**Debug**

לעתים, התוכנית לא תבצע במדויק את מה שתכננו שהיא תבצע. כמעט תמיד בהרצה ובדיקה ראשונית של התוכנית שנכתוב יהיו תקלות.  
התקלות האלו, נגרמות בגלל טעויות של המתכנת, שכתב את הקוד של התוכנית בצורה לא נכונה. בעיות אלו נקראות **באגים.**

**Debug** מסייע לנו למצוא באגים בתוכנית שלנו.

ממש כמו "ריצה יבשה" (טבלת מעקב) שבה אנו עוקבים אחרי הריצה של הקוד שלנו ממש צעד אחר צעד, Debug מאפשר לנו לעשות זאת בצורה יעילה יותר דרך סביבת העבודה.

יש הרבה שגיאות שניתן לאתר בצורה מהירה מאוד ע"י הקומפיילר (שגיאות syntax למשל(.

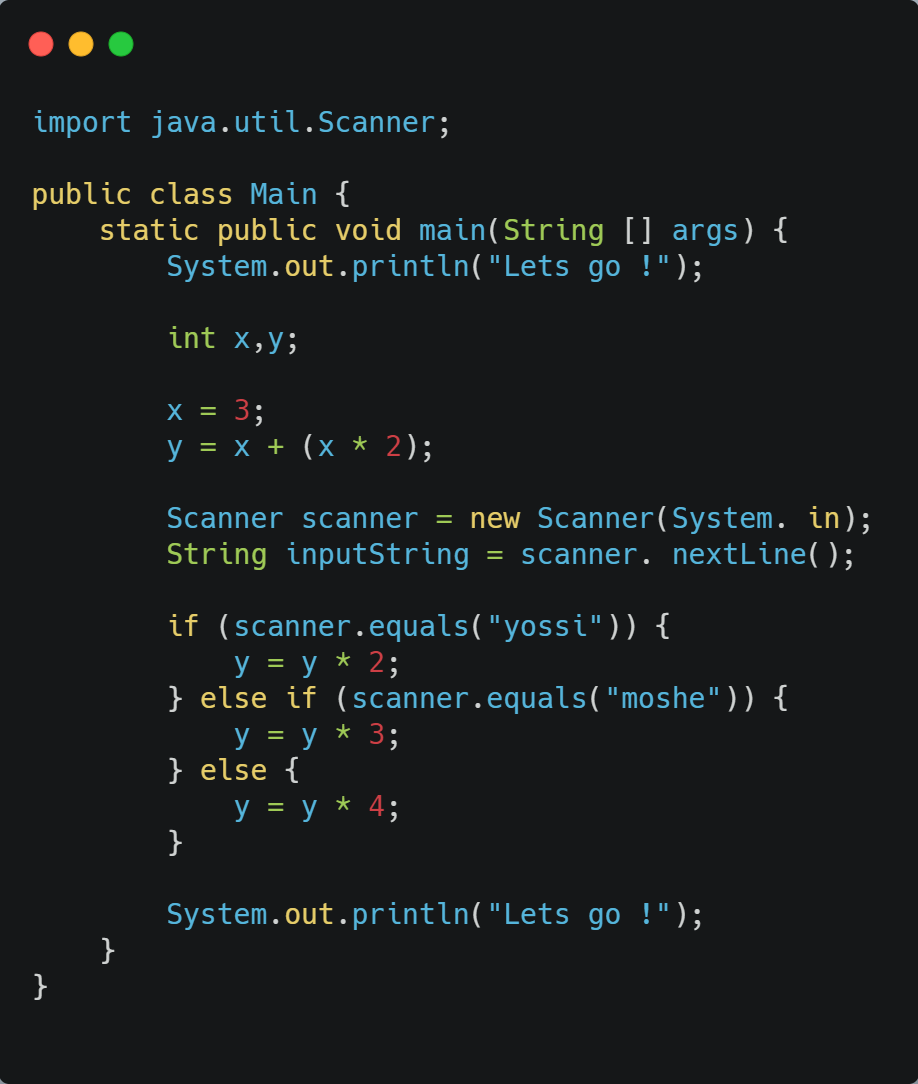
לעתים, גם עבור שגיאות שמתרחשות **בזמן הריצה** קיים תיעוד המסביר על מיקום הבאג וכיצד ניתן לפתור אותו.

לעומת זאת, ישנן שגיאות מסובכות שכמעט בלתי אפשרי לגלות אותן באמצעות הקומפיילר.  
לכן, בעבודה בפרויקטים גדולים המכילים אלגוריתמיקה מורכבת, כדי "לעקוב" אחרי הקוד ולמצוא שגיאות/טעויות או כדי להבין איך הקוד עובד – נשתמש בכלי ה-Debug.

לשמחתנו, אנחנו עובדים בסביבת העבודה intellij אשר מאפשרת יכולות Debug מגוונות עם ממשק משתמש נוח מאוד.

**Debug בסביבת intellij בJava-**

נסו לכתוב קטע קוד קצר, שעושה פעולה כלשהי (באורך של כ-7-10 שורות קוד)

למשל:

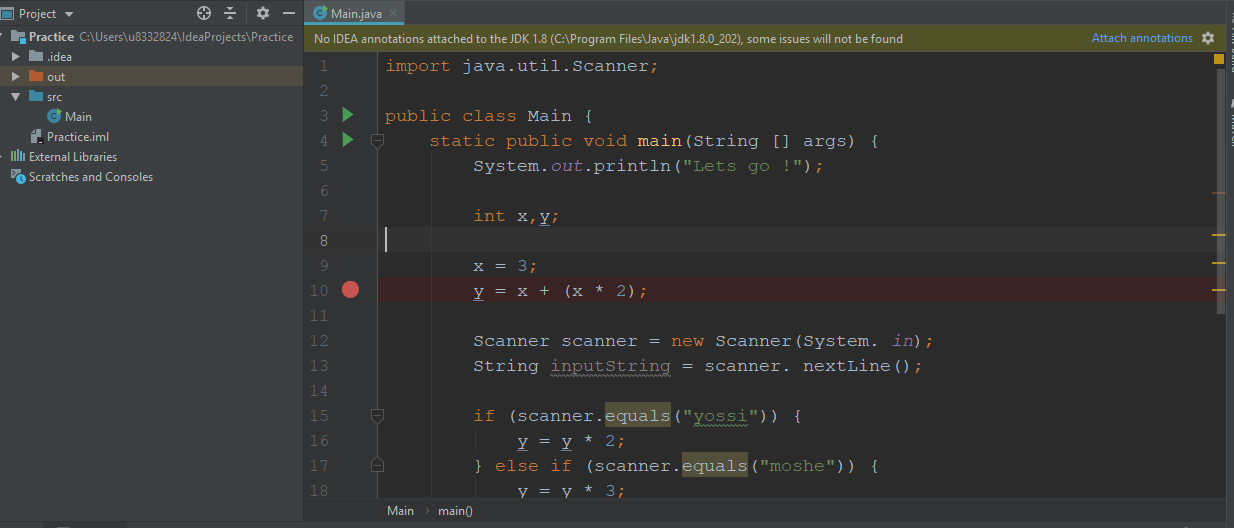
ניתן לערוך את הקוד [בלינק הבא](https://carbon.now.sh/?bg=rgba(171%2C%20184%2C%20195%2C%201)&t=seti&wt=none&l=auto&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Lets%2520go%2520!%2522)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520x%252Cy%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520x%2520%253D%25203%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520y%2520%253D%2520x%2520%252B%2520(x%2520*%25202)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520scanner%2520%253D%2520new%2520Scanner(System.%2520in)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520String%2520inputString%2520%253D%2520scanner.%2520nextLine()%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520if%2520(scanner.equals(%2522yossi%2522))%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520y%2520%253D%2520y%2520*%25202%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%2520else%2520if%2520(scanner.equals(%2522moshe%2522))%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520y%2520%253D%2520y%2520*%25203%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%2520else%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520%2520y%2520%253D%2520y%2520*%25204%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520%257D%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Lets%2520go%2520!%2522)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A).

כעת נרצה לדבג את הקוד שלנו.  
כלומר – לעבור אחרי הריצה שלו שורה אחר שורה ולראות את ערכי המשתנים.

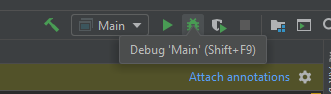
ראשית עלינו להוסיף breakpoint.

**Breakpoint** – נקודה שבה ריצת הקוד תיעצר ונוכל לעקוב אחרי הקוד **בצורה ידנית** מנקודת זו באמצעות debug.

בכדי להוסיף breakpoint , נבחר שורה בקוד שממנה אנחנו רוצים להתחיל לעקוב אחרי ריצת התוכנית, ונלחץ מצד שמאל לשורה – ככה תתווסף נקודה אדומה.  
(הנקודה האדומה היא הbreakpoint)

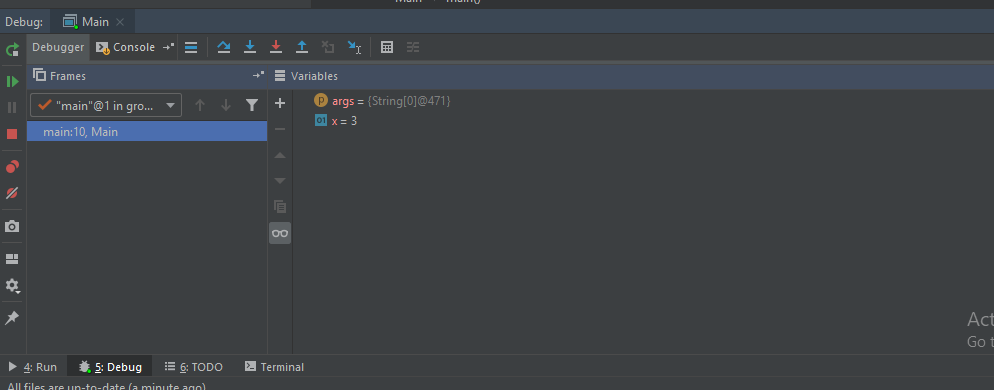


כעת עלינו להריץ את הפרויקט **במצב דיבאג.**  
ע"י לחיצה על הכפתור הבא בסרגל הכלים העליון:



(או ע"י קיצורי המקשים shift + f9)

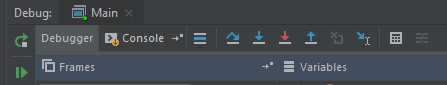
אחרי שהרצנו את הפרויקט במצב debug, נראה את החלון הבא:



זהו חלון הDEBUGGER בו נמצא כל המידע הרלוונטי לתהליך הדיבאג והמצב הנוכחי של התוכנית.

השורה הצבועה כרגע בצבע כחול בקוד שלנו, היא השורה הנוכחית בתוכנית בזמן הריצה הנתון כרגע.

בכדי לעבור לשורה הבאה בריצת התוכנית (Step Over) נשתמש בכפתורים הבאים בחלון הDEBUGGER :

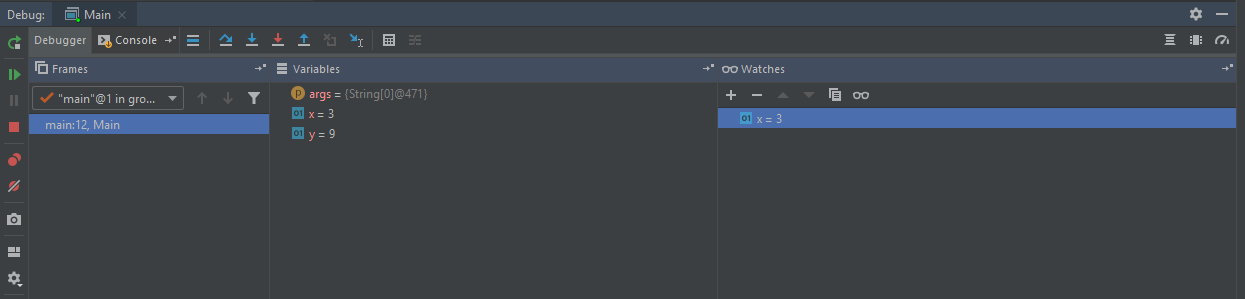


לאחר שעברנו לשורה הבאה, נראה שהפס הכחול מתקדם לשורה הבאה.

* יש מספר רב של דרכים כיצד להתקדם בדיבאג, בחנו את הכפתורים האחרים בסרגל הכלים של חלון הדיבאג.

נכיר עכשיו כלים חשובים שיעזרו לנו לראות את הערך הנוכחי של המשתנים כרגע ולדאבג:

* ניתן לראות שערכי המשתנים המוצגים בחלון זה משתנים בהתאם לריצת התוכנית.
* **חלון הVariables** – מציג את ערך המשתנים שאותחלו ועדיין יש בהם שימוש בתוכנית.
* **חלון הWatchers** – מאפשר לצפות במשתנים שנבחר ע"י לחיצה על ה + וכתיבת שם המשתנה בו נרצה לצפות.



יכולת הדיבאג היא מיומנות שדורשת ניסיון והיא חלק אינטגרלי מהעבודה היומיומית של תוכניתן, חשוב להשקיע בה זמן ולתרגל אותה.  
במיוחד בסביבת intellij, קיימים הרבה כלי יכולות debug מתקדמות נוספות וקיצורי מקלדת יעילים שאינן הוסברו במדריך זה.

ישנם המון הסברים ומדריכים באינטרנט על הנושא, תרגישו בנוח לחקור עוד באינטרנט ולגלות בעצמכם.

דוגמה למדריך של debug בintellij [בלינק הבא](https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html).

**בהצלחה!**