

הכרות ואובייקטים בסיסיים

הרעיון מאחורי חשוב סימבולי.
דוגמה: 0.5 , $1/2$, $\sin(3)$. גבולות, אינטגרלים לא מסויימים.
סינטקס שונה ב Matlab
דוגמה:

```
y:=sin(3);  
z:=sin(4);  
y
```

נקודה פסיק רק מפריד. נקודותיים לא מדפיס.
יורדים שורה עם `.shift-enter`
לביצוע החישוב `enter`

```
דוגמאות:  
1+2/7  
500!  
float(500!)  
DIGITS:=200: float(PI)  
ifactor(48)  
Matlab .sqrt(2)^4-4 להשוות עם Matlab  
reset()
```

אובייקטים:

אובייקטים מורכבים מאופרנדים ופעולות `operations` שמורים במבנה של עץ

```
ex:=2*a*sin(x/y);  
prog:=expree(ex)
```

עלים נקראים אופרנדים אטומיים, אחרת אופרנדים מורכבים.

```
nops(ex);  
op(ex,1);  
op(ex,2);  
op(ex,3);  
op(ex,[2,1])
```

יש אופרנדים מסוגים שונים

```
domtype(2);  
domtype(2/3);  
domtype(0.5);  
domtype(FALSE);  
domtype(ex)
```

פעולות:

```
+, -, *, /, ^,  
x div y, x mod y  
abs, ceil, fact, float, floor, sign, sqrt, and, or, not  
_plus(x,y), _mult, _power
```

identifiers משתנים
כבררת מחדל מילה שאינה שמורה היא משתנה

```
reset();  
domtype(x);  
x:=1+I;  
domtype(x);
```

מחרוזות:

```
s:="this is a string":  
domtype(s);  
s2:="also this":  
s.s2; שרשור  
s[3..6]
```

שרשור עובד גם בשמות משתנים

```
k:=4:  
y.k:=99:  
y4
```

פונקציות:

```
reset();  
f:=-x^2:  
domtype(f)  
g:=x->-x^2: פרוצדורה  
domtype(g);  
f(2);  
g(2)
```

```
h:=(x,y)->x^2+y^2:  
h(1,2)
```

```
myabs:=x->if x>=0 then x else -x end_if:  
myabs(3);  
myabs(-3)
```

```
g2:=x->x+2;  
w:=g2@g; הרכבה  
w(2)
```

```
u:=x-->f: הפיכת פונק' למפה  
u(2)
```

Boolean משתנים בולאנים
TRUE, FALSE, UNKNOWN: 3 אפשרויות

```
TRUE or UNKNOWN;  
TRUE and UNKNOWN;  
not bool(1=2);  
not bool(1<>2)
```

sets קבוצות
סדר וחזרות לא חשובים

```
set:={a,1,3,x->4};  
nops(set);  
set2:={a,3};  
set union set2;  
set intersect set2;  
set minus set2;  
contains(set,1);  
g:=x->-x^2;  
map(set2,g);  
g2:=x->bool(x>0);  
a:=-4;  
select(set2,g2);
```

sequences סדרות
רשימה מסודרת של איברים מופרדים בפסיק

```
seq:=1,a,TRUE;  
domtype(seq);  
nops(seq);  
op(seq,3);  
seq[3];  
seq2:=i^2 $ i=1..6;  
2 $ 10; 2 סדרה עם 10 פעמים 2  
sin(x) $ x in [0,PI,2];  
seq3:=seq,seq2; שרשור  
null();  
seq[2]:=2,3: seq;  
delete seq[2];seq;  
max(seq2); min(seq2)
```

lists רשימות
סדרה בסוגריים. נחשב אובייקט יחיד.

```
lst:=[1,a,TRUE];  
domtype(lst);  
nops(lst);  
op(lst,3);  
lst[3];  
lst2:=[];  
lst3:=lst.[r,y]; שרשור  
[a,b,c]:=[1,2,3]; הצבה סימולטנית  
a:=1: b:=2: [a,b]:=[b,a]: a; b;  
contains(lst,1);  
map([x,1,0],sin);  
sort([4,7,-2]);  
select([4,7,-2],g2);
```

לולאות:

```
x:=1:
for i from 1 to 5 do
x:=x+i:
end_for;
y:=1:
for i from 1 to 5 step 2 do
y:=y+i:
end_for
```

תנאים:

```
if _____
then _____
elif _____ then _____
else _____
end_if
```

```
myabs:=x->if x>=0 then x else -x end_if:
myabs(3);
myabs(-3)
```

פרוצדורות: דוגמה

```
myFact:= proc(n)
begin
f:=1:
for i from 2 to n do
f:=f*i:
end_for;
return(f):
end_proc:
```

debugging:

notebook->debug מתוך פרוצדורה

תרגיל:

השערת גולדבך: כל מספר זוגי גדול מ 2 הוא סכום של שני ראשוניים.
אפשר להשתמש בפונקציה isprime
כתבו תוכנית הבודקת את השערת גולדבך עד מספר נתון n.

נגדיר פרוצדורה הבודקת האם זה נכון עבור מספר מסויים

```
test1:=proc(n)
begin
primes:=select([$2..floor(n/2)],isprime):
lst:=[n $ nops(primes)] - primes:
good:=select(lst,isprime):
return (bool(nops(good)>0)):
end_proc:
```

```
n:=500:
l:=[2*I $ i=2..floor(n/2)]:
select(l,not test1);
```

אלגברה

נתונה קבוצה Ω .

הגדרה: חיבור הינה פעולה בינארית $+: \Omega \times \Omega \rightarrow \Omega$ המקיימת

- אסוציאטיביות $(x+y)+z = x+(y+z)$.
- קומוטטיביות $x+y = y+x$.
- קיים אפס $\exists z \in \Omega, \forall x \in \Omega, x+z = x$. נסמן אותו ב 0 .
- קיים הופכי $\forall x \exists y, x+y = 0$. נסמן אותו ב $-x$.

הגדרה: כפל הינה פעולה בינארית $\cdot: \Omega \times \Omega \rightarrow \Omega$ המקיימת אסוציאטיביות וקיום יחידה. לא חייב להיות קומוטטיבי. לא חייב להיות הופכי.

הגדרה: קבוצה הסגורה תחת חיבור וכפל נקראת טבעת. ring .

הגדרה: טבעת עם כפל קומוטטיבי נקראת שדה. field .

דוגמאות:

- שלמים \mathbb{Z} Dom: Integer
- רציונאליים \mathbb{Q} Dom: Rational
- ממשיים \mathbb{R} Dom: Float, dom: Real
- מרוכבים \mathbb{C} Dom: Complex
- $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ Dom: IntegersMod(n)
- פולינומים P_n .
- מטריצות ריבועיות M_n Dom: SquareMatrix $M_n(\text{Ring})$ מעל טבעת (Ring)
- Dom: SquareMatrix(Dom: Integer)

דוגמאות:

```
constructor:=Dom::IntegerMod(7):
x:=constructor(3):
y:=constructor(5):
domtype(x);
x+y;
x*y;
x^123;
```

פולינומים:

```
p:=poly(1+a*x+3*x^2,[x]):
domtype(p);
p|x=2;    הצבה
degree(p,x);
coeff(p,2);
```

```
list:=[[1,0],[a,3],[b,5]];
p2:=poly(list,[x]);
poly2list(p);
```

```
p3:=poly(x+1,[x]):
p+p3;
p*p3;
divide(p,p3);
factor(p*p3);
gcd(p,p3);
D(p);    גזירה
int(p);  אינטגרציה
```

הגדרת פולינום מעל טבעת

```
p4:=poly(4*x+11,[x],Dom::IntegerMod(3)):
domtype(p4);
p4|x=2;
```

מטריצות:

```
reset():
A:=matrix([[1,2,3,4],
[a,b,c,d],
[sin(x),cos(x),exp(x),ln(x)]]);
Dom::Matrix();
v:=matrix([[x1],[x2],[x3],[x4]]);
A*v;
A[2,3];
A[2,3]:=4;
A[1..2,2..3];
transpose(A);
diff(A,x);
int(A,x);
map(A,x->x^2);  מפעיל מפה על כל האיברים
```

מטריצה מעל טבעת

```
constructor:=Dom::Matrix(Dom::Rational):
A:=constructor(2,3);
B:=constructor([[1,2,3],[1,2,3]]);
C:=constructor(2,3,(i,j)->i*j);
constructor(2,2,[11,12],Diagonal);
constructor::identity(2);
```

מטריצות רבועיות

```
constructor2:=Dom::SquareMatrix(2):
A:=constructor2([[0,y],[x^2,1]]);
```

```
domtype(A);
A^(-1);
exp(A);
```

תרגיל:

צרו את מטריצת הילברט בגודל 15: $H_{i,j} = \frac{1}{i+j-1}$

פתרו את המערכת $Hx = (1, \dots, 1)^T$ על ידי היפוך H .

פתרו את התרגיל עם מטריצות של מספרים רציונאליים ועם מטריצות float עם דיוקים שונים.

```
con:=Dom::Matrix(Dom::Rational);
H:=con(15,15,(i,j)->1/(i+j+1));
e:=con(15,1,1);
x:=H^(-1)*e;
```

לשנות שורה ראשונה ל

```
DIGITS:=100;
con:=Dom::Matrix(Dom::Rational);
```

פקודות נוספות:

```
M:=Dom::Matrix(Dom::Rational);
H:=con(3,3,(i,j)->1/(i+j+1));
M::col(H,2);
M::row(H,2);
M::delCol(H,2);
M::delRow(H,2);
M::matdim(H);
M::tr(H);
M::transpose(H);
M::identity(6);
A:=M(2,2,[[1,2],[3,1]]);
linalg::det(A);
linalg::charpoly(A,y);
linalg::eigenvalues(A);
M:=Dom::Matrix(Dom::Float);
A:=M(2,2,[[1,2],[3,1]]);
linalg::eigenvalues(A);
linalg::eigenvectors(A);
linalg::nullspace(A);
```

```
a:=matrix([[1,2],[3,5]]);
det(a);
numeric::det(a);
linalg::eigenvalues(a);
numeric::eigenvalues(a);
```

תרגיל: חשבו את ההופכי של המטריצה $2A + BB^T$ מעל הרציונאליים ומעל \mathbb{Z}_7 ,

$$A = \begin{pmatrix} 1 & 3 & 0 \\ -1 & 2 & 7 \\ 0 & 8 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 7 & -1 \\ 2 & 3 \\ 0 & 1 \end{pmatrix}$$

תרגיל: A מטריצה $n \times n$ עם אפס באלכסון ו 1 בשאר. מצאו את הדטרמיננטה, פולינום אופייני, ע"ע, והמרחב העצמי של כל ע"ע.

אינפי

```
f:=1/(exp(x^2)+1);
g:=diff(f,x);
int(g,x);
int(g,x=0..PI);
p:=exp(-x^2);
int(p,x=0..5);

simplify( (exp(x)-1)/(exp(x/2)+1) );
simplify( (cos(x))^2+(sin(x))^2 );

limit(sin(x)/x,x=0);
limit((1+1/n)^n,n=infinity);
sum(i,i=1..n);
sum(1/i^2,i=1..infinity);
product(i^3,i=1..n);

eqn:={x+y=a,x-a*y=b};
solve(eqn,{x,y});
solve(x*exp(x)=x,{x});
solve(x*exp(x)=x,{x},Real);
```

תרגיל: מצאו עבור אילו ערכים של a, b, c המטריצה

$$\text{אינה הפיכה.} \begin{pmatrix} 1 & a & b \\ 1 & 1 & c \\ 1 & 1 & 1 \end{pmatrix}$$

מספרים אקראיים

```
die := random(1..6):
die() $ i = 1..20

n := stats::normalRandom(1,3):
n() $ i = 1..20

u := stats::uniformRandom(1,3):
u() $ i = 1..20
```

תרגיל: כתוב פונק' המגרילה משתנה עם התפלגות ...

גרפיקה

```
plot(sin(x));
plot(sin(x)/x);
```

```

plot(sin(x)/x,x=-1..1);
plot(x^2+5*y^2,x=-3..3,y=-3..3,#3D);
plot(x^2-5*y^2,x=-3..3,y=-3..3,#3D);
plot(-x^2-5*y^2,x=-3..3,y=-3..3,#3D);

```

scatter plot

```

plot(plot::PointList2d([[1,1],[2,2],[3,3]]));
plot(plot::Polygon2d([[1,1],[2,4],[3,3]]));

plot(plot::PointList3d([[1,1,1],[1,2,2],[1,3,2],[1,3,4],
[2,1,1],[2,2,3],[2,3.5,4]],PointSize=5));
plot(plot::Polygon3d([[1,1,1],[2,4,2],[3,3,1]]));

```

פונקציות הנתונות בצורה סתומה

```

plot(plot::Implicit2d(x^3 + x + 2 = y^2,x = -5..5, y = -5..5));
plot(plot::Implicit3d(x^2+y^2+z^2=1,x=-2..2,y=-2..2,z=-2..2));
plot(plot::Implicit3d(x^2-y^2+z^2=1,x=-2..2,y=-2..2,z=-2..2));
plot(plot::Implicit3d(x^2-y^2-z^2=1,x=-2..2,y=-2..2,z=-2..2));
plot(plot::Implicit3d(-x^2-y^2-z^2=1,x=-2..2,y=-2..2,z=-2..2));

```

הצגה פרמטרית

```

plot([2*cos(t),sin(t)],t=0..2*PI):
plot([t*cos(t),t*sin(t)],t=0..2*PI):
plot([t-sin(t),1-cos(t)],t=0..6*PI) (עט מחובר לגלגל)

```

קואורדינאטות פולריות

```

plot(plot::Polar([1-cos(t),t],t=0..2*PI)): קרטואידה

```

עוד הרבה אפשרויות ...

תרגיל: כתוב תוכנית המגרילה עשר נקודות במישור, עושה להן התאמה לינארית ומשרטטת את הנק' וההתאמה