

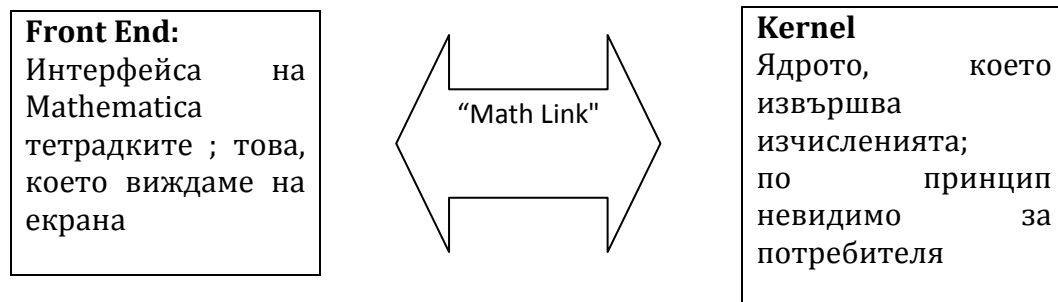
## Кратко въведение в Mathematica

*Mathematica* е програмен език от високо ниво , специално разработен за математици.

Тя е софтуерен пакет, предназначен за:

- нуждите на тъй нареченото техническо програмиране (изчислителна математика)
- компютърна система за символно смятане
- текстов редактор
- калкулатор
- база данни с научна информация
- инструмент за моделиране и анализ на данни
- интерпретиран език от високо ниво

Софтуера е изграден от два главни компонента:  
Front End и Kernel



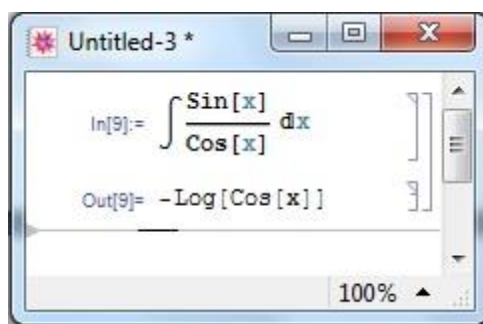
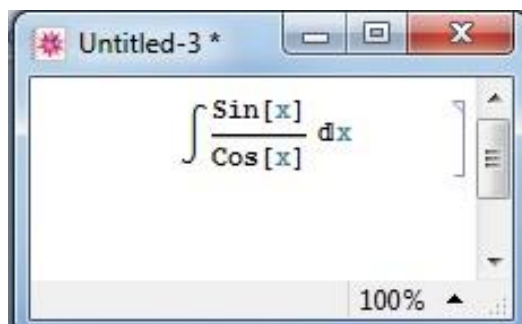
### Основни елементи в Mathematica:

#### 1.Тетрадка(Notebook)

Тетрадките са средата за работа в Mathematica. В тях въвеждаме кода и получаваме резултата от интерпретацията. Разширението на тетрадките е .nb. Нова тетрадка се създава от File/New/Notebook.nb или просто чрез комбинацията Ctrl+N. При стартиране на софтуера .

#### 2.Клетки(Cells)

Тетрадката се състои от клетки, които се конструират от самия потребител в каквато йерархия е нужно. В празна тетрадка има една клетка. Можете да направите нова клетка напрмер с клик под някакъв Input /Output стига курсура да е станал *хоризонтален* .



Всяка клетка в тетрадката се интерпретира( evaluate) **поотделно**. Това става с натискане на Ctrl+Enter докато се намирате в желаната клетка. Опцията Eevaluation/Evaluate Notebook интерпретира последователно цялата тетрадка.

### 3 Изрази (Expressions)

Израз наричаме всяка променлива , число, символ както и всяка символна последователност от вида

$$*(*,*,*....,*) ,$$

където на мястото на \* стоят други променливи или символи или изрази.

Оказва се, че в Mathematica

#### **ВСИЧКО Е ИЗРАЗ**

Тук тази “променлива”, която стои въвн от скобите обикновено е „глава на функция”

Пример:

Sin[x+y+z] , Sin е глава на функция, докато x+y+z е аргумента ѝ.

#### **ОСНОВНИ СИНТАКТИЧНИ ПРАВИЛА**

-езикът Mathematica е case-sensitive (прави разлика между главни и малки букви)

-имената на променливи не запичват с цифра , а с букви; Допълнително, вградените такива започват с главна буква;

-в име на променлива не може да има празен интервал , ако такъв има Mathematica интерпретира двата изрази като отделни имена.

-умножението в Mathematica става по много възможни начини.

A\*B , AB връщат произведението на A и B. Ако използвате втория начин , не забравяйте интервала, защото в противен случай Mathematica ще възприеме че трябва да върне стойността на променлива с име AB.

```
A = 4;
```

```
B = 3;
```

```
{A B, A * B, AB}
```

```
Out[21]= {12, 12, AB}
```

-Mathematica е и език за символно смятане и има много добра система за записване на изразте математически. За да използвате този запис може да използвате Palettes/Writing Assistant или Palettes/ Classroom Assistant или Palettes/ BasicMath Assistant, от където чрез кликуване върху съответния символ. Това, обаче, е бавен вариант и е добре да се свикне да се пише по следния начин:

\*Всички оператори, като интеграл, сума, дроб и т.н. могат бързо да се запишат чрез клавиатурата, чрез:

1. Посредством натискане на <Esc> , <въвеждане на съкращението на съответния оператор> ,<Esc> .

\*Освен променливи така можем и да пишем специални символи .

Примери:

<Esc> int <Esc> -> интеграл

<Esc> sum <Esc> -> сума

<Esc> e <Esc> неперово число

<Esc> pi <Esc> пи и други.

2. Посредством <Ctrl>+<клавиш>

<Ctrl>+</> -> дроб

<Ctrl>+<6> -> степен, горен индекс

<Ctrl>+<-> -> долен индекс и други.

В Mathematica следователно можем да изписваме голяма част от математическите формули (този стил на писане прилича идейно на този на езикът LaTeX)

Пример:

$\sum_{i=0}^{n-1} \int_i^{i+1} F'[x] dx = \int_0^n F'[x] dx = F[n] - F[0]$  е изписано на Mathematica.

Използване на скоби

1. {}

Къдравите скоби се използват за образуване на списъци(играят роля на стандартните масиви)

Пример: {a,b,c,{1,{b,c},2}}

2. [] използват използват се за да оградят аргументите на функция

MyFunction[a,b,c]

3. () Огражда изрази. Използват се за да се дава приоритет на дадени операции (също както в математиката скобите задават къде първо се прави пресмятането)

Пример:

Sin[pi/6] -> Резултат : 1/2

sin(pi/6)-> pi\*sin/6, където sin се интерпретира като променлива

Структурите от най-ниско ниво в Mathematica се наричат атоми. Има шест типа атоми: символи (Symbol), низове (String), цели числа (Integer), рационални дроби (Rational), реални числа (Real) и комплексни числа (Complex). Тогава можем да обобщим, че основните единици са: променливи (символи), текст и числа. От гледна точка на езика атомите имат само глава и нямат елементи, т.е. информацията, която съдържат, не се намира под формата на изрази (и в такъв смисъл не е спазен принципът за обобщеност). Символите представляват поредица от знаци (например: x, Sin, Exp, t23, a, b и т.н.), които чрез изразите влизат в някакви взаимовръзки, т.е. чрез тях ще изградиме зависимостите между

елементите в програмите си. String-овете са "мъртъв" текст, заграден в кавички (например: "Heu, heu!"), т.е. текст, който не води до зависимости.

Дефиниция на променливи:

При дефиницията на променливи няма нужда от предварителна декларация нито посочване на „тип“ както е в повечето езици. Можем също да предекларираме променлива :

Пример:

A=5;

A+3 -> резултат 8 ;

A={3,1}

A+3 -> резултат {6,4}

Забележете, че 3 се прибави към всеки всеки елемент на списъка.

### Работа със списъци:

#### 1) Създаване на списък:

Посредством функцията List или чрез директно изреждане във къдрави скоби.

Примери

a=List[1,1,2,3,5,8]; списък на числата 1,1,2,3,5,8

V={1,1,2,3,5,8}; списък на числата 1,1,2,3,5,8

C={{1,1},{2,2}} матрица с първи ред 1 1 и втори ред 2 2

D={{1},2}

лист с два елемента:

първия елемент е списък , с единствен елемент списък с единствен елемент 1;

втория представлява списък с единствен елемент 2

M={ {1,0,0},{0,1,0},{0,0,1} } единичната матрица

! За да визуализирате матрици използвайте

MatrixForm[<списък, имащ структура на матрица>]

Забележка: За удобство в Mathematica имаме много начини да прилагаме функции:

Ако „fun“ е глава на функция, а пък „a“ е променлива,

fun[a] , fun@a , a//Fun са еквивалентни

#### 2) Операции върху списъци:

2.1 събиране – желателно е да се събират се списъци с еднаква структура.

Събирането е покоординатно и се извършва с оператора „+“

Примери :

\*{1,2,3}+{-1,-2,1}={0,0,4}

\*{{1},1}+{{2},2}={{3},3}  
\*

```
In[38]:= M1 = {{1, 2}, {3, 4}};  
M2 = {{4, 5}, {6, 7}};  
M1 + M2 // MatrixForm
```

```
Out[40]//MatrixForm=  
( 5 7 )  
( 9 11 )
```

В някои случаи се позволява да се събират списъци с различна структура:

Пример:

$\{\{1\}, 2, \{\{3\}\}, \{\{\{4\}\}\}\} + 1 = \{\{2\}, 3, \{\{4\}\}, \{\{\{5\}\}\}\}$  (събиране на списък с число се интерпретира като прибавяне на числото до елементите с максимална дълбочина във всеки подсписък.

2.2 „умножение”

2.2.1 матрични умножения – използваме точка „.”

Примери:

Скалярно умножение на вектори – умножаваме списъци с дълбочина 1 и равен брой елементи:

$\{1, 2, 3\} \cdot \{1, 1, 1\} = 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 = 6$

Умножение на матрици с подходящи размерности

```
In[53]:= ( 1 2 ) . ( 1 0 1 ) // MatrixForm  
( 3 4 ) . ( 0 1 2 )
```

```
Out[53]//MatrixForm=  
( 1 2 5 )  
( 3 4 11 )
```

2.2.2 поелементно умножение- използваме „\*”.

$\{1, 2, 3\} * \{1, 1, 1\} = \{1 \cdot 1, 2 \cdot 1, 3 \cdot 1\} = \{1, 2, 3\}$

```
In[56]:= ( 1 2 ) * ( 1 2 ) // MatrixForm  
( 3 4 ) * ( 3 4 )
```

```
Out[56]//MatrixForm=  
( 1 4 )  
( 9 16 )
```

### 3) Извличане на елементи от списъци, работа с матрици

Ако имаме списък с име А, извличаме елементи от него по следния начин:

$A[[i1, i2, i3 \dots]]$

Тук броя на i-тата е най- много равен на максималната дълбочина на списъка

С i1 „влизаме” в i1-тата компонента на списъка А, с i2 достъпваме i2-тата компонентата на i1-тата компонента на списъка А ...

Примери:

$A = \{ \{1,2\}, 3, \{1, \{2\}\} \}$

$A[[1]] = \{1,2\}$

$A[[2]] = 3$

$A[[1,2]] = 2$

$A[[3,2,1]] = 2$

#### 4) Оператор Flatten[ <списък> , <естествено число n>]

Flatten сваля дълбочината на списъците на n.

Например Flatten[A,1] ,приложен към списъка A, дефиниран горе, връща {1,2,3,1,{2}};

Ако още веднъж приложим този оператор ще получим {1,2,3,1,2};

Можем да използваме и Flatten[A] без число – така премахваме всички вътрешни скоби.

Пример:  $B = \{ \{ \{ \{ 1 \}, 2 \}, 3 \} \}$ ; Flatten@B -> {1,2,3}.