

דף עזר לתלמיד - שפת Python גרסה 3

מבנים בשפה

if <תנאי>: הוראות לביצוע	משפטי תנאי
if <תנאי>: הוראות לביצוע else : הוראות לביצוע	
if <תנאי>: הוראות לביצוע elif <תנאי>: הוראות לביצוע else : הוראות לביצוע	
while <תנאי>: הוראות לביצוע	לולאות
for index in range(start, end, step): הוראות לביצוע	
הגדרת פעולה בשם func שאינה מקבלת פרמטרים: def func (): הוראות לביצוע הגדרת פעולה בשם func שמקבלת פרמטרים: def func (param1, param2, ...): הוראות לביצוע	פעולות

אופרטורים

- + * / ** // %	אופרטורים חשבוניים
-= += *= /= **= //= %= לדוגמה: x += 2 שקול כמו x = x+2	אופרטורים חשבוניים מקוצרים (כולל השמה)
> < >= <= == !=	אופרטורים לוגיים

הוראות קלט - פלט

פעולה הקולטת מחרוזת מהמשתמש: <pre>string = input('Enter your input:')</pre> ניתן להסב (casting) משתנים מטיפוס אחד לאחר. לדוגמה: פעולת הסבה של מחרוזת הנקלטת לטיפוס מספר שלם (integer) <pre>number = int(input('Enter your integer number:'))</pre> ניתן לקלוט ערך שלם מהמשתמש ולשלב בפעולה חישוב מתמטית, כמו צבירה לערך קיים. לדוגמה: <pre>total = total+ int(input('Enter your integer number:'))</pre>	קלט
פעולת הדפסה למסך: <pre>print ('your output!')</pre> <pre>print (<שם משתנה>)</pre>	פלט
<pre>var = int(var)</pre> <pre>var = float(var)</pre> <pre>var = str(var)</pre>	המרה

פעולות שימושיות מתוך ספריות

import math במידה והפעולה היא פעולת ספרייה (כגון sqrt)	ייבוא ספרייה מתמטית
<pre>n = abs(number)</pre> <pre>n = max(arg1, arg2, arg3, ...)</pre> <pre>n = min(arg1, arg2, arg3, ...)</pre> <pre>n = math.sqrt(number)</pre>	פעולות מתמטיות
import random	ייבוא ספריית random
הגרלת מספר אקראי לא שלם בין 0 ל - 1: <pre>num = random.random()</pre> הגרלת מספר אקראי (num) בטווח $start \leq num < end$ <pre>num = random.randrange(start, end)</pre> הגרלת מספר אקראי (num) בטווח $start \leq num \leq end$ (כולל 2 הקצוות): <pre>num = random.randint(start, end)</pre>	פעולות ליצירת מספר אקראי

מבנה נתונים - מחרוזות

משרד החינוך
הפיקוח להוראת מדעי המחשב בחט"ב

מחרוזות מחרוזת הינה סדרה של תווים (אותיות, סימנים, מספרים) מוקפים מצדם בגרש בודד או בגרשיים (מרכאות).	
msg = str() msg = ' '	אתחול מחרוזת ריקה
msg = 'Hi' note = 'a123\$'	אתחול מחרוזת עם ערך התחלתי
ch = msg[index]	החזרת איבר במחרוזת במיקום (index) מסוים (החל מ-0)
length = len (msg)	החזרת אורך מחרוזת
new_msg = msg + note >>>Hia123\$	שרשור מחרוזות (חיבור שלהן)
msg * num new_msg = msg * 3 >>>HiHiHi	כפל מחרוזות, במספר שלם חיובי num
msg[start: end: step] דוגמה: msg = 'Hello' new_msg = msg[: -1] >>>'olleH' new_msg = msg[1:3:] >>>'el'	החזרת מחרוזת חדשה על פי חיתוך מחרוזת קיימת start - מאיפה end - עד איפה, לא כולל step - בקפיצות
if ch in msg:	בדיקת האם תת-מחרוזת שבמשתנה ch קיימת במחרוזת msg
place = msg. find ('s')	החזרת מציין ראשון של המחרוזת s במחרוזת msg. אם לא קיים מחזיר -1
place = msg. index ('s')	החזרת מציין ראשון של המחרוזת s במחרוזת msg. אם לא קיים מחזיר הודעת שגיאה בזמן ריצה!
ch = 'a' ord(ch) >>>97	החזרת ערך אסקי של התו ch
val = 97 chr(val) >>>'a'	החזרת התו שהערך האסקי שלו הוא val

משרד החינוך
הפיקוח להוראת מדעי המחשב בחט"ב

msg. upper () msg. lower ()	החזרת מחרוזת חדשה עם אותיות גדולות / קטנות בהתאמה
msg. startswith (s, start, end) msg. endswith (s, start, end)	בדיקה האם המחרוזת msg במיקום start מתחילה / מסתיימת במחרוזת s
msg. replace (old_s, new_s)	החלפת המחרוזת old_s במחרוזת new_s במחרוזת msg
msg. replace (old_s, new_s, count)	החלפת המחרוזת old_s במחרוזת new_s במחרוזת msg מספר פעמים count
times = msg. count (ch)	ספירת מספר הפעמים שתת-המחרוזת שבמשתנה ch מופיעה במחרוזת msg

מבנה נתונים - רשימות

רשימות	
רשימה היא מבנה נתונים המורכב מאיבר אחד או יותר, המופרדים ביניהם בפסיקים.	
my_list = list () my_list = []	יצירת רשימה ריקה
my_list [index]	קבלת ערך במצוין (=במיקום) index
len (my_list)	מספר איברים ברשימה
my_list =[0,1] new_lst= my_list + [2,3,4] print (new_lst) >>> [0,1,2,3,4]	חיבור בין שתי רשימות
my_list. count (value)	החזרת כמות המופעים של value ברשימה
my_list. index (value)	החזרת המצוין הראשון ברשימה בו מופיע value אם לא קיים מחזיר הודעת שגיאה בזמן ריצה!
if item in my_list:	בדיקה האם item קיים ברשימה
My_list [start : end : step]	החזרת רשימה ע"פ חיתוך רשימה קיימת start - מאיפה end - עד איפה, לא כולל step - בקפיצות
my_list. append (item)	הוספת איבר item לסוף רשימה my_list
my_list. insert (pos, item)	הוספת איבר item במצוין/מיקום pos של הרשימה

משרד החינוך
הפיקוח להוראת מדעי המחשב בחט"ב

<code>my_list.remove(value)</code>	מחיקת ערך מהרשימה, לפי ערך (מוחק את המופע הראשון של value). אם לא נמצא ברשימה, מחזיר שגיאה.
<code>x = my_list.pop(index)</code> <code>x = my_list.pop()</code>	מחיקת ערך מהרשימה, לפי מיקום (מוחק את האיבר במקום ה-index), והחזרת ערכו. אם אין מיקום כזה, מחזיר שגיאה. מוחק את האיבר האחרון ברשימה.
<code>my_list.sort()</code> <code>my_list.sort(reverse=True)</code>	מיון רשימה בסדר עולה (מהקטן לגדול) מיון רשימה בסדר יורד (מהגדול לקטן)
<code>lst = ['This', 'is', 'a', 'test', 'List']</code> <code>lst.sort(key=str.lower)</code> <code>print(lst)</code> <code>['a', 'is', 'List', 'test', 'This']</code>	מיון רשימה ע"פ מפתח לדוגמה: המפתח <code>key=str.lower</code> ממין קודם את האותיות הקטנות, ואחריהן את האותיות הגדולות.
<code>lst = ['This', 'is', 'a', 'test', 'List']</code> <code>lst2 = sorted(lst, key=str.lower)</code> <code>print(lst2)</code> <code>print(lst)</code> <code>['a', 'is', 'List', 'test', 'This']</code> <code>['This', 'is', 'a', 'test', 'List']</code>	מיון רשימה, ללא שינוי הרשימה המקורית, מחזיר רשימה ממוינת ע"פ הבקשה.
<code>my_list.reverse()</code>	הפיכת סדר הפריטים ברשימה
<code>my_list = ['I', 'love', 'Python']</code> <code>delimiter = '_'</code> <code>s = delimiter.join(my_list)</code> <code>print(s)</code> <code>>>> 'I_love_Python'</code>	החזרת מחרוזת מתוך איברי הרשימה באמצעות הוספת delimiter בין האיברים
<code>s = 'I_love_Python'</code> <code>delimiter = '_'</code> <code>my_list = s.split(delimiter)</code> <code>print(my_list)</code> <code>>>> ['I', 'love', 'Python']</code>	החזרת רשימה חדשה באמצעות פיצול מחרוזת s לאיברים לפי delimiter
<code>my_list = s.split()</code>	החזרת רשימה חדשה באמצעות פיצול מחרוזת s לאיברים לפי רווח
<code>my_list = [0,1]</code> <code>new_lst = my_list.extend([2,3,4])</code> <code>print(new_lst)</code>	חיבור בין שתי רשימות

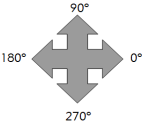
משרד החינוך
הפיקוח להוראת מדעי המחשב בחט"ב

>>> [0,1,2,3,4]	
-----------------	--

מבנה נתונים - תור

תור תור הוא מבנה נתונים המורכב מאיברים סדורים. לתור יש סדר הכנסה (מכניסים לסוף התור) וסדר הוצאה (מוציאים מראש התור). סדר זה נקרא FIFO = First In First Out כלומר: הראשון שנכנס הוא הראשון שיוצא.	
<code>import queue</code>	קריאה לספריה של תור
<code>q = queue.Queue()</code>	יצירת-תור-ריק
<code>q.put(ערך_שנרצה_להכניס_לתור)</code>	הכנסה-לתור
<code>q.get()</code>	הוצאה-מהתור
<code>q.empty()</code>	בדיקה-האם-התור-ריק?
לא ממומש בספריה	בדיקת-ערך-בראש-התור

מודל Turtle

import turtle	קריאה למחלקה הגרפית	תחילת עבודה
wn = turtle.Screen()	יצירת מסך עבודה עבור צב	
ayer= turtle.Turtle()	ירת צב בשם player	
player.forward(steps) player.fd(steps)	התקדמות קדימה steps צעדים	פעולות תנועה
player.backward(steps) player.back(steps)	תקדמות אחורה steps צעדים	
ayer.goto(x, y)	קפיצה למיקום על המסך	
ayer.setx(x) ayer.sety(y)	שינוי מיקום של הצב - x שינוי מיקום של הצב - y	
= player.pos()[0] = player.pos()[1] : = player.xcor() = player.ycor()	קבלת ערכי מיקום הצב למשתנים	
ayer.circle(r)	ציור עיגול שלם ברדיוס r	
ayer.circle(r,a)	ציור עיגול חלקי (קשת a מעלות) ברדיוס r	
ayer.stamp()	חתימת צב במיקום הנוכחי	
ayer.left(alpha) ayer.lt(alpha)	 סיבוב צב שמאלה ב alpha מעלות (נגד כיוון השעון) יחסית לזווית נוכחית	
ayer.right(alpha) ayer.rt(alpha)	 סיבוב צב ימינה ב alpha מעלות (עם כיוון השעון) יחסית לזווית נוכחית	
ayer.setheading(alpha) 	סיבוב צב לזווית alpha. בזווית מוחלטת יחסית לזווית ה-0 מעלות. (זווית חיובית היא נגד כיוון השעון)	פעולות סיבוב
ayer.heading()	מחזיר את זווית הסיבוב הנוכחית	

משרד החינוך
הפיקוח להוראת מדעי המחשב בחט"ב

player.pendown() player.down() player.pd()	עט מטה – הצב יצייר מסלול כשיבצע הוראות	פעולות מאפייני צב
player.penup() player.up() player.pu()	עט מעלה – הצב לא יצייר מסלול כשיבצע הוראות	
player.pensize(num)	שינוי עובי העט שאיתו מציירים	
player.pencolor('blue')	שינוי צבע העט שאיתו מציירים	
player.color('red')	שינוי צבע הצב	
player.shape("turtle") 'turtle', 'arrow', 'square', 'circle', 'triangle', 'classic', 'blank'	שינוי צורת הצב הצורות הקיימות בשפה:	
player.hideturtle()	הסתר צב	
player.showturtle()	הצג צב	
player.isvisible()	האם הצב מוצג? (מחזיר True או False)	
player.turtlesize(size)	שנה גודל צב לערך size (הערה: כל גודל צב מיוצג כ 20 פיקסלים בציור)	
wn.onkeypress(func,key)	לחיצה על מקש מקלדת. יש להגדיר פעולה שקוראים לה בלחיצה. דוגמה: בלחיצה על מקש חץ שמאלה, נקרא לפעולה kL: wn.onkeypress(kL,'Left')	האזנה ללחיצת מקלדת
wn.onkeyrelease(func,key)	עזיבה של מקש מקלדת. יש להגדיר פעולה שקוראים לה בעזיבה. דוגמה: בעזיבה של מקש חץ שמאלה, נקרא לפעולה rL: wn.onkeyrelease(rL,'Left')	
wn.onclick(function_name)	בלחיצה על העכבר, קוראים לפעולה function_name, שנגדיר מראש.	האזנה ללחיצת עכבר
turtle.mainloop() wn.exitonclick()	הרצת פעולה ברקע. יציאה בלחיצת עכבר.	פקודות סיום