יכולים להיות גם יותר).**
* **ניתן לשים ב-FORM יחיד יותר מכפתור אחד של submit.**
* **בתוך FORM שמים אלמנטים של קלט.**
* **מיועד לשליחת נתונים מלקוח html לשרת.**
* **לחיצה על כפתור submit שולחת נתונים לשרת ומעבירה לעמוד חדש.**
* **אלמנטים של input ממוקמים בתוך אלמנט form.**
* **יכול להיות רק אלמנט form אחד (יכול, לא חייב).**

### מה נכון בנוגע לאלמנט a במסמך HTML ולשורת הקוד הבאה?

### <a href=http://mis.haifa.ac.il> to IS site </a>

נכון:

* **אלמנט a משמש ליצירת קשר למסמכים אחרים.**
* **ערך המאפיין href של אלמנט a יכול להיות שם קובץ. מהמקרה הזה קשר יהיה לקובץ שחייב להיות באותה ספריה (directory) שהקובץ המסמך הנוכחי.**
* **אלמנט a יאפשר להשתמש לקביעת קשרים בתוך אותו מסמך שהאלמנט כתוב בו.**
* **ערך המאפיין href של אלמנט a לא נהוג להגדיר כמסלול שלם בתוך הכונן, למשל: c:\root\myDoc.html**

לא נכון:

* בשורת קוד יש שגיאה מפני שב-URL אין שם קובץ.
* ערך המאפיין href של אלמנט a נהוג להגדיר כמסלול שלם (qualified path fully) בתוך הכונן (כמו c:\root\myDoc.html).

### מה נכון בנוגע ל- CSS?

נכון:

* **קיימים סגנונות שיש להם מראה שונה בדפדפנים שונים.**
* **סגנון color מוגדר באלמנט style בתוך head וגם כן כתכונה (attribute) בשם style באלמנט. בתוך האלמנט הגדרה של תכונה יהיה בתוקף. מחוץ לאלמנט הגדרה של head יהיה בתוקף.**
* **סגנונות CSS אפשר להגדיר בקובץ נפרד ולקשר למסמך HTML.**
* **באמצעות סגנונות אפשר להגדיר מיקום של כל אלמנט HTML בכל מקום בחלון של דפדפן (browser).**
* **שולט בסגנון הטקסט**
* **מגדיר איך להציג אלמנטים של HTML.**
* **מאפשר למפתחים לשלוט בעיצוב וב-layout של כמה דפי web בו-זמנית.**
* **לא יכול להחליף שפת HTML.**
* **ישנה דרך להגדיר סגנון לאות בודדת שבתוך מילה אפילו אם להוסיף אלמנטים HTML נוספים.**

לא נכון:

* אין דרך להגדיר סגנון לאות בודדת שבתוך מילה, אפילו אם להוסיף אלמנטים HTML נוספים.
* לא שולט בתכונות ה-frame.

## **Presentation 2 - JavaScript:**

### מה נכון לגבי Javascript?

נכון:

* **זוהי שפת תכנות.**
* **קוד Javascript יכול להיות מוכל בכל חלק של מסמך HTML.**
* **משתנה של Javascript יכול לשנות את סוגו בזמן ריצת הקוד.**
* **לא נכון שקוד JavaScript מקומפל פעם אחת ורץ כמה פעמים (מקומפל מחדש עם כל ריצה)**
* **פונקציית JavaScript יכולה לרוץ ע"י אירוע**

לא נכון:

* Javascript מקומפל פעם אחת ורץ מספר פעמים. (תשובה נכונה: מקומפל מחדש עם כל ריצה).
* \*\*פונקציית javascript לא יכולה לרוץ ע"י one page event (תשובה נכונה: רצה בזמן הרצת הדף).\*\*

### מה נכון במוגע למשתנים ב- Javascript?

נכון:

* **\*\*למשתנה שמוגדר בתוך הפונקציה ללא שימוש בהגדרת var יש גישה גם מחוץ לפונקציה.\*\***
* **אם המשתנה מוגדר מחוץ לפונקציה תחום הכרתו כל המסמך כולל חלק שבתוך אלמנט body.**
* **למשתנה שקיבל ערך מספרי אפשר בהמשך לתת ערך של מחרוזת.**
* **סוג נתונים של משתנה מוגדר ע"י ערך של המשתנה.**
* **משתנה שמוגדר מחוץ לפונקציה מוכר בכל הדף.**
* **מערך הוא אובייקט.**
* **משתנים שמקבלים ערך מספרי יכולים להיות אובייקטים בהמשך.**
* **משתנים שמוגדרים בקובץ JS המקושר למסמך HTML כן קשורים למשתנים שמוגדרים בתוך המסמך.**
* **לא נכון שמשתנים אפשר להגדיר רק בתוך חלק של הקוד שנמצא באלמנט <head> של מסמך ה-HTML.**
* **למשתנים יש סוג נתונים.**

לא נכון:

* למשתנים אין סוג הנתונים.
* משתנים אפשר להגדיר רק בתוך חלק של הקוד שנמצא באלמנט head של מסמך HTML.

### מה נכון בנוגע לתכנות מונחה עצמים (OOP) ב- Javascript?

נכון:

* **לבניית אובייקט משתמשים באופרטור new.**
* **שימוש ב- JSON מאפשר דרך להגדרת קבוצת נתונים בצורת אובייקט.**
* **גישה למשתנים/נתונים חברי מחלקה אפשר לקבל גם באמצעות אופרטור [ ].**
* **אפשר להגדיר בנאי (constructor).**
* **להגדרת מחלקה לא משתמשים במילה שמורה class.**
* **אובייקט מכיל גם נתונים וגם פונקציות.**
* **מערך נתונים מספריים הוא אכן אובייקט.**
* **\*\*אובייקטים לא יכולים להוריש לאובייקטים אחרים\*\***
* **תמיד קיימים אובייקטים מרמת על.**

לא נכון:

* קיימים רק משתנים חברי מחלקה ולא קיימים פונקציות חברי מחלקה.
* אי אפשר להגדיר בנאי (constructor).

### מה נכון בנוגע לפונקציות ב-JS?

נכון:

* **אפשר להגדיר פונקציות בקובץ נפרד עם סיומת JS.**
* **אפשר להגדיר משתנים מקומיים שמוכרים בתוך הפונקציה בלבד.**
* **פונקציות יכולות להחזיר ערך באמצעות המילה השמורה return.**
* **פונקציות יכולות לקבל פרמטרים.**
* **בתוך כל פונקציה לא חייב להיות return.**
* **לא תמיד מועבר by reference.**
* **לא מוגדר בחלק ה-body.**
* **לא ניתן להשתמש בלי סוגריים.**
* **Date.parse() היא מתודה סטטית.**

### למה יכול לשמש JS?

נכון:

* **שינוי דפים בצורה דינאמית.**
* **Forms.**
* **לא נוגע לצד לקוח בלבד.**
* **לא נוגע לעיצוב העמוד.**
* **מאפשר אינטראקטיביות בין דפי web ולא עבור משתמשי web.**

## **Presentation 4 - jQuery Essentials:**

### מה נכון בנוגע ל- JQuery?

נכון:

* **קובץ שמכיל קוד JQuery אפשר להעתיק לפרויקט מקומי או להשתמש online מאתרים התומכים ב- JQuery.**
* **JQuery בנויה כך שאין צורך בקוד שונה לדפדפנים (browsers) שונים.**
* **לשימוש ב- JQuery חייבים במסמך HTML להגדיר קישור js שמכיל את הקוד של JQuery.**
* **מטרת JQuery היא לאפשר לכתוב מינימום קוד לביצוע עיבודים טיפוסיים עבור קבוצות אלמנטים של מסמכי HTML.**
* **בשפת תכנות Javascript קיימת רק ספריית פונקציות אחת שקיבלה שם JQuery.**

לא נכון:

* פונקציה $() מחזירה מערך אובייקטים של מסמך HTML לצורך עיבודם.
* ספריית פונקציות JQuery נוצרה לתמוך בקשר בין Javascript לבין Java.
* בשפת תכנות JS לא קיימת רק ספריית פונקציות אחת שקיבלה שם JQuery

### מה נכון בנוגע לפונקצית $() של JQuery?

נכון:

* **$(“tr:first”) מתייחס לשורה הראשונה בטבלה.**
* **$(“img”) מתייחס לכל אלמנטים תמונות.**
* **$(“tr”).css(“color:red”).css(“background-color:green”) זו לא שגיאה – אפשר להשתמש במתודה זהה פעמיים.**
* **$(“#div”) לא מתייחס לכל האלמנטים של div.**
* **$(“.div”) זו לא שגיאה כי אפשר להשתמש בנקודה במקום זה.**
* **פונקציית $() לא מחזירה מערך אובייקטים של מסמך HTML לצורך עיבודם.**

## **Presentation 5 - HTML5:**

### מה נכון בנוגע ל – HTML5?

נכון:

* **ב-HTML5 אפשר להשתמש בקבצי וידאו שונים למחשבים שיש להם תמיכת מולטימדיה שונה.**
* **ב-HTML5 לאלמנטים יש ערך סמנטי.**
* **ב-HTML5 יש מנגנון להגדיר גרפיקה באמצעות אלמנטים גרפיים ללא שימוש בקובץ נוסף.**

לא נכון:

* ב-HTML5 אין התייחסות ליכולות java script.
* HTML5 שמות האלמנטים באותיות גדולות קטנות קבלו משמעות שונה.

### מה נכון בנוגע לאלמנט DIV?

נכון:

* **אלמנט DIV אינו כולל עיצוב מיוחד כברירת מחדל.**
* **תוכן של אלמנט DIV אפשר לעשות בצורה דינאמית ע"י שינוי מאפיין inner HTML.**
* **אלמנט div בונה פיסקה (כמו שאלמנט P עושה).**
* **לאלמנט DIV אין תפקיד שונה מאלמנט div .**
* **במסמך HTML יכול להיות יותר מ-DIV אחד.**

## **Presentation 6 - Ajax-JSON:**

### מה נכון בנוגע ל- AJAX?

נכון:

* **AJAX מאפשר לקבל נתונים משרת ללא קבלת מסמך HTML כולו.**
* **מנגנון AJAX מאפשר תצוגה בדפדפן (browser) נתונים נוספים משרת תוך המשך עבודה במסמך המוצג.**

לא נכון:

* נתונים שמקבלים דרך AJAX חייבים להיות בצורת מסמך XML.
* מגדירים פונקציות תגובה שמנגנון AJAX הפעיל אותה רק פעם אחת בעת קבלת נתונים מהשרת.
* כפתור back של דפדפן (browser) מאפשר לחזור לנתונים שהיו מוצגים לפני הפעלת מנגנון AJAX.

### מה נכון בנוגע לפונקציית תגובה (callback) של AJAX?

נכון:

* **נתונים שהתקבלו מהשרת אפשר להציג למשתמש הקצה בתוך פונקציית התגובה.**
* **Http Status משתמשים בתוך פונקציית תגובה לצורך לבדוק האם נתונים מהשרת התקבלו בהצלחה.**
* **כל שינוי במצב שליחת/קבלת נתונים מביא להרצת פונקציית תגובה.**
* **נתונים מהשרת מתקבלים לא רק בצורת מסמך XML.**
* **פונקציית תגובה אי אפשר לכתוב בשפת תכנות JAVA.**

### מה נכון בנוגע ל-JSON?

נכון:

* **ביטויים בצורת JSON אפשר להשתמש להעברת נתונים בין שרת ללקוח.**
* **סימן : (נקודתיים) מפריד בין שם הנתון לערכו.**
* **ביטויים בצורת JSON יכולים להיות בתוך קוד JS.**
* **להגדרת אובייקט משתמשים בסוגריים מסולסלים { }.**
* **להגדרת אובייקט לא משתמשים בסוגריים מרובעים [ ].**
* **ערך בביטוי/שם הנתון של JSON לא חייב להיות ללא מרכאות (יכול להיות עם מרכאות או לא- ערכים מספריים לא צריכים מרכאות).**
* **JSON לא משתמשים רק בקוד של JS.**
* **JSON היא לא סוג של שפת תכנות.**
* **בביטוי אחד של JSON אפשר להגדיר מערך אובייקטים.**

## **Presentation 7 - JSP Basics:**

### מה נכון בנוגע למשפט הבא בקובץ JSP?

### <%! Public static final int num=100; %>

נכון:

* **הקבוע num נגיש לשימוש בכל מקום בקובץ.**
* **Num זה קבוע חבר מחלקה.**
* **ההצהרה הזו/המשפט הזה יכול להיות ממוקם בכל מקום בקובץ JSP.**
* **משתנה סטטי מסוג int יכול להיות בשימוש בכל שורה של קובץ JSP.**
* **אין שגיאה – מותר להשתמש בהגדרות final, public, static בקבצי JSP.**
* **לא נכון שניתן להשתמש ב-num רק בשורות שאחרי שורת קוד זו ב-JSP.**

לא נכון:

* המשפט יכול להיות ממוקם בקובץ רק לפני תווית <body>.
* ההצהרה הזו יכולה להיות ממוקמת רק לפני תגית <html>.
* שגיאה מפני שהגדרות כמו public, static, final אסור להשתמש בקבצי JSP.
* הקבוע num נגיש לשימוש רק אחרי השורה הזו של הגדרתו.
* משתנה סטטי מסוג int יכול להיות בשימוש רק בשורות אשר נמצאות אחרי הצהרה זו ב-JSP.

### מה נכון בנוגע לשימוש של JSP ו- servlet?

נכון:

* **אפשר לפתח יישום web שמשתמש ב- servlets ללא שימוש ב-JSP.**
* **שרת שתומך ב-JSP חייב לתמוך ב- servlets.**
* **שרת שתומך ב-servlet לא חייב לתמוך ב-JSP.**
* **אי אפשר לפתח יישום web שמשתמש ב-JSP ללא תרגום קבצי JSP ל-servlet ע"י המערכת.**

לא נכון:

* אפשר לפתח יישום web שמשתמש ב-JSP ללא תרגום קבצי JSP ל-servlets ע"י המערכת.
* שרת שתומך ב- servlets חייב לתמוך ב-JSP.

## **Presentation 9 - Web Applications:**

### מה נכון בנוגע ליישום web?

נכון:

* **ליישום web חייב להיות קובץ של הגדרת פריסה (descriptor deployment) ששמו web.xml.**
* **קובץ web.xml יכול להיות ריק שמכיל אלמנט ריק שורש של xml.**
* **שרת משתמש בהגדרות של קובץ web.xml רק פעם אחת כאשר יש פניה ראשונה של לקוח לקובץ כלשהו של יישום web.**
* **ליישום web חייב להיות ספריית שורש (root directory) שמכילה ספרייה בשם WEB-INF.**

לא נכון:

* שרת משתמש בהגדרות של קובץ web.xml כל פעם כאשר יש פנייה של לקוח לקובץ כלשהו של היישום web.
* שרת יכול להריץ יישום web אחד בלבד.

### מה נכון לגבי יישום web ב-JAVA?

נכון:

* **כל קבצי יישומי web (html, jsp, images, java classes , etc.) חייבים להיות תחת אותה תיקיית root.**
* **קיימים יישומי web אשר אין להם קבצי html.**
* **בתוך תיקיית יישומי web חייבת להיות תת תיקייה של web-inf.**

לא נכון:

* הימצאות קובץ web.xml בתוך תיקיית web-inf היא אופציונלית.

### נתון קוד ה- JSP הבא (מתקמפל):

**1. <HTML>  
2. <body>  
3. <% String gr=”Hello”; %>  
4. <h1><%= gr %></h1>  
5. <%=gr+” “+request.getParameter(“name”)%>  
6. </body>  
7. </html>**

**מה נכון לגבי קוד זה מההצהרות הבאות :**

נכון:

* **בתוך <% %> כתוב קוד JAVA כך ש gr משתנה לוקאלי.**
* **קוד בשורה 4 יוצר בדפדפן פלט "Hello".**
* **קוד בשורה 3 אי אפשר לראות בצד הלקוח בדפדפן.**

**לא נכון:**

* שגיאת קומפילציה בשורה 4 מפני שחסר סימן ; בסוף – לא נכון, מתקמפל.
* שגיאת קומפילציה בשורה 5 מפני שחסר סימן ; בסוף.
* שגיאת קומפילציה בשורה 5 מפני שאין הגדרה של משתנה request.
* קוד בשורה 3 יוצר בדפדפן פלט "Hello".

### מה נכון בנוגע ל resultset של jdbc ?

נכון:

* **מתודת getString() משתמשים לקבלת נתונים.**
* **אובייקט של resultset יכול להיות ריק.**
* **אובייקט שלresultset מחזיק טבלת נתונים שהוגדרו בdb ע"י פקודת select של sql.**
* **לא מיועד לשימוש בישומי web בלבד.**
* **במתודת next לא משתמשים לבדיקת השורה האחרונה של הנתונים המוחזרים**

לא נכון:

* מיועד לשימוש ביישומי web בלבד.
* מתודת next() משתמשים לבדיקת השורה האחרונה ... של הנתונים המוחזרים.

### מה נכון לגבי java web application?

נכון:

* **לכל קבצי application web (html, JSP, Picture, Java class וכו') חייב להיות common root application directory.**
* **קיימים web application שאין להם קבצי html.**
* **חייבת להיות תת תיקיה WEB-INF.**
* **קיום קובץ web.xml בתיקיית WEB-INF לא אופציונאלי!**

## **Presentation 11 - JSP and JavaBeans:**

### מה נכון בנוגע ל- session?

נכון:

* **Session משתמשים לשמירת נתונים של משתמש הקצה בנפרד לנתונים של משתמשים אחרים.**
* **יש שימוש רחב ב- sessions עבור שמירת מידע על פעולות המשתמש**
* **Session חדש נוצר כל פעם שלקוח שאין לו session שולח שאילתה (request) לשרת.**
* **אפשר לסיים session בצורה תכנותית ללא קשר ל- timeout.**
* **זה אפשרי לסיים Session באמצעות תכנות לפני שה- timeoutמסתיים.**
* **Session של משתמש הסתיים כאשר המשתמש לא שולח שאילתה (request) לפרק זמן ארוך מזמן מסוים שנקרא timeout.**
* **Session נפתח לכל user.**
* **אובייקט ה-session נגיש מכל דף JSP.**
* **בכל פעם שמשתמש שולח בקשה ליישום ווב ולא משויך לו Session, Session חדש יתחיל.**
* **אם משתמש לא שולח בקשה לאינטרוול גדול יותר, אז יש timeout וה- Session של המשתמש ייסגר באופן אוטומטי.**
* **Session משמש לאחסון מידע על פעולות משתמש אישיות.**
* **לא נכון שבכל פעם שמשתמש שולח בקשה נפתח session חדש.**
* **אפשר לשנות timeout דרך הקוד (זה מאפיין של web server).**
* **Session לא משמש לאחסון מידע על המצב הנוכחי של האפליקציה.**

לא נכון:

* Timeout מוגדר עבור יישום ולא ניתן לשנות אותו בצורה תכנותית.
* Session חדש נוצר כל פעם שלקוח שולח שאילתה (request) לשרת (במילים אחרות: בכל פעם שמשתמש שולח בקשה ליישום Web (למשל אתר אינטרנט), session חדש מתחיל).
* יש שימוש נרחב של sessions עבור שמירת מידע של מצב היישום.
* אובייקט ה-session ניגש לכל javaBean class.
* Timeout זהו תכונה של שרת ווב ולא ניתן לשנות אותה באמצעות תכנות.

### מה נכון בנוגע לקוד הבא?

1. **<%@ page import = "java.sql.Connection" %>**
2. **<jsp:useBean id = "con"**
3. **Type = "java.sql.Connection" scope = "session"/>**

נכון:

* **אם אובייקט בשם con אינו קיים ב-session הנוכחי הוא יוצר.**
* **אובייקט מסוג java.sql.Connection יוצר לכל משתמש בנפרד.**
* **קיימת אפשרות להחליף <jsp:useBean> בשורות 2-3 לקוד השקול בתוך <% %>.**

לא נכון:

* סוג האובייקט בשורה 3 אפשר לקצר ל- type = "Connection" מפני שבשורה 1 יש import המתאים.

### מה נכון לגבי javaBeans?

נכון:

* **חייב להיות למחלקה זו default constructor.**
* **אם למחלקת javaBeans יש מתודה בשם getData(), אז ל- bean יש מאפיין בשם data.**
* **javaBean זה סוג מיוחד של מחלקה שמשתמשים בה לא רק ב-web application.**
* **יכול להיות למחלקת javaBean בנאי שונה מברירת המחדל.**
* **מאפייניjavaBean לא מוצגים ע"י מילת המפתח property.**

לא נכון:

* זהו סוג מיוחד של מחלקה אשר משתמשים בה רק עבור יישומי Web.
* לא יכול להיות למחלקה זו constructor שונה מברירת המחדל.
* מאפייני javaBeans מוצגים ע"י מילת המפתח property.

### מה נכון לגבי שימוש ב-javaBeans ב-JSP ?

נכון:

* **הכוונה בסקופ session: היא שנתוני bean נשמרים בנפרד עבור כל משתמש.**
* **זה כדאי לשמור פרמטרים של בקשות של משתמשי קצה באובייקטים של javaBeans.**
* **אפשר לגשת לאובייקט javaBean ב-JSP לא רק באמצעות תאג: <jsp: useBean>.**
* **ה-scope של request לא אומר שנתוני bean נשמרים עבור כל עמוד בנפרד.**
* **לא חייבת להיות תגית <jsp: setProperty> ל-javaBean.**

לא נכון:

* פנייה לאובייקט javaBean ע"י JSP נעשית רק ע"י תגית <jsp : useBeans>.
* הכוונה בסקופ request: היא שנתוני bean נשמרים בנפרד עבור כל דף.
* ל- javaBeans חייבת להיות תגית <jsp: setProperty>.

### מה נכון לגבי Statement Object?

נכון:

* **משתמשים בו לחיבור לשאילתת sql.**
* **ניתן לייצרו ע"י מתודה מיוחדת של קישוריות.**
* **להצהרה יש נתונים על מיקום וסכמת מסד הנתונים כי היא נוצרת ע"י החיבור למסד הנתונים.**
* **לא נוצר ע"י הבנאי שלו.**
* **אפשרי להריץ יותר משאילתה אחת באמצעות אובייקט statement אחד.**

לא נכון:

* ניתן לייצרו ע"י ה- constructor שלו.
* זה בלתי אפשרי להריץ יותר משאילתה אחת ע"י אובייקט הצהרה אחד.

### מה נכון לגבי שימוש JDBC ב-JSP?

נכון:

* **מומלץ לשים קוד SQL בקבצי javaBeans.**
* **מומלץ לעשות שימוש חוזר עבור משתמשים שונים ע"י מנגנון התחברות.**
* **לא מומלץ ליצור connection חדש עבור כל דף JSP**
* **לא עדיף לייבא את כל המידע מה-DB לאובייקטים של JAVA מאשר לנתח את המידע בשרת ה-SQL.**

לא נכון:

* מומלץ ליצור חיבור חדש עבור כל דף JSP.
* זה עדיף להביא את כל הנתונים ממסד הנתונים על אובייקטים של java , מאשר לנתחם בתוך שרת ה- SQL.

### מה נכון בנוגע ל-JSP?

נכון:

* **קוד JAVA יכול להשתמש במחלקות אחרות כמו כל פיתוח תוכנה.**
* **במסמך JSP יש ערבוב של JS, HTML, JAVA.**
* **קוד של JS אפשר ליצור ב-JAVA.**
* **מסמך JSP אי אפשר לקמפל בפרויקט רגיל של JAVA.**
* **נתונים של JS אי אפשר להשתמש גם ב-JAVA.**

### מה נכון בנוגע למשפט הבא בקובץ JSP?

### <%=a.toString()%>

נכון:

* **המשפט יוצר מחרוזת במסמך שנשלח למשתמש הקצה.**
* **משתנה a חייב להיות מוגדר היטב לפני המשפט.**
* **אין שגיאה - זה בסדר שאין ; (נקודה פסיק) בסוף.**
* **אי אפשר להחליף את המשפט בביטוי זהה בJS.**
* **המשפט לא חייב להיות ממוקם בקובץ רק אחרי תווית <body>.**

### מה נכון בנוגע לקוד הבא?

1. **<%@ page import = “ik.myApp.user”%>**
2. **<jsp:UseBean id=”user”**
3. **Type=”ik.myApp.user” scope=”session”/>**

נכון:

* **אם אובייקט בשם user אינו קיים ב-session הנוכחי הוא יוצר.**
* **אותו אובייקט בשם user יהיה נגיש בכל קבצי JSP כל עוד ה-session אינו הסתיים.**
* **קיימת אפשרות להחליף <JSP:UseBean> בשורות 2 ו-3 לקוד השקול בתוך <% %>.**
* **אובייקט מסוג ik.myApp.user יוצר לכל משתמש בנפרד.**
* **סוג האובייקט בשורה 3 אי אפשר לקצר ל-type=”user” (גם לא בגלל שבשורה 1 יש import מתאים).**

### מה נכון בנוגע לקוד הבא?

1. **<%@ page import = “java.sql.connection”%>**
2. **<jsp:UseBean id=”con”**
3. **Type=” java.sql.connection” scope=”session”/>**

נכון:

* **אותו דבר כמו הקודם**

### מה נכון בנוגע לשימוש JDBC?

נכון:

* **JDBC יכול לעבוד עם DBMS שונים (SQL, MYSQL, ACCESS SERVER, ORACLE).**
* **לקבלת גישה לבסיס הנתונים חייבים להשתמש בשם משתמש וסיסמא נכונים של בסיס הנתונים.**
* **JDBC דרייבר חייב להיות נטען פעם אחת לפני יצירת קשר לבסיס הנתונים.**
* **DBMS יכול לעבוד עם יותר מטבלת נתונים אחת מכל מיני סוגי DBMS.**
* **במקרה של כמות נתונים גדולה לא מומלץ למנוע שימוש ב-DBMS.**

תוכן עניינים

[Presentation 1 – HTML – Hyper Text Markup Language: 1](#_Toc456447807)

[**נקודות חשובות:** 1](#_Toc456447808)

[HTML: 1](#_Toc456447809)

[Forms: 1](#_Toc456447810)

[עיצובים ו-CSS: 1](#_Toc456447811)

[תמונה: 1](#_Toc456447812)

[**סיכום מצגת:** 2](#_Toc456447813)

[העברת מידע: 2](#_Toc456447814)

[HTML – הגדרה: 2](#_Toc456447815)

[כללים בסיסיים: 2](#_Toc456447816)

[Presentation 2 – Java Script Basics: 7](#_Toc456447817)

[**נקודות חשובות:** 7](#_Toc456447818)

[כללי: 7](#_Toc456447819)

[פונקציות: 7](#_Toc456447820)

[מערכים: 7](#_Toc456447821)

[DOM: 7](#_Toc456447822)

[**סיכום מצגת:** 8](#_Toc456447823)

[Java Script – הגדרה: 8](#_Toc456447824)

[חיבור בין HTML לבין JS: 8](#_Toc456447825)

[טיפוסי נתונים: 8](#_Toc456447826)

[פונקציות: 8](#_Toc456447827)

[מערכים: 9](#_Toc456447828)

[תמונות: 10](#_Toc456447829)

[אובייקטים: 10](#_Toc456447830)

[DOM (Domain Object Model) – הגדרה: 10](#_Toc456447831)

[Presentation 3 – DHTML Basics: 11](#_Toc456447832)

[**נקודות חשובות:** 11](#_Toc456447833)

[**סיכום מצגת:** 11](#_Toc456447834)

[DHTML (Dynamic HTML) – הגדרה: 11](#_Toc456447835)

[שיטות שימוש עיקריות: 11](#_Toc456447836)

[Presentation 4 – jQuery Essentials: 13](#_Toc456447837)

[jQuery – הגדרה: 13](#_Toc456447838)

[קישור לספריית jQuery: 13](#_Toc456447839)

[שימוש בסיסי ב-jQuery בתוך JS: 13](#_Toc456447840)

[שימוש קצת יותר מתקדם ב-jQuery בתוך JS: 13](#_Toc456447841)

[הבדלי Set ו-Get: 13](#_Toc456447842)

[Presentation 5 – HTML5: 14](#_Toc456447843)

[Presentation 6 – Ajax-JSON: 16](#_Toc456447844)

[**נקודות חשובות:** 16](#_Toc456447845)

[**סיכום מהמצגת:** 16](#_Toc456447846)

[Ajax (Asynchronous JavaScript and XML) – הגדרה: 16](#_Toc456447847)

[דוגמאות מחיי היומיום: 16](#_Toc456447848)

[יתרונות של Ajax: 16](#_Toc456447849)

[חסרונות של Ajax: 16](#_Toc456447850)

[איך זה עובד: 16](#_Toc456447851)

[JSON (JavaScript Object Notation) – הגדרה: 17](#_Toc456447852)

[Syntax of JSON: 18](#_Toc456447853)

[Presentation 7 – JSP Basics: 19](#_Toc456447854)

[JSP (Java Server Pages – הגדרה: 19](#_Toc456447855)

[Servlet – הגדרה: 19](#_Toc456447856)

[יתרונות ה-Servlet: 19](#_Toc456447857)

[Presentation 9 – Web Applications: 20](#_Toc456447858)

[J2EE (Java 2 Enterprise Edition) – הגדרה: 20](#_Toc456447859)

[Components, Containers and Deployment: 20](#_Toc456447860)

[Web Application – הגדרה: 21](#_Toc456447861)

[מבנה התיקיות: 21](#_Toc456447862)

[Deployment Web Application: 22](#_Toc456447863)

[Web Application Archive (WAR): 22](#_Toc456447864)

[The web.xml File: 22](#_Toc456447865)

[**סיכום:** 23](#_Toc456447866)

[Presentation 10 – Cookies and Session Management: 24](#_Toc456447867)

[**נקודות חשובות:** 24](#_Toc456447868)

[**סיכום מצגת:** 24](#_Toc456447869)

[Session – הגדרה: 24](#_Toc456447870)

[שימושים נפוצים: 24](#_Toc456447871)

[יצירת Session: 24](#_Toc456447872)

[Session Management: 24](#_Toc456447873)

[שיטות מעקב אחר Session: 25](#_Toc456447874)

[שיטות לסגירת Session: 26](#_Toc456447875)

[Presentation 11 – JSP and JavaBeans: 27](#_Toc456447876)

[**נקודות חשובות:** 27](#_Toc456447877)

[**סיכום מצגת:** 27](#_Toc456447878)

[JavaBeans – הגדרה: 27](#_Toc456447879)

[שימוש ב-JavaBeans: 27](#_Toc456447880)

[JavaBeans ו-JSP: 27](#_Toc456447881)

[מתי להשתמש ב-JavaBeans: 27](#_Toc456447882)

[שאלות לדוגמה מההרצאות שעלולות להיות במבחן: 28](#_Toc456447883)

[האם HTML היא שפת טקסט? 28](#_Toc456447884)

[האם HTML היא שפת תכנות? 28](#_Toc456447885)

[האם תצוגת מסמך HTML מכילה רק טקסט? 28](#_Toc456447886)

[מי המפרש (Compiler) של שפת HTML? 28](#_Toc456447887)

[לכל משתנה ב-JS תמיד קיים טיפוס נתונים? 28](#_Toc456447888)

[האם הפקודה $("div") מחזירה מערך של אובייקטים? 28](#_Toc456447889)

[האם ניתן לכתוב Ajax בלי jQuery? 28](#_Toc456447890)

[האם יכול להיות שרת שתומך ב-JSP ולא תומך ב-Servlet? 28](#_Toc456447891)

[האם ניתן להגדיר משתנה מקומי בתור private? 28](#_Toc456447892)

[האם ניתן להגדיר משתנה גלובלי בתור private? 28](#_Toc456447893)

[האם מי שלא מתחבר לשרת לא פותח Session? 28](#_Toc456447894)

[שאלות מתרגול חזרה למבחן: 29](#_Toc456447895)

[שאלות משחזורים: 33](#_Toc456447896)

[מה נכון לגביי HTML? 33](#_Toc456447897)

[מה נכון לגבי אלמנט div? 33](#_Toc456447898)

[מה נכון לגבי אלמנט form במסמך HTML? 33](#_Toc456447899)

[מה נכון לגביי משתנים ב-JS? 33](#_Toc456447900)

[מה נכון בנוגע לתכנות מונחה עצמים ב-JS? 34](#_Toc456447901)

[מה נכון לגביי JS? 34](#_Toc456447902)

[מה נכון לגביי מערכים ב-JS? 34](#_Toc456447903)

[מה נכון בנוגע ל-Ajax? 34](#_Toc456447904)

[מה נכון לגביי שורת הקוד הבאה: 34](#_Toc456447905)

[Document.getElementById(a).style.backgroundColor = "red" 34](#_Toc456447906)

[מה נכון לגבי DHTML ו-DOM? 35](#_Toc456447907)

[מה נכון לגבי המשפט הבא בקובץ JSP? 35](#_Toc456447908)

[<%=a.toString() %> 35](#_Toc456447909)

[מה נכון בנוגע לשימוש של JSP ו-Servlet? 35](#_Toc456447910)

[מה נכון לגביי שורת הקוד הבאה? 35](#_Toc456447911)

[<a href=http://....> link here</a> 35](#_Toc456447912)

[מה נכון לגביי CSS? 35](#_Toc456447913)

[מה נכון לגביי שורת הקוד הבאה? 36](#_Toc456447914)

[<IMG SRC...onclick="do.this;do.that" 36](#_Toc456447915)

[מה נכון לגביי jQuery? 36](#_Toc456447916)

[מה נכון לגביי JSON? 36](#_Toc456447917)

[שאלות ותשובות ממבחנים מרוכז: 37](#_Toc456447918)

[**Presentation 1 - HTML fundamentals CSS:** 37](#_Toc456447919)

[מה נכון בנוגע ל HTML? 37](#_Toc456447920)

[מה נכון לגבי אלמנט form במסמך HTML? 38](#_Toc456447921)

[מה נכון בנוגע לאלמנט a במסמך HTML ולשורת הקוד הבאה? 38](#_Toc456447922)

[<a href=http://mis.haifa.ac.il> to IS site </a> 38](#_Toc456447923)

[מה נכון בנוגע ל- CSS? 39](#_Toc456447924)

[**Presentation 2 - JavaScript:** 40](#_Toc456447925)

[מה נכון לגבי Javascript? 40](#_Toc456447926)

[מה נכון במוגע למשתנים ב- Javascript? 40](#_Toc456447927)

[מה נכון בנוגע לתכנות מונחה עצמים (OOP) ב- Javascript? 41](#_Toc456447928)

[מה נכון בנוגע לפונקציות ב-JS? 41](#_Toc456447929)

[למה יכול לשמש JS? 41](#_Toc456447930)

[**Presentation 4 - jQuery Essentials:** 42](#_Toc456447931)

[מה נכון בנוגע ל- JQuery? 42](#_Toc456447932)

[מה נכון בנוגע לפונקצית $() של JQuery? 42](#_Toc456447933)

[**Presentation 5 - HTML5:** 43](#_Toc456447934)

[מה נכון בנוגע ל – HTML5? 43](#_Toc456447935)

[מה נכון בנוגע לאלמנט DIV? 43](#_Toc456447936)

[**Presentation 6 - Ajax-JSON:** 44](#_Toc456447937)

[מה נכון בנוגע ל- AJAX? 44](#_Toc456447938)

[מה נכון בנוגע לפונקציית תגובה (callback) של AJAX? 44](#_Toc456447939)

[מה נכון בנוגע ל-JSON? 44](#_Toc456447940)

[**Presentation 7 - JSP Basics:** 45](#_Toc456447941)

[מה נכון בנוגע למשפט הבא בקובץ JSP? 45](#_Toc456447942)

[<%! Public static final int num=100; %> 45](#_Toc456447943)

[מה נכון בנוגע לשימוש של JSP ו- servlet? 45](#_Toc456447944)

[**Presentation 9 - Web Applications:** 46](#_Toc456447945)

[מה נכון בנוגע ליישום web? 46](#_Toc456447946)

[מה נכון לגבי יישום web ב-JAVA? 46](#_Toc456447947)

[נתון קוד ה- JSP הבא (מתקמפל): 47](#_Toc456447948)

[מה נכון בנוגע ל resultset של jdbc ? 47](#_Toc456447949)

[מה נכון לגבי java web application? 48](#_Toc456447950)

[**Presentation 11 - JSP and JavaBeans:** 49](#_Toc456447951)

[מה נכון בנוגע ל- session? 49](#_Toc456447952)

[מה נכון בנוגע לקוד הבא? 50](#_Toc456447953)

[מה נכון לגבי javaBeans? 50](#_Toc456447954)

[מה נכון לגבי שימוש ב-javaBeans ב-JSP ? 50](#_Toc456447955)

[מה נכון לגבי Statement Object? 51](#_Toc456447956)

[מה נכון לגבי שימוש JDBC ב-JSP? 51](#_Toc456447957)

[מה נכון בנוגע ל-JSP? 51](#_Toc456447958)

[מה נכון בנוגע למשפט הבא בקובץ JSP? 52](#_Toc456447959)

[<%=a.toString()%> 52](#_Toc456447960)

[מה נכון בנוגע לקוד הבא? 52](#_Toc456447961)

[מה נכון בנוגע לקוד הבא? 52](#_Toc456447962)

[מה נכון בנוגע לשימוש JDBC? 52](#_Toc456447963)