**DOC­S – Advanced JavaScript 2023-02-02**

TypeScript

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

* הרצת קומפיילר של TypeScript בטרמינל – tsc
* הרצת קובץ ה compile ב hot reloading : tsc – w

**Literal Object**

זה הוא אובייקט שניתן לבנות – { } , המכיל מאפיינים וערכים עבור המאפיינים.

const kitten1 = { name: 'Mitsi', age: 4, color: 'black' };

אם אנו רוצים מספר רב של אובייקטים, לדוגמה : מספר רב של חתולים, עלינו לשכפל את הקוד על ידי יצירת אובייקט חדש, המכיל מאפיינים חדשים וערכים חדשים.

**OOP – object oriented programming**

שיטה תכנותית – המאפשרת לנו לבנות תבנית ליצירת אובייקטים (ישויות).

התבנית מגדירה מה יש באובייקט שיווצר ממנה.

היא מכילה מאפיינים ופונקציות (מתודות).

המילה השמורה שממנה יוצרים תבנית, נקראת –

Class

בעברית : מחלקה.

את ה Class יוצרים באופן נפרד, כאשר הקונבנציה היא – ששם הקובץ הוא גם שם התבנית.

קונבנציה נוספת : שם ה Class, יתחיל באות גדולה.

**Properties/Fields/Data members**

אלו משתנים שמוגדרים בתוך ה Class

כל משתנה כזה יוצר בתוך האובייקט שמשויך ל Class

**מתודות - Methods**

אלו הן פונקציות שמוגדרות בתוך ה class

אובייקט שנוצר מה class, יוכל להפעיל אותן. הן פועלות על אותו אוביקט שהפעיל אותן.

הרשאות גישה - Access Modifiers

אלו הן מילים שמורות, המציינות את ה scope (המרחב – מהיכן ניתן לגשת) אל משתנה או אל פונקציה (מתודה) בתוך ה Class

הרשאות הגישה הקיימות ב TypeScript – הן:

1. Public – נגיש גם מתוך ה class וגם מחוץ ל class, זו היא גם ברירת המחדל.
2. Private – נגיש אך ורק מתוך ה class, קוד שנמצא מחוץ ל class, לא יכול להשתמש או לגשת למשתנים או לפונקציות המוגדרות כ private
3. Protected – נגיש מתוך ה class או מ classים (מחלקות) אחרים שיורשים מאותו ה class

**ה constructor**

זו היא פונקציה מיוחדת שהמחלקה (ה class) קורא לה אוטומטית עבורנו, ברגע שנוצר אובייקט (ברגע שנוצר Instance חדש מתוך המחלקה).

אי אפשר לקרוא לה, היא מופעלת פעם אחת בלבד ברמת אובייקט (כל אובייקט) שנוצר מתוך המחלקה.

תפקידה הוא לאתחל את האובייקט ברגע שהוא נוצר (כמו כל פונקציה, היא יכולה לקבל ארגיומנטים , אבל היא לא מחזירה כלום).

**Inheritance - הורשה**

זו פעולה של העברת תוכן מ class א ל class ב, ללא שכפול הקוד.

ה מחלקה הראשונה נקראת מחלקת בסיס - Super Class or Base Class

ה מחלקה השנייה נקראת מחלקה נגזרת - Sub Class or Derived Class

זה מאפשר ליצור בקלות מחלקה נוספת שמכילה את כל מה שהיה במחלקה הקודמת (וגם מאפשרת לנו להוסיף עוד דברים במידת הצורך)

אם ברצוננו להוסיף מאפיינים ל Class היורש, אנו יכולים להוסיף עוד מאפיינים.

אם ברצוננו להוסיף פונקציות לClass היורש, אנו יכולים להוסיף עוד פונקציות.

אם ה constructor לא מתאים לנו, אפשר לבנות אחד אחר שדורס את המקורי

במצב של דריסת constructor , חובה עלינו לקרוא ל constructor של המחלקה שממנה ירשנו אותו (מ SUPER).

אם פונקציה מסוימת שקיבלנו ממחלקת בסיס, לא מתאימה לנו, אפשר לדרוס אותה גם על ידי מתן שם זהה לפונקציה החדשה.