**פקודות קלט/פלט**

בשפת Java קיימות פקודות מובנות בעזרתן נוכל "להדפיס" (פלט) ולקרוא/לקלוט (קלט) מידע מהמשתמש.

**פקודות פלט**

שפת Java מאפשרת לנו 'להדפיס' ערכים ומידע מסוים לצורך תצוגה למשתמש, ממש תוך כדי ריצת התוכנית.

הכוונה ב'להדפיס' למשתמש, היא למעשה כתיבה לחלון הconsole בסביבת העבודה שלנו intellij או בכל פלטפורמה אחרת באמצעותה נריץ את התוכנית.

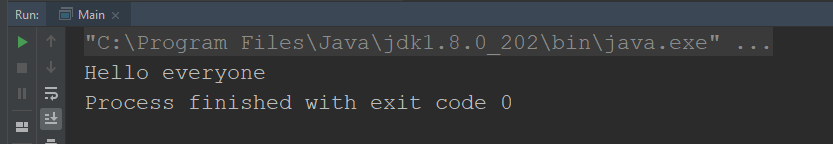
הדפסות אלו רצות **בזמן ריצת הקוד**.  
ניתן להדפיס גם ערכים של משתנים והערך שיודפס הוא ערך המשתנה בנקודת הזמן הנתונה.

נשתמש בפקודה הבאה עבור פלט:



לא מבינים את הsyntax ומה הוא אומר? מוזמנים לחקור עוד באינטרנט.

התוצאה שתודפס לנו בחלונית הconsole תהיה:

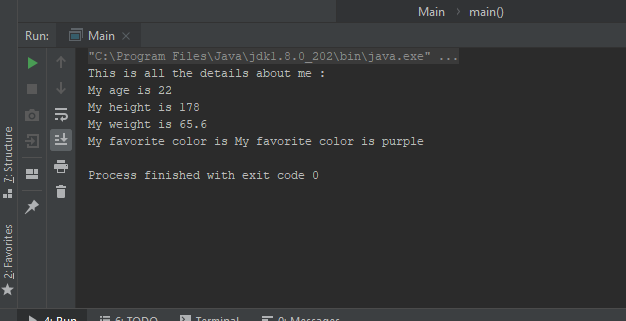


בנוסף, קיימת גם הפקודה **System.out.println** שמוסיפה ירידת שורה בסוף התוכן המודפס.

פקודה זו יותר נפוצה ותשמש אותנו לרוב בהדפסות הסטנדרטיות שנעשה.

דוגמאות לשימוש בהדפסות של והתוצאות שלהם: 

עבור אותן פקודות, נקבל את הפלט:



כך ניתן להדפיס תפריט למשתמש, מידע על משתנים, ולמעשה כמעט כל דבר שנרצה.

איך ניתן לכתוב System.out.println בצורה מהירה בלי שנצטרך לכתוב את כל הפקודה הזו ידנית? **(חפשו באינטרנט!)**

**פקודות קלט**

בפקודות קלט **נקרא** מידע מהמשתמש.

בכדי לעשות זאת אנו נשתמש במחלקת Scanner.  
(מחלקה שבאה מובנית בשפת Java, אומנם יש לייבא אותה כדי שנוכל להשתמש בה).

ראשית נייבא את מחלקת Scanner ע"י שימוש בפקודה הבאה:

import Java.util.Scanner;

יש לרשום את הפקודה בתחילת **הקובץ** Main שלנו  
(מעל תחילת הגדרת המחלקה)

פקודה זו מייבאת את היכולת המובנת בJava שמאפשרת לנו לקרוא דרך הconsole ערכים לפרויקט שלנו.

(**חידה:** למה שלא נייבא מראש את כל הפקודות והספריות הקיימות בJava? למה יש לייבא רק את מה שאנחנו משתמשים?)

בכדי להשתמש בScanner ולהתחיל לקרוא קלטים מהמשתמש דרך ה console, עלינו קודם כל ליצור 'אובייקט' Scanner.

לא נעמיק במשמעות ה'אובייקט' בשלב הזה, נחשוב עליו בתור משתנה מסוג Scanner אותו ניצור באופן הבא:

Scanner input = new Scanner( System.*in* );

כאמור השם input הוא שם שרירותי של המשתנה שבחרנו.

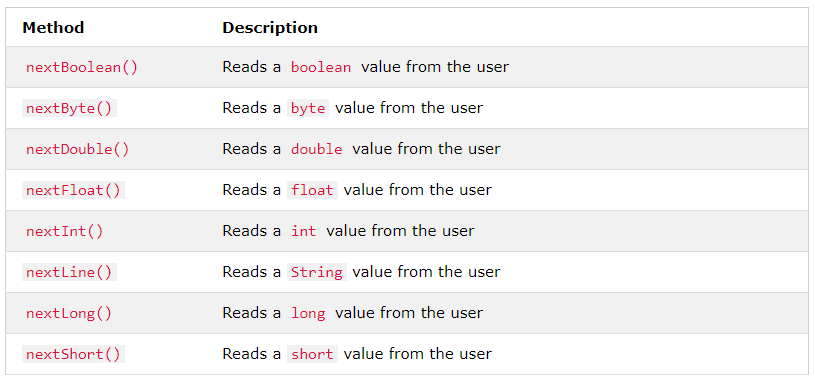
כעת בכדי לקלוט קלט מהמשתמש, נכתוב את הפקודה הבאה בנקודה שנרצה בקוד שלנו:

int age = input.nextInt();

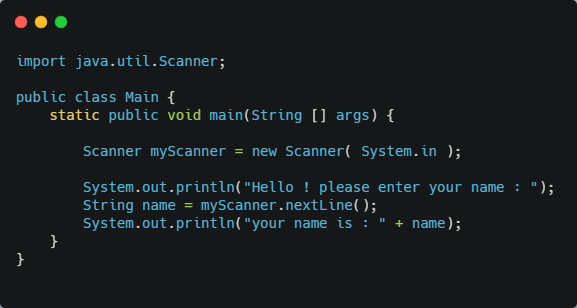
כשהתוכנית תרוץ, ריצת הקוד 'תיעצר' בשורה זו והתוכנית תחכה שהמשתמש ממש יכתוב ערך כלשהו בconsole.

לאחר הזנת הקלט מהשתמש ולחיצה על 'enter' , הקלט יושם למשתנה הרלוונטי (למשל בדוגמה שלעיל זה יהיה למשתנה age) ותהליך ריצת הקוד ימשיך כרגיל.

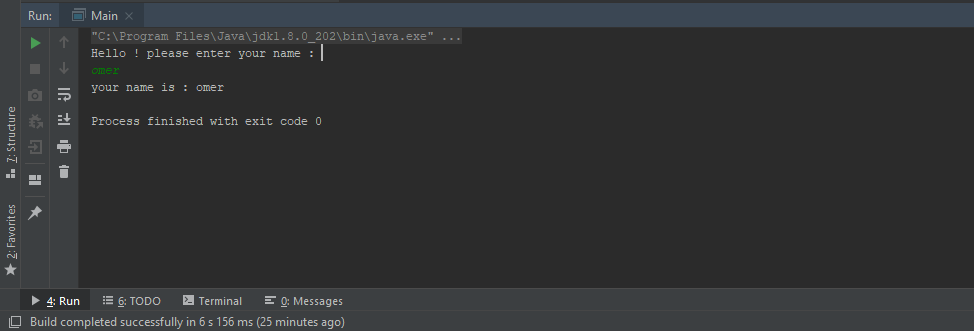
נשתמש בפקודות הבאות עבור כל סוג של ערך שאנחנו רוצים לקרוא מהconsole



דוגמה **לקריאה וכתיבה** דרך הconsole:

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&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=1x&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%250A%2520%2520%2520%2520%2520%2520%2520%2520Scanner%2520myScanner%2520%253D%2520new%2520Scanner(%2520System.in%2520)%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522Hello%2520!%2520please%2520enter%2520your%2520name%2520%253A%2520%2522)%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520String%2520name%2520%253D%2520myScanner.nextLine()%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520System.out.println(%2522your%2520name%2520is%2520%253A%2520%2522%2520%252B%2520name)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

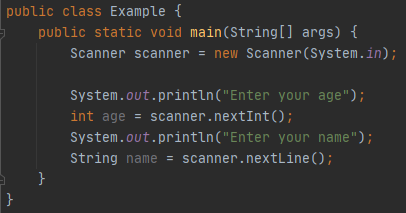
כך יראה הconsole: (הקלדנו את הקלט 'omer')



באותה מידה שקראנו קלט מסוג String, ניתן לקרוא קלט מסוג Int ,Double ,Boolean וכו'.

**קליטת מספרים ומחרוזות**

בואו ננסה להריץ את קטע הקוד הבא:



תוצאת הריצה תהיה:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

אם שמתם לב, קורה כאן משהו ממש מוזר, לא?

התוכנית "מדלגת" על קליטת המחרוזת וממשיכה הלאה בריצה שלה. אם נסתכל בערך של המשתנה name, נראה שיש בו מחרוזת ריקה.

**למה זה קורה?**

משהו שחשוב שנדע קודם, זה שלמקש "אנטר", או "שורה חדשה" יש ייצוג מאחורי הקלעים, והייצוג הזה נראה ככה: **\n**. כאשר אנחנו לוחצים אנטר, או רוצים לייצג שורה חדשה בהדפסה, ניתן להשתמש ב-**.\n**

בדוגמה שלנו, כאשר אנחנו מקלידים מספר, אנחנו כותבים 21, ולוחצים אנטר.

אבל, מה שנשמר בscanner מאחורי הקלעים, זה דווקא **\n21**.

למשתנה age ייכנס הערך 21, ולכן בscanner יישאר הערך **\n**.

הפקודה הבאה תהיה scanner.nextLine(), שכמו שאנחנו זוכרים, היא עובדת עד שהיא מקבלת אנטר, או במילים אחרות **\n** כמובן!

ומכיוון שהscanner שלנו לא ריק, וקיים בו הערך הנכסף **\n**, אז למשתנה של השם נכנס ערך ריק.

אם נצטרך להמחיז את זה, זה ייראה בערך כך:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**אז מה הפתרון לבעיה?**

אפשרות 1:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

להוסיף אחרי כל קליטת מספר, את הפקודה nextLine(), ובכך לרוקן את הscanner מה**\n**.

אבל בכל זאת, אנחנו תוכניתנים שאוהבים לחסוך בשורות קוד, לכן נעדיף להשתמש באפשרות היותר טובה:

אפשרות 2:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

בעת קליטת מספר, נקלוט אותו כמחרוזת, ונמיר אותו למספר בעזרת הפעולה **Integer.parseInt()**.

מה יקרה אם נרצה להמיר את המחרוזת לDouble, Float ועוד...? נוכל להשתמש בפעולה **Double.parseDouble()**, וכדומה.

**תרגול:**

* כתוב תוכנית **הקולטת** מהמשתמש (דרך הConsole)

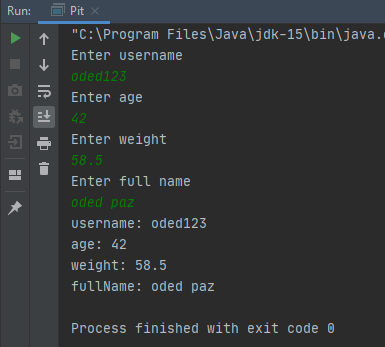
שם משתמש, גיל, משקל, ושם מלא.

יש לאחסן כל משתנה במשתנה מהסוג המתאים

ולאחר מכן יש **להדפיס** את כל הקלטים בחזרה למשתמש בצורה מובנת.

**יש לוודא שריצת התוכנית עובדת במלואה.**

דוגמה לריצת התכנית:



**בהצלחה!**