

המשתנה תו char

המשתנה תו char יכול להכיל תו/סימן אחד בלבד.

השמה למשתנה תו

char ch = 'a';
 השמת האות a במשתנה ch
 שים לב, יש לעטוף את התו בסימן גרש אחד מכל צד.

קלט למשתנה תו

קלט תו בודד:

```
System.out.print("a character --> ");
char ch = input.next().charAt(0);
/*    a character --> a
    ch1 = a
*/
```

ב- java לא קיימת פעולה מיוחדת לקליטת תו. הפעולה input.Next() קולטת מחרוזת. הפעולה charAt(i) היא פעולה של מחרוזת ומשמעותה – קח את התו במקום ה- i במחרוזת. התו הראשון במחרוזת נמצא במקום 0.

ומכאן שהוראת הקלט משמעותה: קח את התו במקום 0 ממחרוזת הקלט: input.next().charAt(0)

קלט תו בודד בתוך רצף:

כאשר מבקשים לקלוט תווים, בזה אחר זה, נוצרת בעיה בחלק מהתווים. למשל - לא ניתן לקלוט תו רווח (המחשב מתעלם מהקלט שהקלדנו, וממשיך להתעקש על קלט).

הפתרון: הוספת השורה הבאה אחרי הגדרת אובייקט הקלט:
 input.useDelimiter("");
 (בסוגריים – 2 גרשיים צמודים המציינים מחרוזת ריקה).

את הקלט נותנים בשורה אחת, כמו מחרוזת. אובייקט הקלט ידע לגשת לכל תו ברשימת התווים, בזה אחר זה. קטע תכנית לדוגמה הקולט רצף של תווים לתוך מערך של תווים:

```
Scanner input = new Scanner (System.in);
input.useDelimiter("");
char [] arr = new char [5];
System.out.print("enter 5 letters --> ");
for (int i = 0 ; i < arr.length ; i++)
    arr[i] = input.next().charAt(0);
```

הוקלדו 5 תווים, אחד מהם הוא תו רווח:

```
enter 5 letters --> ab cd
a b c d
```

השתמש במחרוזת ריקה כמפריד בין נתוני הקלט. התווים יוקלדו ברצף ללא רווח

מפריד ברירת המחדל הינו תו רווח. (כשקלטנו רצף מספרים הפרדנו ביניהם בתו רווח)

ניתן להחליט על תו אחר כמפריד בין נתוני הקלט, למשל פסיק:
 input.useDelomoter(",");

קלט עד לסוף שורת הקלט :

```
input.useDelimiter("");
while (ch != '\r')
{
    ch = input.next().charAt(0); // הקלט נעשה בתוך הלולאה
    ...
}
```

ייצוג תווים

תווים ב-Java נשמרים בטבלה. לכל תו יש מקום בטבלה. למשל: תו הרווח נמצא במקום 32 בטבלה הספרה 5 נמצאת במקום 53 בטבלה (*) האות A נמצאת במקום 65 בטבלה האות א' נמצאת במקום בטבלה

(*) שים לב! ספרה אינה מספר. הקשר שבין ספרה למספר הוא כמו הקשר שבין אות ומילה. כלומר - מספר מורכב מספרות כמו שמילה מורכבת מאותיות.

סדר התווים העיקריים:

המקום בטבלה		48		57		65		90		97		122		1491		1514	
התו		'0'	...	'9'		'A'	...	'Z'		'a'	...	'z'		'א'	...	'ת'	

אין צורך ללמוד בע"פ את מיקום התווים בטבלה, מספיק לזכור את הסדר:

- אותיות ABC קודמות לאותיות abc
- כל אות מה-abc נמצאת במרחק 32 מהאות במקבילה לה ב-ABC

השוואת תווים

ניתן להשוות תווים ע"י סימני השוואה המוכרים: == שווה, != שונה וכן סימני גדול, קטן וכו'.

הביטוי הלוגי: ch1 > ch2

יחזיר אמת אם התו שבמשתנה התווי ch1 נמצא אחרי התו שב- ch2 בטבלה, ושקר אם לא.

כדי לדעת אם התו שנקלט הוא ספרה, נוכל לבדוק אם התו נמצא בין '0' ו-'9':


```
if (ch >= '0' && ch <= '9') ...
```

התו ואיקונו

נוכל להתייחס לערך התווי בשני אופנים: כתו (char) ואז נקבל את התו, או כמספר (int) ואז נקבל את המיקום של התו בטבלה. דוגמה:

```
char ch = 'a';
System.out.print ("ch = " + ch + "\t" + (int)ch);

/*
   ch = a      ch = a   97
*/
```



פעולה חישובית (הוספה, החסרה) מתו, מתבצעת על מיקום התו בטבלה ותחזיר את המיקום החדש. אם רוצים את התו שנמצא במיקום החדש, יש לבצע המרה של התוצאה לטיפוס char:

```
char ch1 = 'A';           // המיקום בטבלה של 'A' הוא 65
int n = ch1 + 2;          // ב-n יהיה הערך 67
char ch2 = (char) n;      // לאחר ההמרה יהיה ב-ch2 התו 'C'

char ch2 = (char)(ch1 + 2); // ובהוראה אחת:
```

מחלקת שרות לתווים תרגיל

1. כתוב מחלקת שרות בשם Chars שתכיל את הפעולות הבאות :

ערך מוחזר	הפעולה	תאור הפעולה
boolean	isUpper (char ch)	מחזירה "אמת" אם התו הוא "אות גדולה" (ABC ...), ו-"שקר" אחרת
boolean	isLower (char ch)	מחזירה "אמת" אם התו הוא "אות קטנה" (abc ...), ו-"שקר" אחרת
boolean	isDigit (char ch)	מחזירה "אמת" אם התו הוא ספרה ו-"שקר" אחרת
char	toUpper (char ch)	מקבלת תו מסוג "אות קטנה" (abc ...) וממירה אותו ל"אות גדולה" (ABC...)
char	toLower (char ch)	מקבלת תו מסוג "אות גדולה" (ABC ...) וממירה אותו ל"אות קטנה" (abc...)
int	charToNum (char ch)	מקבלת ספרה ומחזירה את המספר התואם
char	nextChar (char ch)	מקבלת תו ומחזירה את התו העוקב לו
char	prevChar (char ch)	מקבלת תו ומחזירה את התו הקודם לו
char	nextLetter (char ch)	מקבלת אות (גדולה או קטנה) ומחזירה את האות ה"עוקבת מעגלית". (האות העוקבת מעגלית ל- Z היא A ול- z היא a)
char	prevLetter (char ch)	מקבלת אות (גדולה או קטנה) ומחזירה את האות ה"קודמת מעגלית". (האות הקודמת מעגלית ל- A היא Z ול- a היא z)

2. התרגילים הבאים נועדו לבדוק את מחלקת השרות שכתבתם.

- א. כתוב תכנית הקולטת מסר מוצפן ומפענחת אותו. ההצפנה תהיה החלפת כל תו בתו העוקב לו מעגלית במרחק של תו אחד. הקלט יהיה רצף של תווים המסתיימים במקש Enter. הפלט יהיה המסר המוצפן והפיענוח.
- ב. כתוב תכנית הקולטת מסר מוצפן ומפענחת אותו. ההצפנה תהיה החלפת כל תו בתו העוקב לו מעגלית במרחק n תווים אחד. הקלט יהיה מספר שלם חד ספרתי n ואחר כך רצף של תווים המסתיימים במקש Enter. הפלט יהיה המסר המוצפן והפיענוח.
- ג. כתוב תכנית שתקלוט הודעה המכילה אותיות (באנגלית), ספרות ותווי רווח. שרשר את התווים למחרוזת באופן הבא: עבור כל תו בקלט: אם הוא אות גדולה, יש להחליפו באות קטנה מתאימה. אם הוא אות קטנה, החלף אותה באות גדולה מתאימה. אם ספרה יש להדפיס את ערכה לפי הערך המספרי שלה (למשל, עבור '3' יודפס 333. תו רווח יישאר במקומו ללא שינוי.