

Лекція №4

Тема : Математичний пакет Mathematica



План лекції

1. Загальна характеристика пакету.
2. Основи роботи в пакеті.
3. Елементи програмування.

1. Загальна характеристика пакету.

Mathematica – система комп'ютерної алгебри, яка використовується в багатьох наукових, інженерних, математичних і комп'ютерних областях.

Спочатку система була придумана Стівеном Вольфрамом, в даний час розробляється компанією WolframResearch.

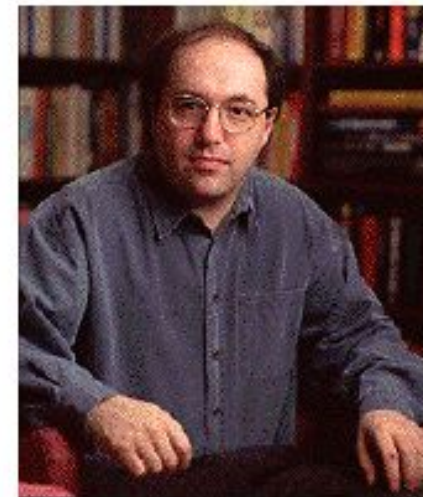
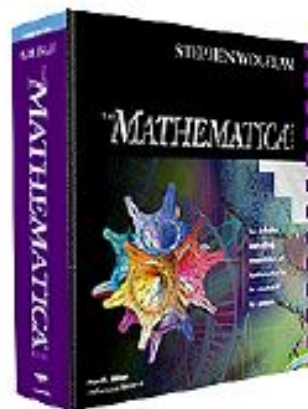
WolframMathematica – це програмне забезпечення, не тільки для математичних обчислень, це набагато більше: від моделювання та симуляції, візуалізації, документації, до створення веб-сайтів.

Mathematica володіє можливістю здійснювати виклики функцій і приймати виклики з C, .NET, Java та інших мов, генерувати C код, компілювати автономні бібліотеки і виконувати файли.

Перша версія Mathematica з'явилась 1 червня 1988 року.

- Перша версія - 1988 рік
- Друга версія - 1991 рік
- Третя версія - 1996 рік
- Четверта версія - 1999 рік
- П'ята версія - 2003 рік
- ...
- Десята версія - 2015 рік
- Одинадцята - 2018 рік
- ...

THE **MATHEMATICA** BOOK
STEPHEN WOLFRAM



НОВАЯ ВЕРСИЯ

MATHEMATICA 13.2

Лютий 2023 року

<https://www.wolfram.com/mathematica>

При розробці і експлуатації самих різних виробів використовується система Mathematica.



Система Mathematica складається з

- ядра **Mathematica Kernel** (обчислювальний механізм)
- зовнішньої оболонки **Mathematica** (візуальний інтерфейс).

Після установки пакета в головному меню створюються ярлики на два файли: Mathematica і Mathematica Kernel.

Справа в тому, що ярлик Mathematica Kernel запускає ядро пакету, яке робить всі обчислення, а ярлик Mathematica запускає інтерфейсну частину пакету
рис.1.

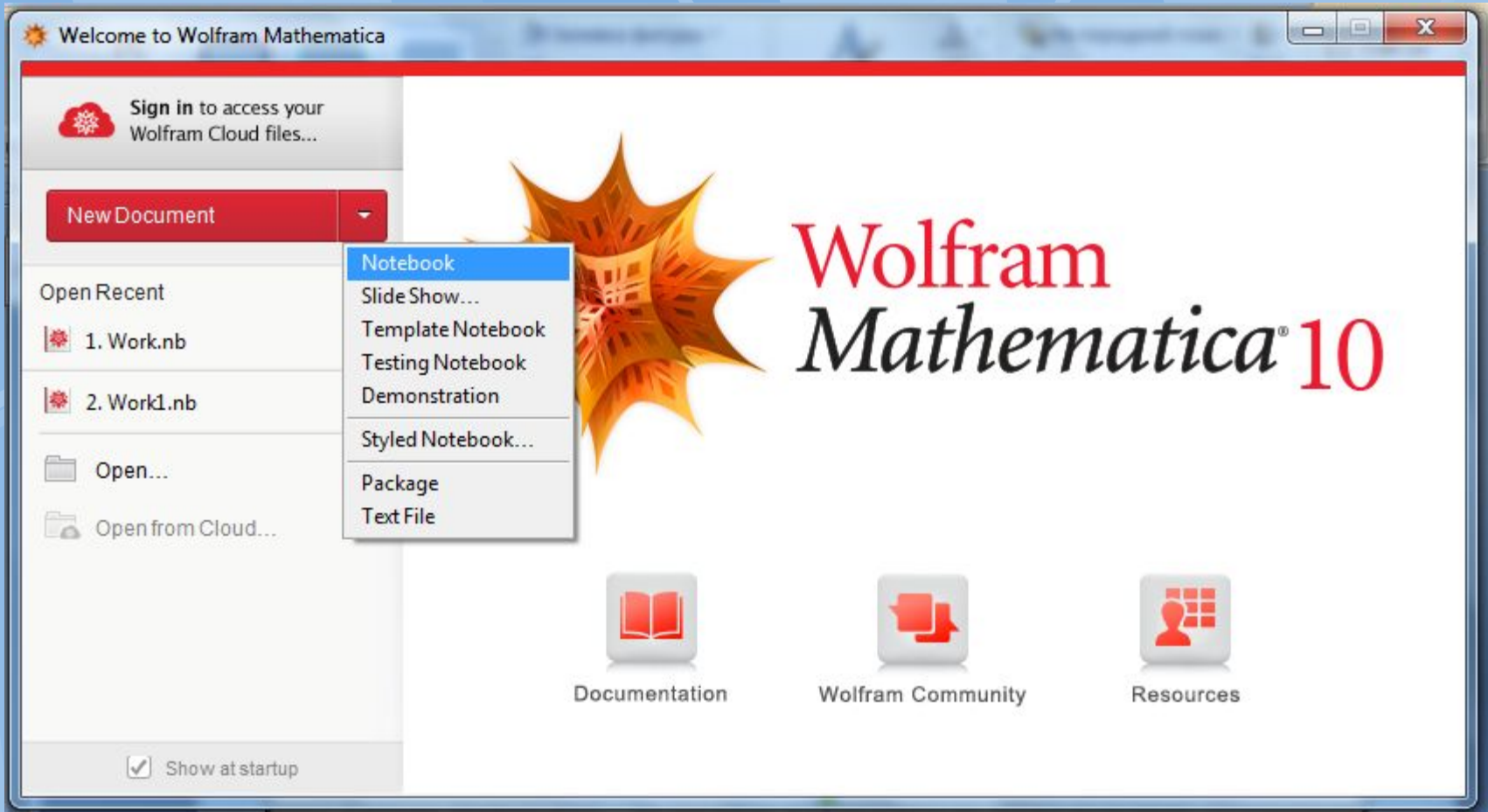


Рис.1.

Інтерфейс системи Mathematica реалізує відображення вікон, палітр, панелей інструментів, знаків і розташування їх у різному вигляді і в різних місцях екрану монітора. Головне вікно програми показано на рис.2.

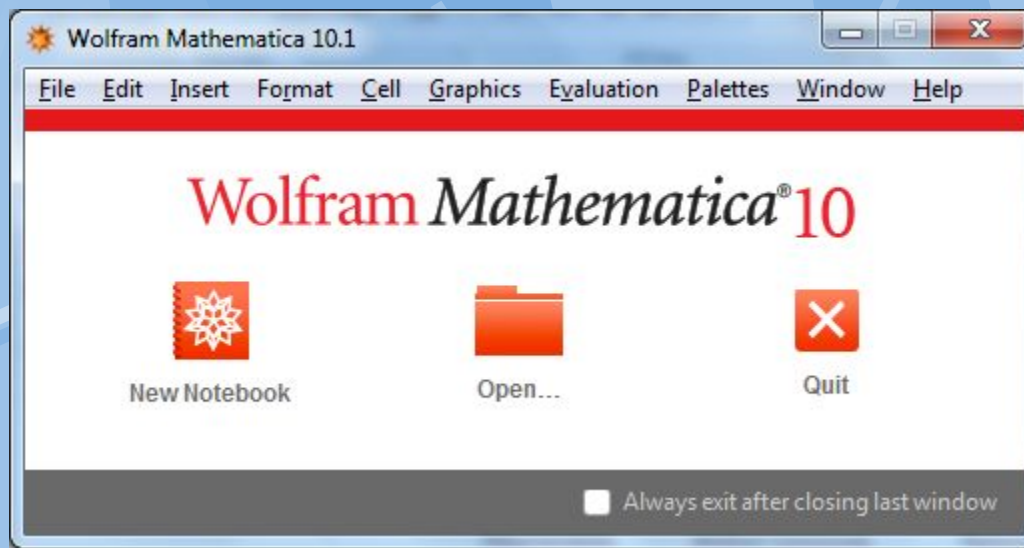


Рис. 2. Головне вікно

Вікно складається з основного меню програми (у верхній частині екрана), вікна робочого документа або «блокноту» (notebook) і панелі (палітри) для введення спец символів і знаків найбільш вживаних математичних операцій.

Основне меню програми містить кілька сотень найменувань пунктів меню, підменю, команд, функцій. Вивчити їх відразу неможливо: з короткого опису не можна зрозуміти зміст.

Зміст пунктів меню, підменю, команд можна зрозуміти тільки в процесі роботи з системою.

Після виконання команди **New Notebook**, з'являється вікно робочого документу рис.3.

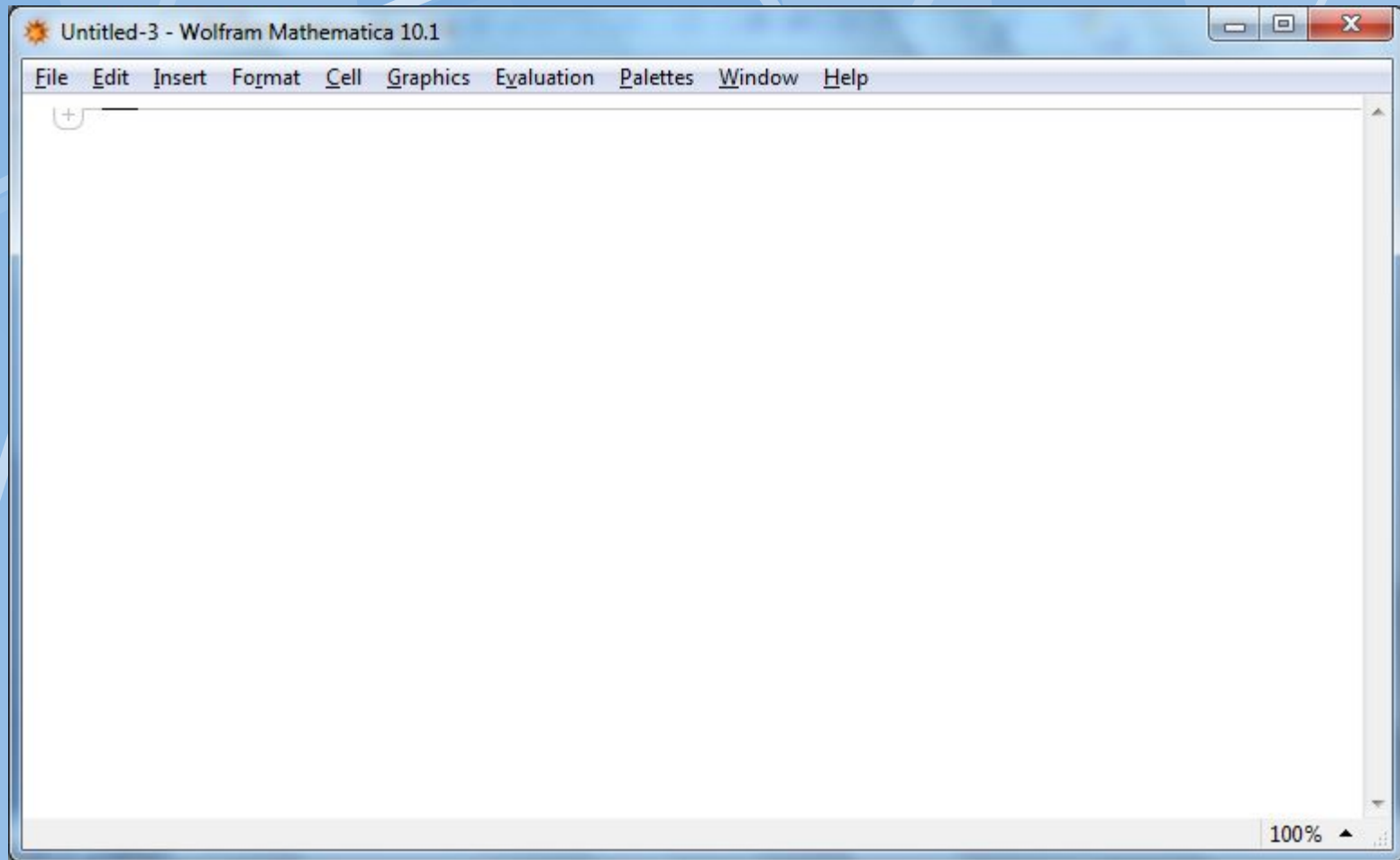


Рис.3.

Вікно робочого документа або блокнот складається з комірок. Комірку можна порівняти з параграфом у текстовому редакторі. Вся інформація, яка є в блокноті, зберігається в його комірках. Як тільки в порожньому новому файлі набирається хоча б один символ, Mathematica створить для нього комірку. Комірка також є мінімальною одиницею, яку можна обчислити.

Усі комірки можна розділити на три типи:

- **комірки введення** – в них задаються команди (формули), які будуть обчислені;
- **комірки результату** – у них Mathematica виводить результат обчислень;
- **не обчислювані комірки** – комірки з текстом, заголовки і все інше, що вводить користувач і обчислювати не потрібно.

Будь-які клітинки можна об'єднувати і розбивати за допомогою команд меню **Cell: Divide Cell** (розбити клітинку) і **Merge Cells** (об'єднати комірки).

Введення даних здійснюється в комірки. Пакет підтримує кирилицю і грецькі літери нарівні з англійським алфавітом. Можна називати змінні кирилицею, також як і грецькими. У той же час, ідентифікатори розрізняються по регістру (тобто змінна А не те саме, що змінна а).

Для швидкого доступу до функцій, розробники Mathematica ввели спеціальні типи вікон, які називаються палітрами. Палітри містять вікна з кнопками, які виконують дії.

Дії можуть бути абсолютно різними: від додавання грецької букви, до розкриття дужок у алгебраїчному виразі.

Різні палітри доступні через меню **Palettes** (рис.4).

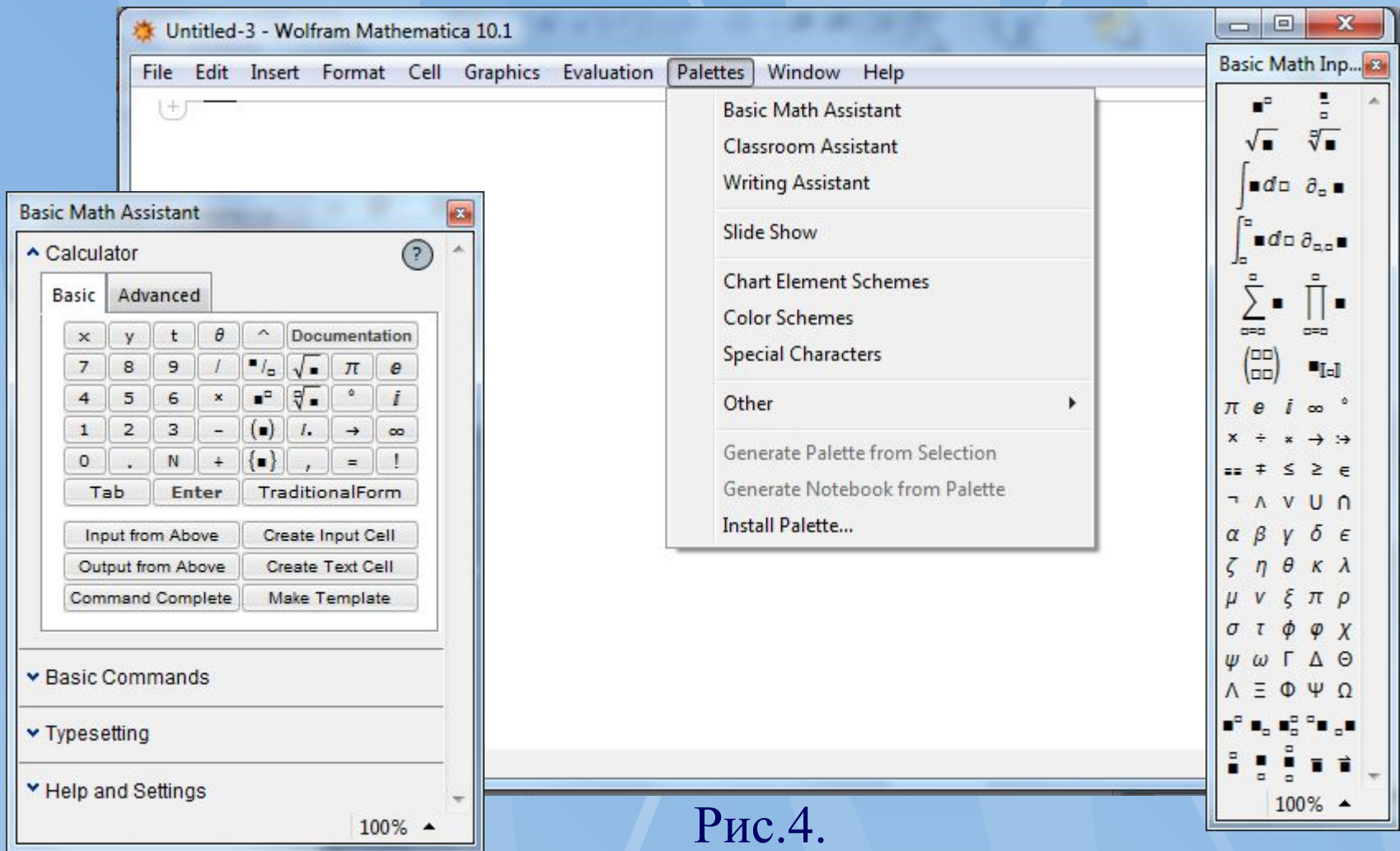


Рис.4.

Wolfram Mathematica має розвинені засоби форматування тексту. За допомогою їх можна розбивати блокнот на розділи, вводити пояснювальний текст і т.д. Стилї можна задати як всьому блокноту, так і окремій комірці цілком, або частково. Також можна змінити відображення всіх стандартних стилів і додати нові.

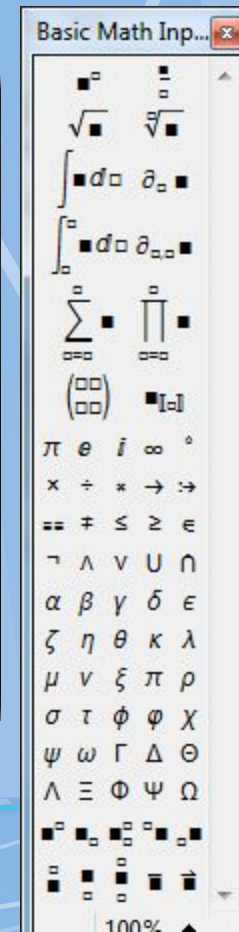
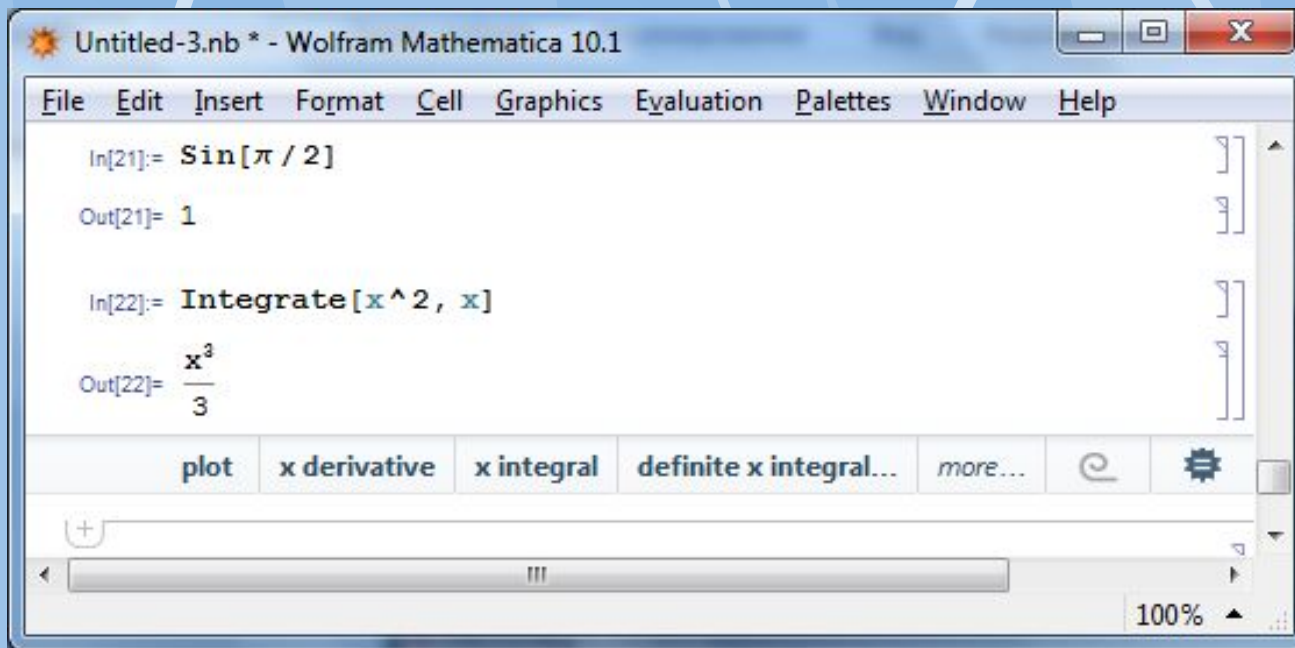
2. Основи роботи в пакеті.

Інтерфейс **Mathematica** =

= панель меню + робоча область + палітри

В робочій області вводяться команди.

Для їх виконання натискають **<Shift+Enter>**



Можливості пакету Mathematica

Пакет Mathematica дозволяє розв'язувати наступні задачі:

1. Виконувати алгебраїчні розрахунки та операції з виразами
2. Розв'язувати алгебраїчні рівняння та системи
3. Виконувати інтегрування та диференціювання
4. Розв'язувати диференціальні рівняння
5. Створювати програмні коди
6. Виконувати числові розрахунки
7. Створювати графіку та анімацію

Основні арифметичні оператори:

1. **+** додавання
2. **-** віднімання
3. ***** множення
4. **/** ділення
5. **^** піднесення в степінь

Додаткові оператори:

6. **%** результат виконання останньої команди
7. **%%** результат виконання передостанньої команди

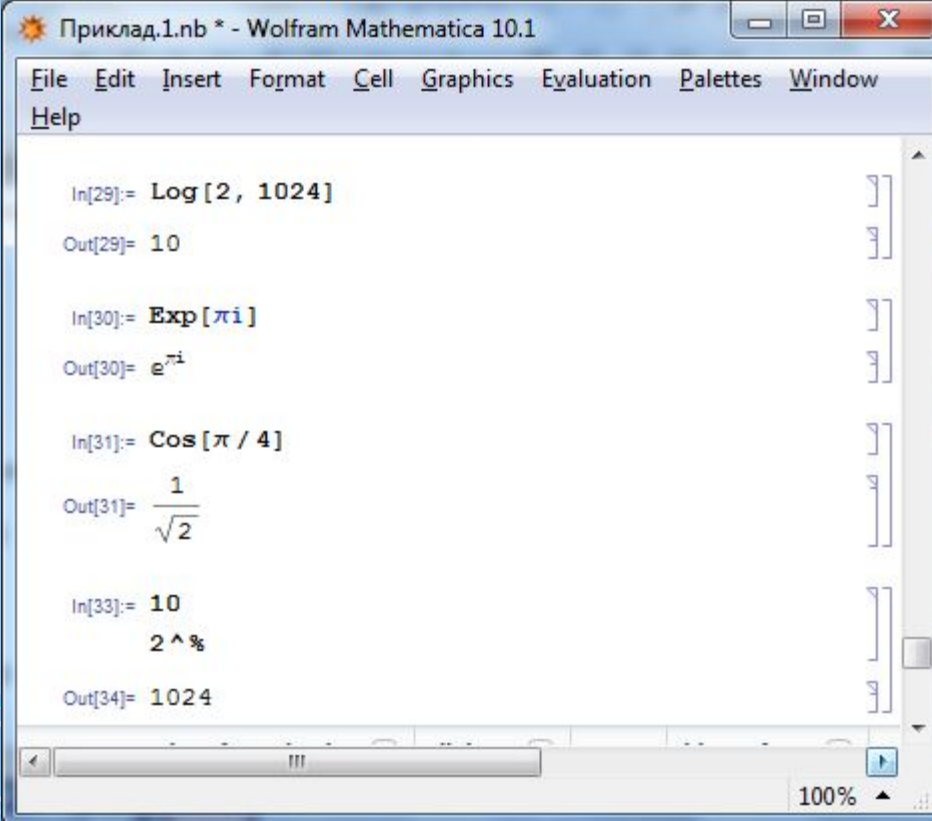
Функції

Основні правила:

1. Назва функції - з великої літери
2. Аргумент - в квадратних дужках
3. Аргументи розділяються комами

Наприклад:

1. Sin[x]
2. Log[a,b]
3. Exp[πi]



Приклад.1.nb * - Wolfram Mathematica 10.1

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
In[29]:= Log[2, 1024]
```

Out[29]= 10

```
In[30]:= Exp[ $\pi i$ ]
```

Out[30]= $e^{\pi i}$

```
In[31]:= Cos[ $\pi / 4$ ]
```

Out[31]= $\frac{1}{\sqrt{2}}$

```
In[33]:= 10
```

```
2 ^ %
```

Out[34]= 1024

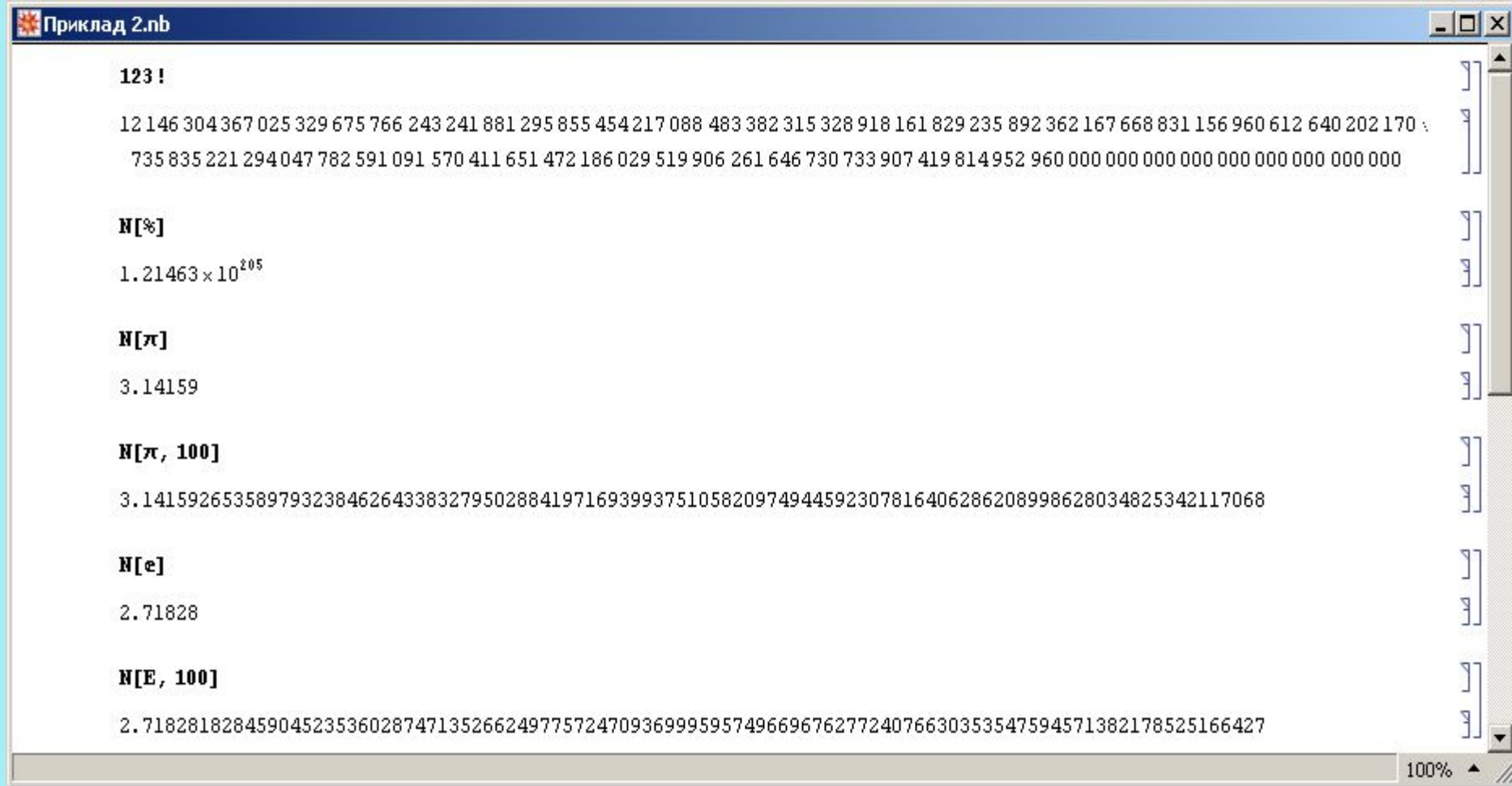
100%

Математичні функції

Функція	Опис	Функція	Опис
Abs[x]	Модуль комплексного числа x	Cot[z]	Котангенс від z
ArcCos[z]	Арккосинус від z	Coth[z]	Котангенс гіперболічний від z
ArcCosh[z]	Арккосинус гіперболічний від z	Csc[z]	Косеканс від z
ArcCot[z]	Арккотангенс від z	Csch[z]	Косеканс гіперболічний від z
ArcCoth[z]	Арккотангенс гіперболічний від z	Exp[z]	Експонента
ArcCsc[z]	Арккосеканс від z	Im[z]	Уявна частина z
ArcCsch[z]	Арккосеканс гіперболічний від z	Log[a,z]	Логарифм числа z за основою a
ArcSec[z]	Арксеканс від z	Re[z]	Дійсна частина z
ArcSech[z]	Арксеканс гіперболічний від z	Sec[z]	Секанс від z
ArcSin[z]	Арксинус від z	Sech[z]	Секанс гіперболічний від z
ArcSinh[z]	Арксинус гіперболічний від z	Sign[x]	Знак числа x
ArcTan[z]	Арктангенс від z	Sin[z]	Синус від z
ArcTanh[z]	Арктангенс гіперболічний від z	Sinh[z]	Синус гіперболічний від z
Arg[z]	Аргумент числа z	Sqrt[z]	Корінь квадратний від z
Conjugate[z]	Комплексно спряжене до z	Tan[z]	Тангенс від z
Cos[z]	Косинус від z	Tanh[z]	Тангенс гіперболічний від z
Cosh[z]	Косинус гіперболічний від z		

Приклади розрахунків - числа

Приклад 2. Числа



```
Приклад 2.nb

123!

12 146 304 367 025 329 675 766 243 241 881 295 855 454 217 088 483 382 315 328 918 161 829 235 892 362 167 668 831 156 960 612 640 202 170 \
735 835 221 294 047 782 591 091 570 411 651 472 186 029 519 906 261 646 730 733 907 419 814 952 960 000 000 000 000 000 000 000 000 000

N[%]

1.21463 × 10205

N[π]

3.14159

N[π, 100]

3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342117068

N[e]

2.71828

N[E, 100]

2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427

100%
```

$N[expr]$

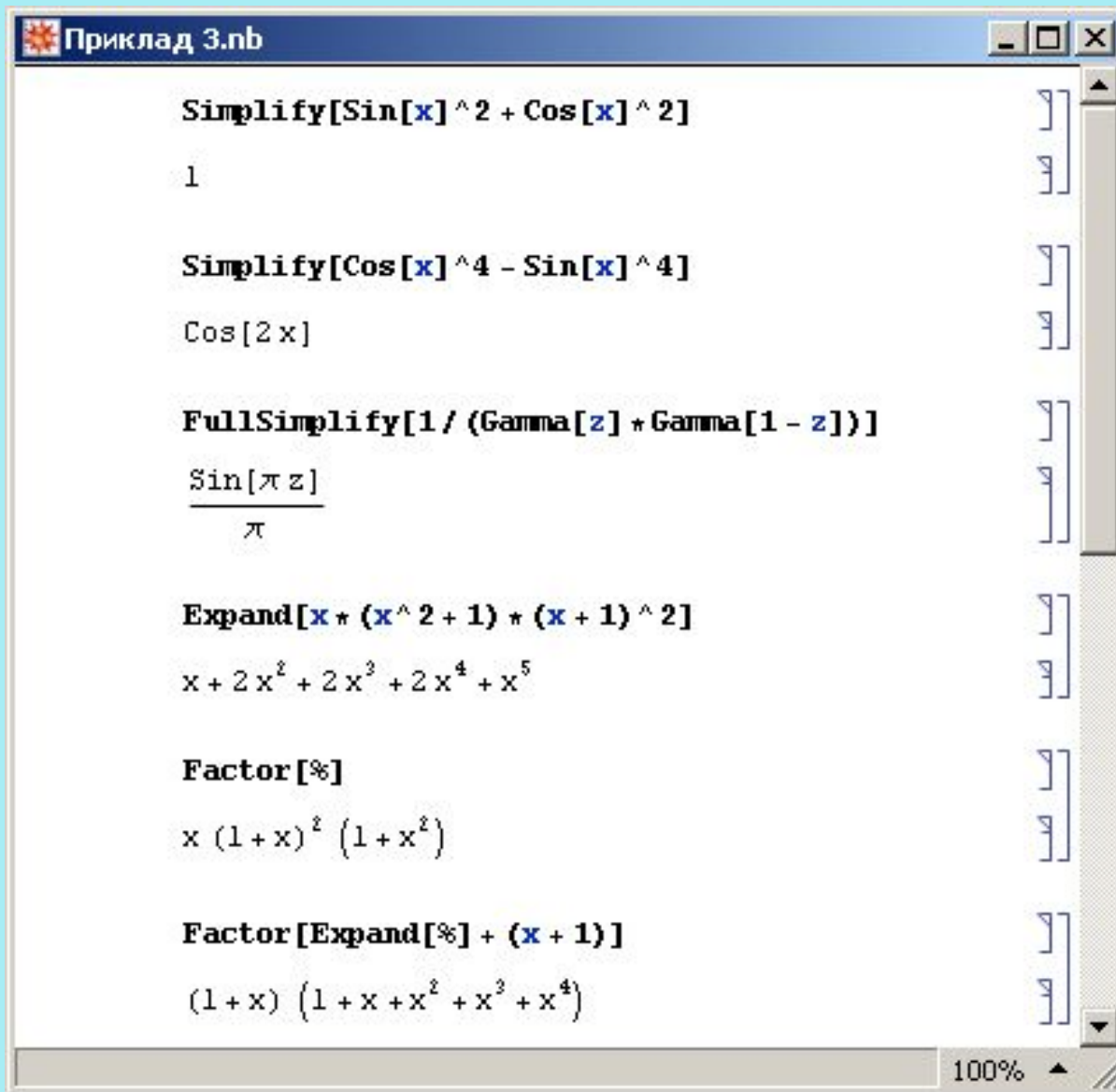
gives the numerical value of *expr*.

$N[expr, n]$

attempts to give a result with *n*-digit precision.

Приклади розрахунків - вирази

Приклад 3. Операції з виразами



```
Приклад 3.nb

Simplify[Sin[x]^2 + Cos[x]^2]
1

Simplify[Cos[x]^4 - Sin[x]^4]
Cos[2 x]

FullSimplify[1 / (Gamma[z] + Gamma[1 - z])]
Sin[π z]
π

Expand[x * (x^2 + 1) + (x + 1)^2]
x + 2 x^2 + 2 x^3 + 2 x^4 + x^5

Factor[%]
x (1 + x)^2 (1 + x^2)

Factor[Expand[%] + (x + 1)]
(1 + x) (1 + x + x^2 + x^3 + x^4)
```

Команди для спрощення виразів:

Simplify[]

FullSimplify[]

Команда для розкладу виразу:

Expand[]

Команда для згортки виразу:

Factor[]

Приклади розрахунків - змінні та функції

Приклад 4

```
Приклад 4.nb  
  
1 = 2 * π * R  
2 π R  
  
S = π * R^2  
π R^2  
  
R = 5  
5  
  
1  
10 π  
  
N[1, 10]  
31.41592654  
  
S  
25 π  
  
N[S, 12]  
78.5398163397
```

Приклад 5. Функції

```
Приклад 5.nb  
  
S[R_] := π * R^2  
1[R_] := 2 * π * R  
  
S[5]  
25 π  
  
1[5]  
10 π  
  
N[S[5], 10]  
78.53981634  
  
N[1[5], 12]  
31.4159265359
```

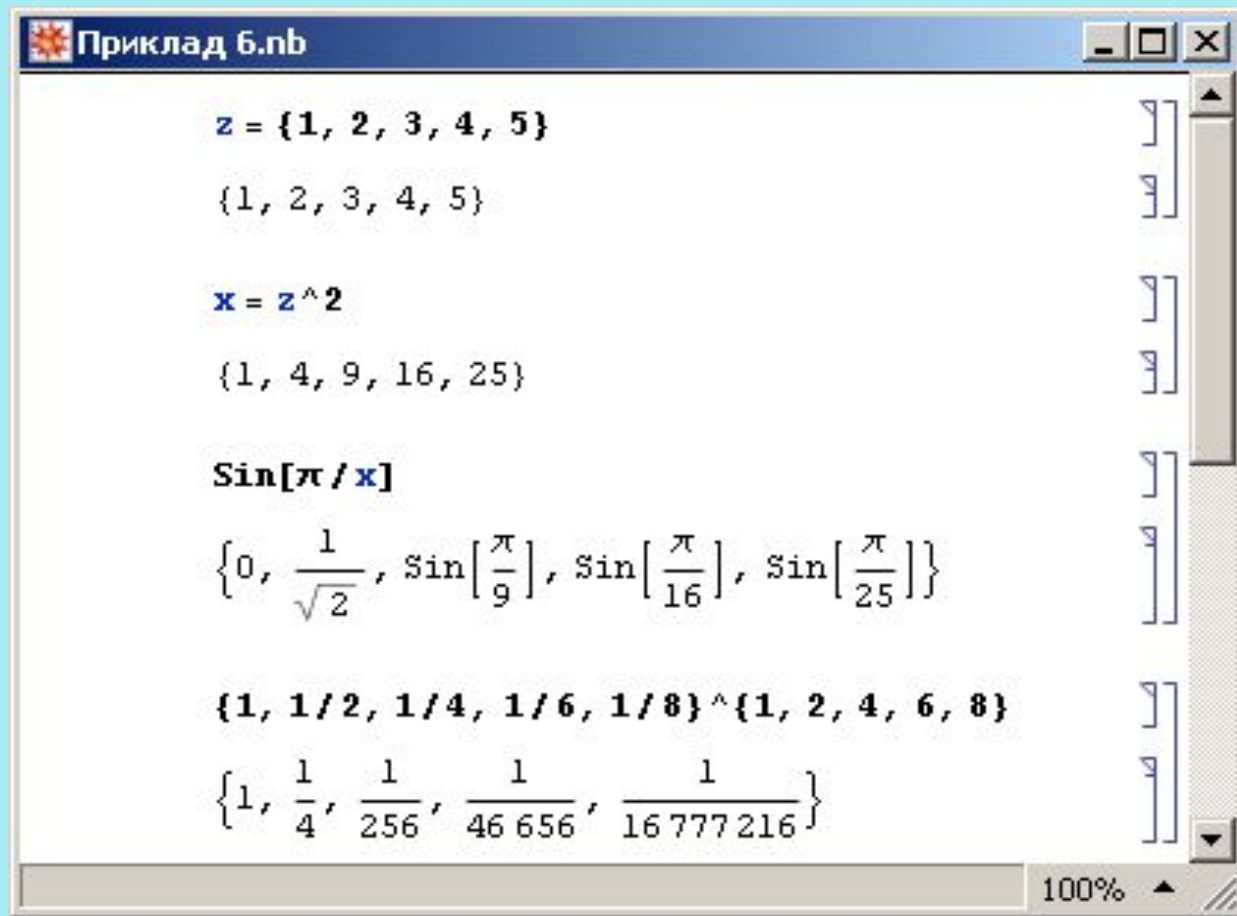
Зверніть увагу на різницю
у використанні операторів

= та :=

Списки

Список реалізується у вигляді набору елементів (взагалі різного типу), розділених комами і взятих у фігурні дужки. Порядок слідування елементів має значення.

Приклад 6. Списки



```
Приклад 6.nb

z = {1, 2, 3, 4, 5}

{1, 2, 3, 4, 5}

x = z^2

{1, 4, 9, 16, 25}

Sin[π / x]

{0, 1/√2, Sin[π/9], Sin[π/16], Sin[π/25]}

{1, 1/2, 1/4, 1/6, 1/8}^{1, 2, 4, 6, 8}

{1, 1/4, 1/256, 1/46656, 1/16777216}
```

100%

Побудова графіків на площині

У відношенні графіки система Mathematica є лідером серед систем комп'ютерної алгебри. Велика кількість опцій дозволяє оформляти графічні образи практично в будь-якому бажаному вигляді.

Графіки в системі Mathematica є об'єктами і тому вони можуть бути значеннями змінних.

Почнемо розгляд графічних можливостей системи з побудови найпростіших графіків функцій однієї змінної виду $y = f(x)$ або просто $f(x)$.

Графік таких функцій будується на площині, тобто в двовимірному просторі. При цьому використовується прямокутна (декартова) система координат. За замовчуванням будуються і лінії координатної системи.

Для побудови двовимірних графіків функцій виду $f(x)$ використовується вбудована в ядро функція **Plot**:

Plot [f , $\{x, x_{min}, x_{max}\}$] – повертає об'єкт, що представляє собою графік функції f аргументу x в інтервалі від x_{min} до x_{max} ;

Plot [$\{f1, f2, \dots\}$, $\{x, x_{min}, x_{max}\}$] – повертає об'єкт у вигляді графіків ряду функцій f_i .

Функція **Plot** використовується для побудови однієї або кількох ліній, що дають графічне представлення для зазначених функцій $f, f1, f2$ і т. д.

Приклади застосування функції **Plot** показані на рис.5,6.

Зауважимо, що графіки побудовані без використання будь-яких опцій (точніше, з набором опцій за замовчуванням).

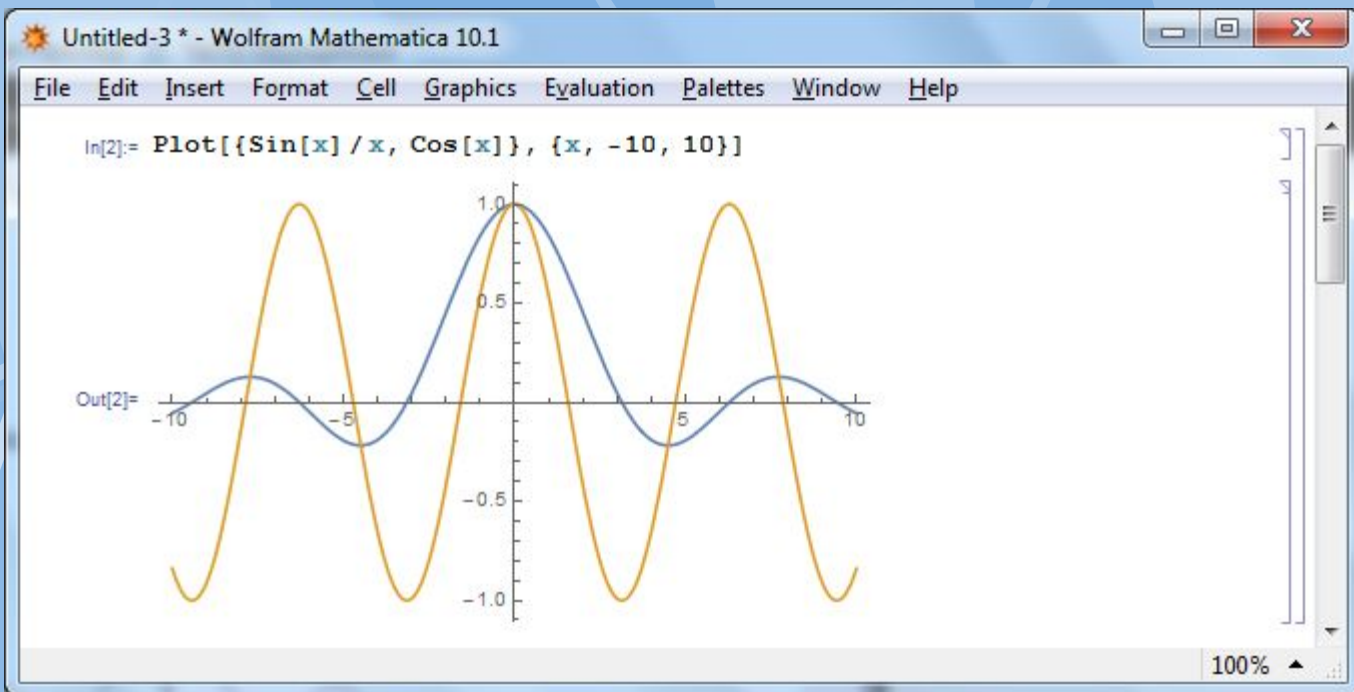


Рис.5.

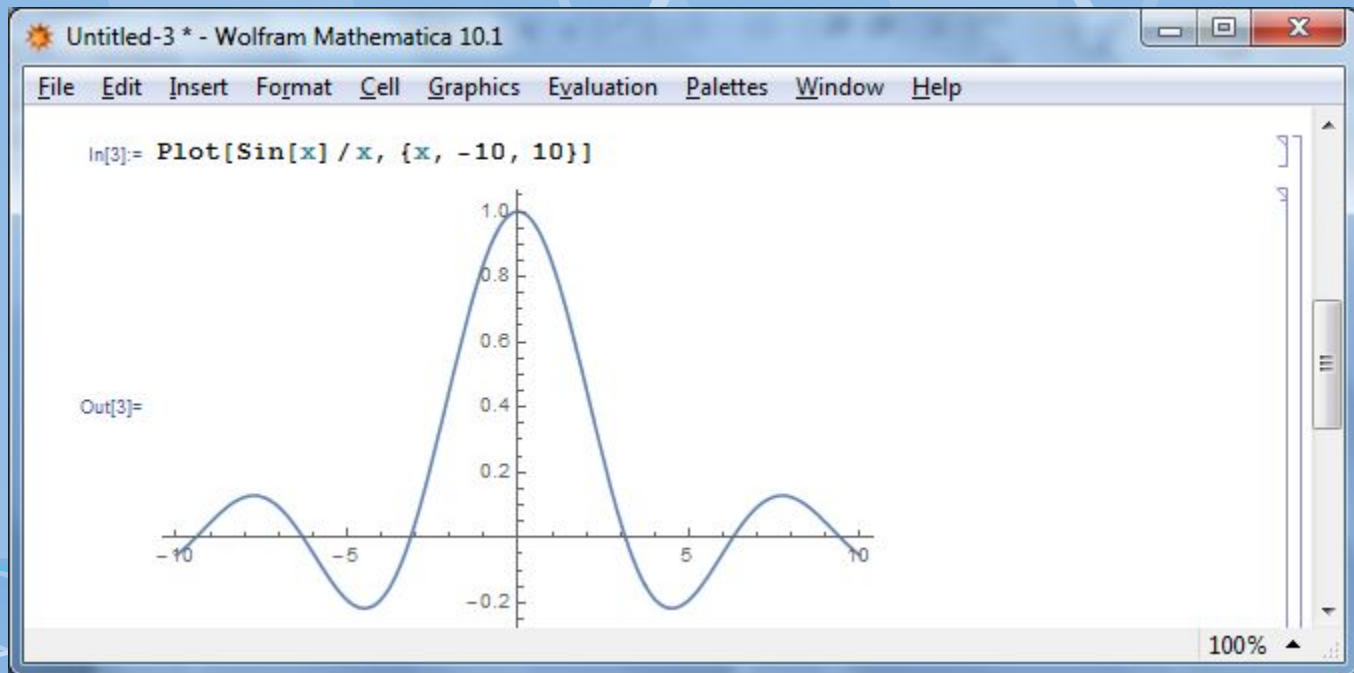


Рис.6.

В міру ускладнення задач користувачеві рано чи пізно перестануть влаштовувати графіки, одержувані при автоматичному виборі їх стилю та інших параметрів.

Для точного налаштування графіків Mathematica використовує спеціальні опції графічних функцій. Для виведення їх списку треба використовувати команду : **Options [Plot]**.

Ще одним важливим засобом настроювання графіків є графічні директиви. Синтаксис їх подібний синтаксису функцій. Однак директиви не повертають об'єктів, а лише впливають на їх характеристики.

Застосування графічних директив спільно з опціями дозволяє створювати графіки самого різного виду.

Опції задаються своїм іменем `name` і значенням `value`
name -> value

Символьні значення опцій:

Automatic — використовується автоматичний вибір;

None — опція не використовується;

All — використовується в будь-якому випадку;

True — використовується;

False — не використовується.

Багато опцій можуть мати числові значення.

Options[Plot]

```
{AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ ,  
Axes → True, AxesLabel → None, AxesOrigin → Automatic,  
AxesStyle → {}, Background → None, BaselinePosition → Automatic,  
BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic,  
ColorFunctionScaling → True, ColorOutput → Automatic,  
ContentSelectable → Automatic, CoordinatesToolOptions → Automatic,  
DisplayFunction := $DisplayFunction, Epilog → {}, Evaluated → Automatic,  
EvaluationMonitor → None, Exclusions → Automatic,  
ExclusionsStyle → None, Filling → None, FillingStyle → Automatic,  
FormatType := TraditionalForm, Frame → False, FrameLabel → None,  
FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {},  
GridLines → None, GridLinesStyle → {}, ImageMargins → 0.,  
ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic,  
LabelStyle → {}, MaxRecursion → Automatic, Mesh → None,  
MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic,  
Method → Automatic, PerformanceGoal := $PerformanceGoal,  
PlotLabel → None, PlotLegends → None, PlotPoints → Automatic,  
PlotRange → {Full, Automatic}, PlotRangeClipping → True,  
PlotRangePadding → Automatic, PlotRegion → Automatic,  
PlotStyle → Automatic, PlotTheme := $PlotTheme,  
PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &),  
RotateLabel → True, TargetUnits → Automatic, Ticks → Automatic,  
TicksStyle → {}, WorkingPrecision → MachinePrecision}
```

```
{СоотношениеСторон = AspectRatio, ЗолотоеСечение = GoldenRatio,  
Оси = Axes, Да = True, ПодписиОсей = AxesLabel, Никакой = None,  
ПересечениеОсей = AxesOrigin, Авто = Automatic,  
СтильОсей = AxesStyle, Фон = Background,  
СтильОбрезки = ClippingStyle, ЦветоваяФункция = ColorFunction,  
МасштабированиеЦветовойФункции = ColorFunctionScaling,  
Эпилог = Epilog, Исключения = Exclusions,  
СтильИсключений = ExclusionsStyle, Заливка = Filling,  
СтильЗаливки = FillingStyle, Рамка = Frame, Нет = False,  
ПодписиРамки = FrameLabel, СтильРамки = FrameStyle,  
ЗасечкиРамки = FrameTicks, СтильЗасечекРамки = FrameTicksStyle,  
ЛинииСетки = GridLines, СтильЛинийСетки = GridLinesStyle,  
РазмерИзображения = ImageSize, СтильПодписи = LabelStyle,  
Сетка = Mesh, ФункцииСетки = MeshFunctions,  
ЗатенениеСетки = MeshShading, СтильСетки = MeshStyle,  
ПодписьГрафика = PlotLabel, ЛегендыГрафика = PlotLegends,  
КоличествоТочек = PlotPoints, ДиапазонОтображения = PlotRange,  
Полностью = Full, СтильГрафика = PlotStyle, Пролог = Prolog,  
ФункцияОбласти = RegionFunction, ВращатьПодпись = RotateLabel,  
Засечки = Ticks, СтильЗасечек = TicksStyle,  
РабочаяОтносительнаяТочность = WorkingPrecision,  
МашиннаяТочность = MachinePrecision};
```

Опції функції Plot[]

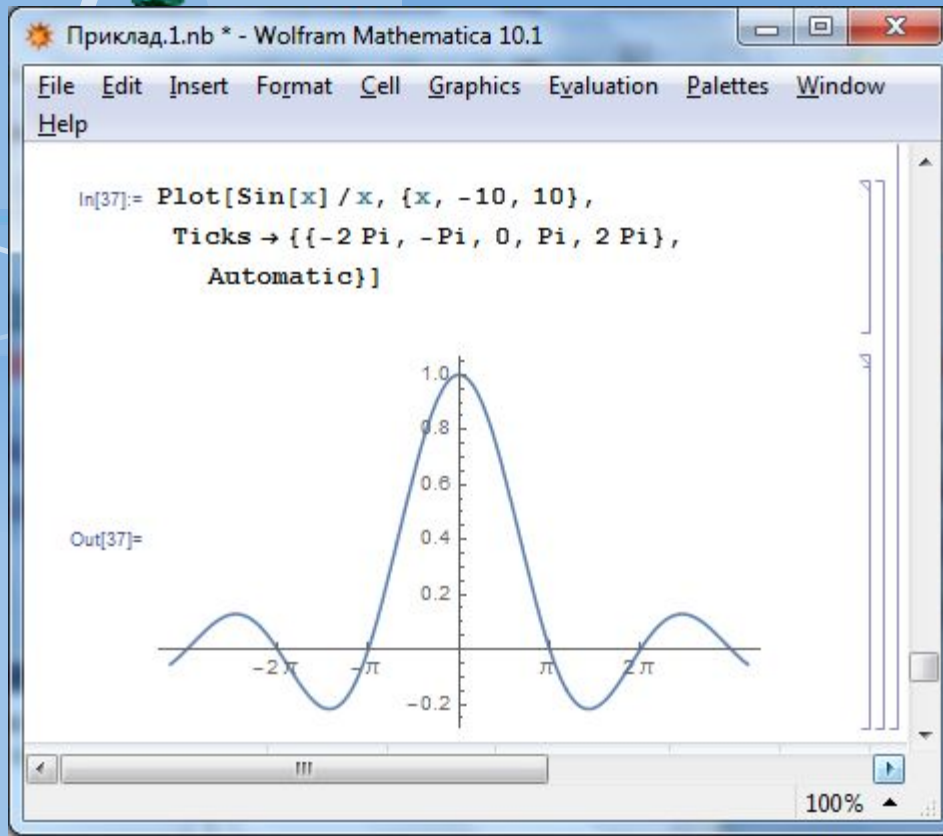
◆ Приклад◆ Приклад_1◆ Приклад 1:

`Plot[Sin[x]/x,{x,-10,10},Ticks->{{-2 Pi,-Pi,0,Pi,2 Pi},Automatic}]`

мітки шкали
("рисочки")

координати
"рисочок"

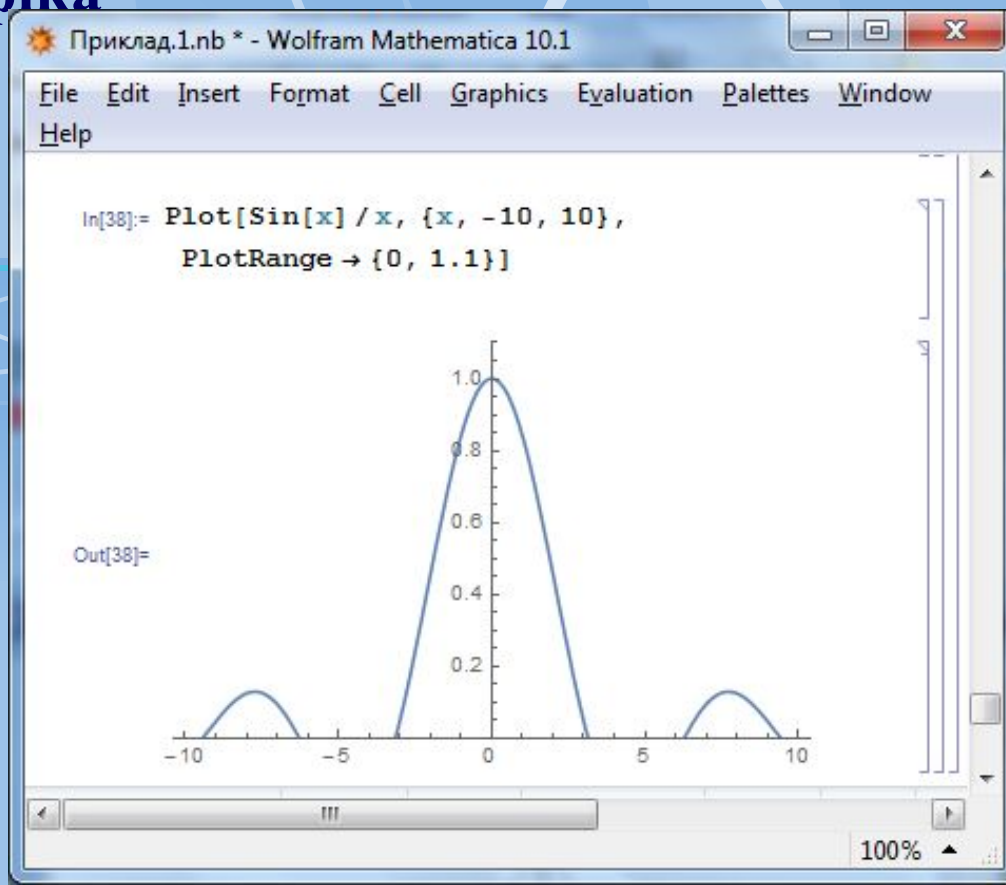
режим відображення
"рисочок" для вісі Y



◆ Приклад◆ Приклад 2◆ Приклад 2:
 $\text{Plot}[\text{Sin}[x]/x, \{x, -10, 10\}, \text{PlotRange} \rightarrow \{0, 1.1\}]$



Діапазон (по вісі Y) для відображення графіка



◆ Приклад ◆ Приклад 3 ◆ Приклад 3:

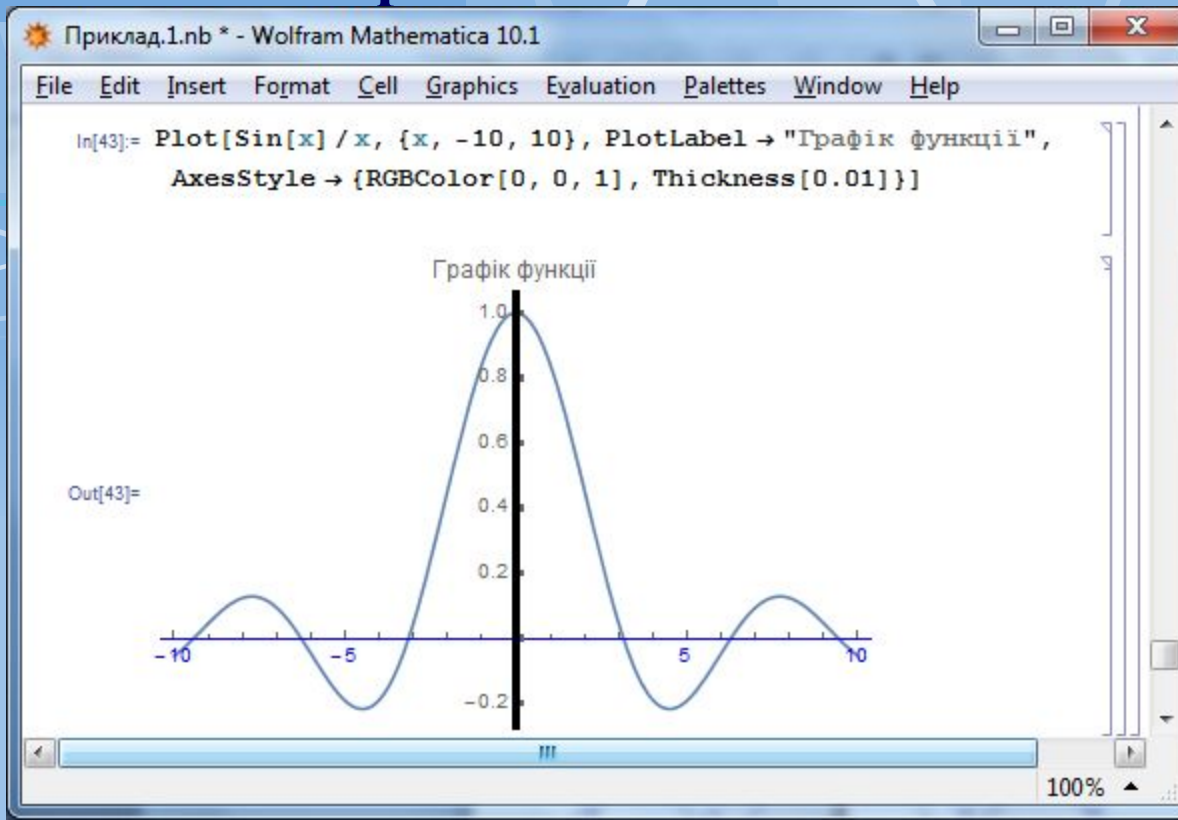
**Plot[Sin[x]/x, {x, -10, 10}, PlotLabel -> "Графік функції",
AxesStyle -> {RGBColor[0, 0, 1], Thickness[0.01]}**

Стиль координатних осей:

вісь X - колір

Заголовок для графіка

ТОВЩИНА

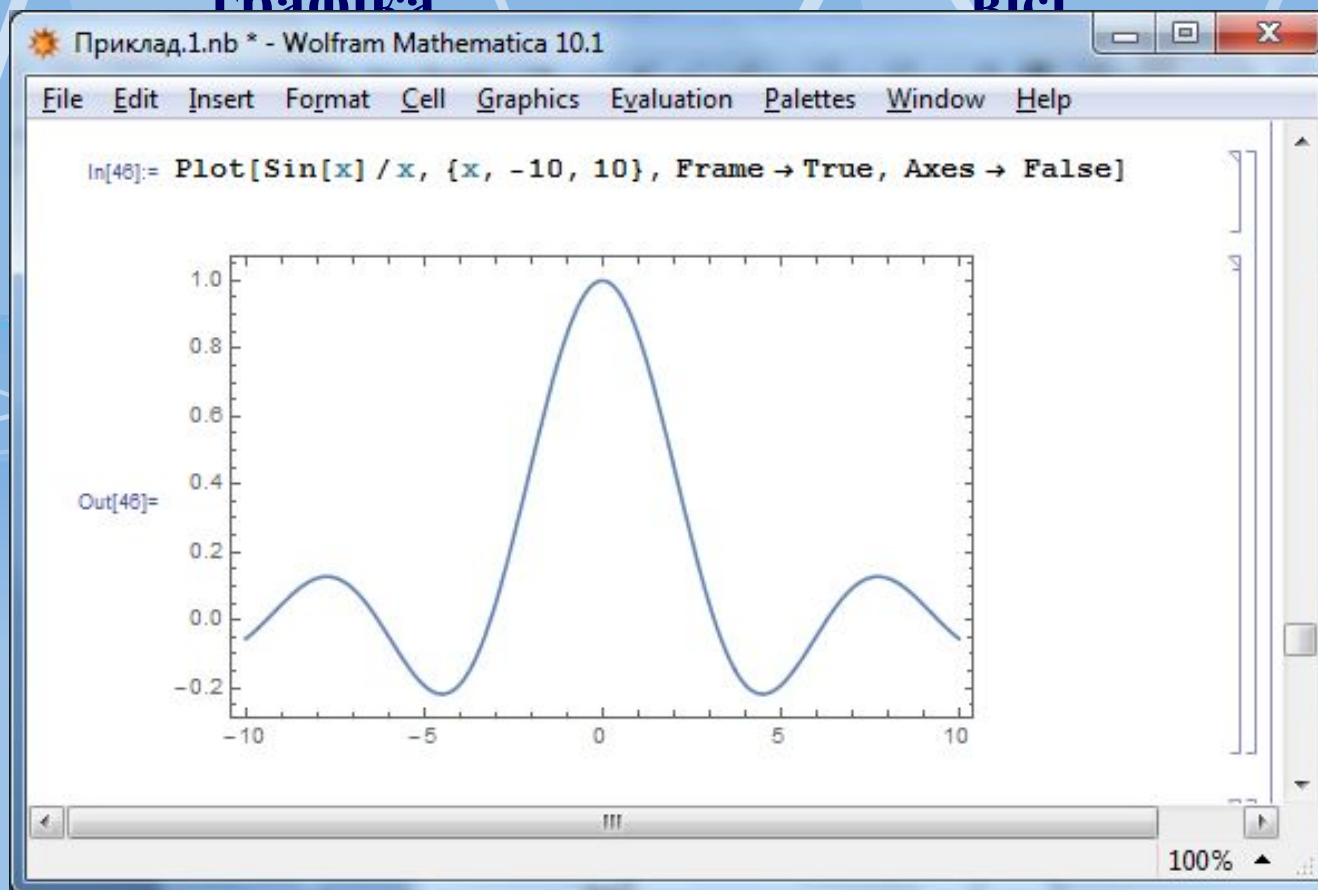


◆ Приклад ◆ Приклад_4 ◆ Приклад 4:

Plot[Sin[x]/x, {x, -10, 10}, Frame -> True, Axes -> False]

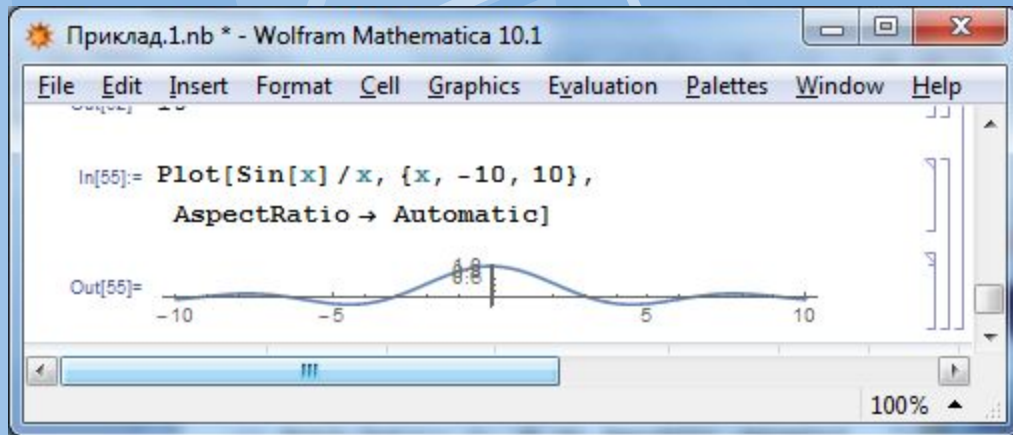
Рамка навколо
графіка

Координатні
вісі

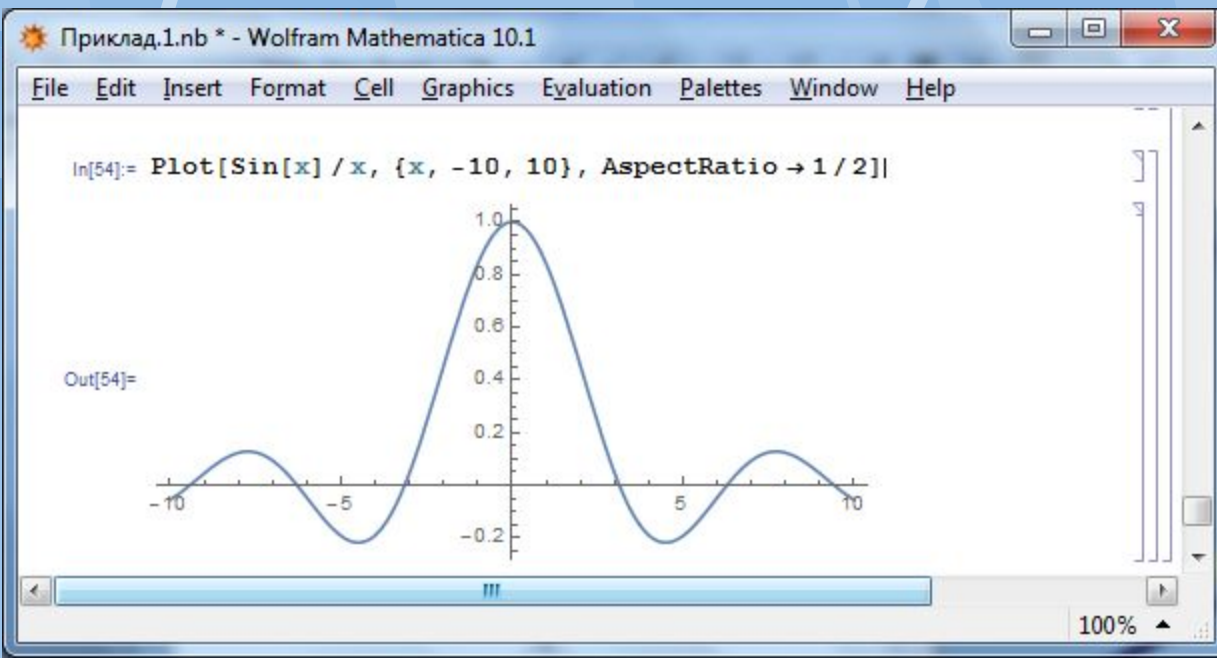


◆ Приклад 5:

`Plot[Sin[x]/x, {x, -10, 10}, AspectRatio -> Automatic]`



Масштаб (співвідношення
висота/ширина графіка)



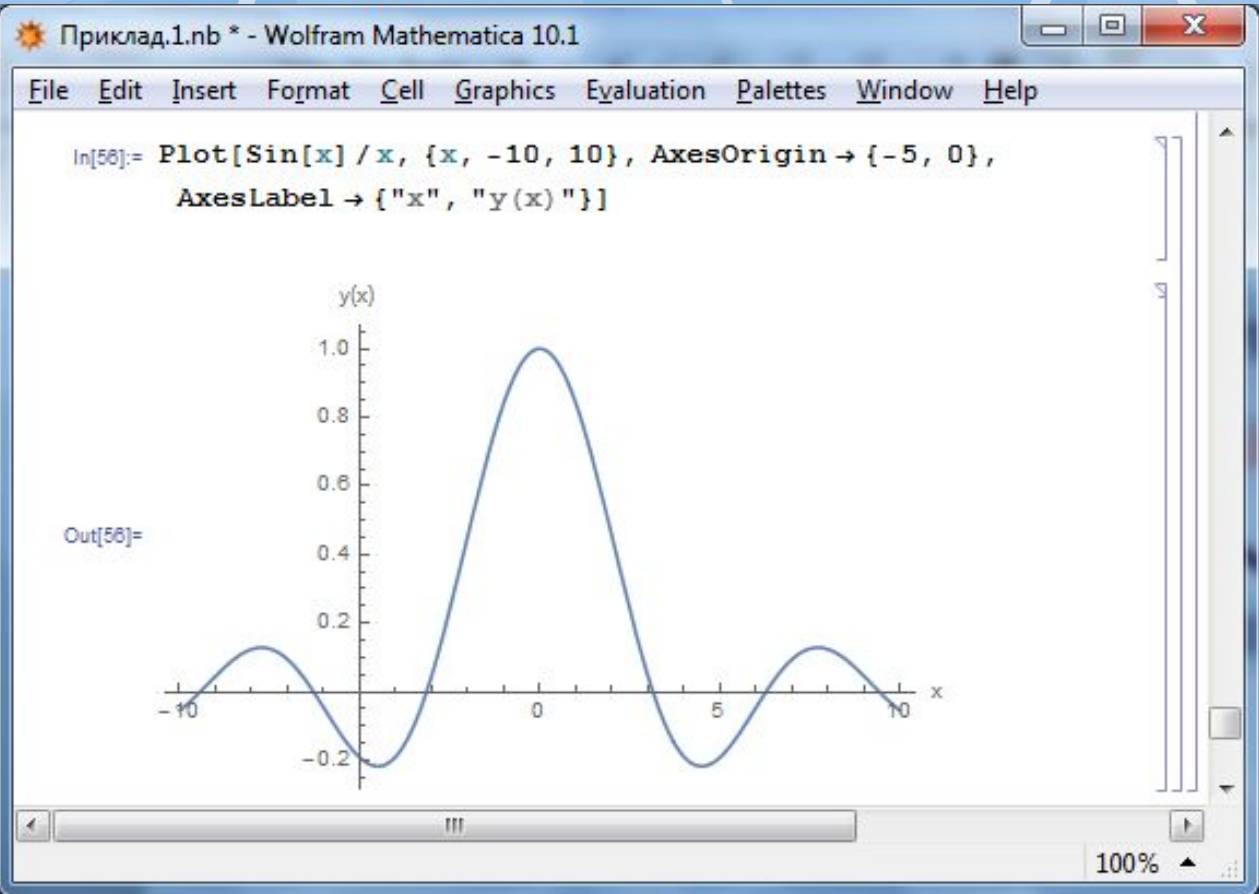
AspectRatio -> 1/2

◆ Приклад ◆ Приклад_6 ◆ Приклад 6

```
Plot[Sin[x]/x, {x, -10, 10},  
AxesOrigin -> {-5, 0}, AxesLabel -> {"x", "y(x)"}]
```

Точка перетину осей

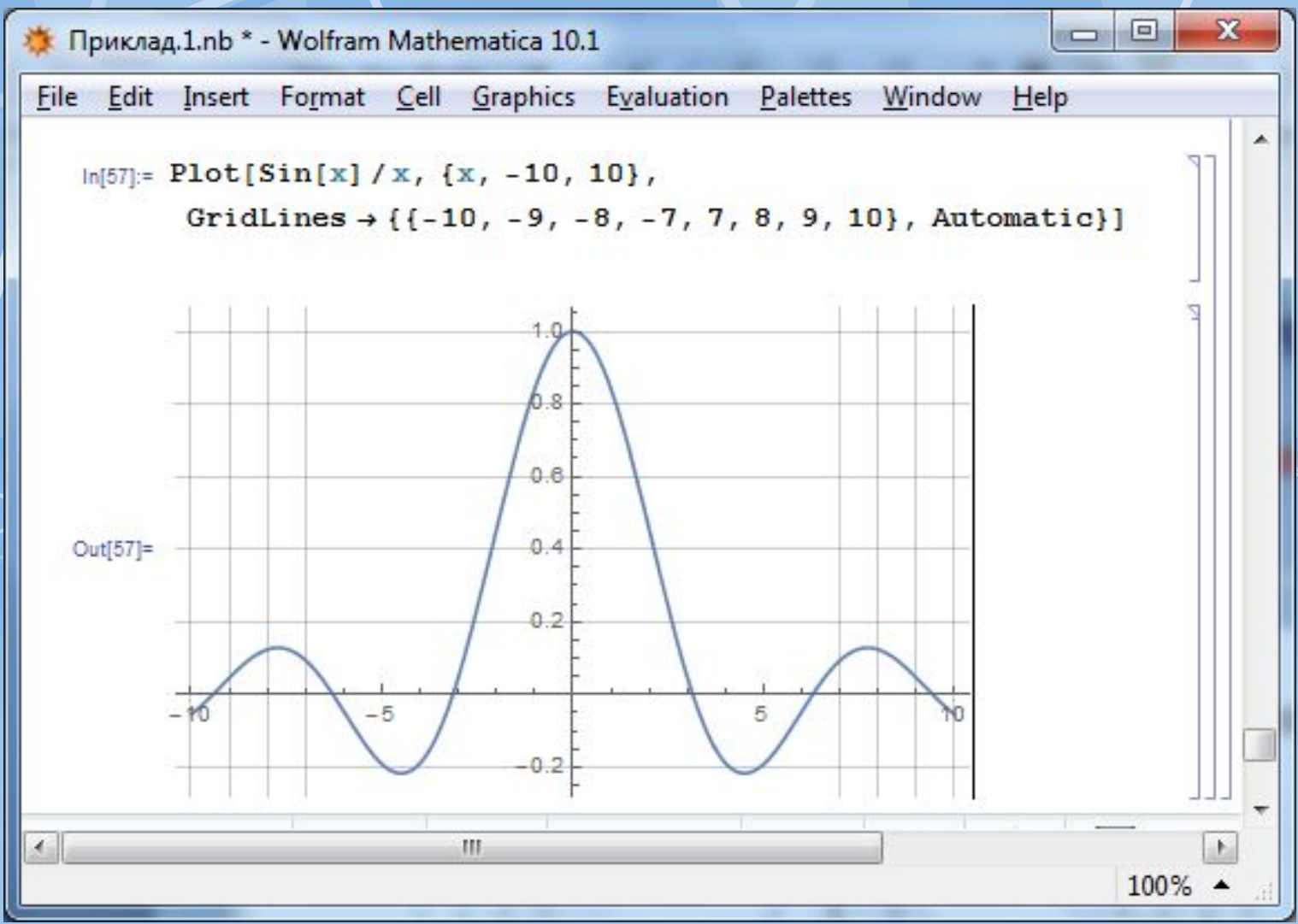
Підписи для осей



Лінії координатної сітки

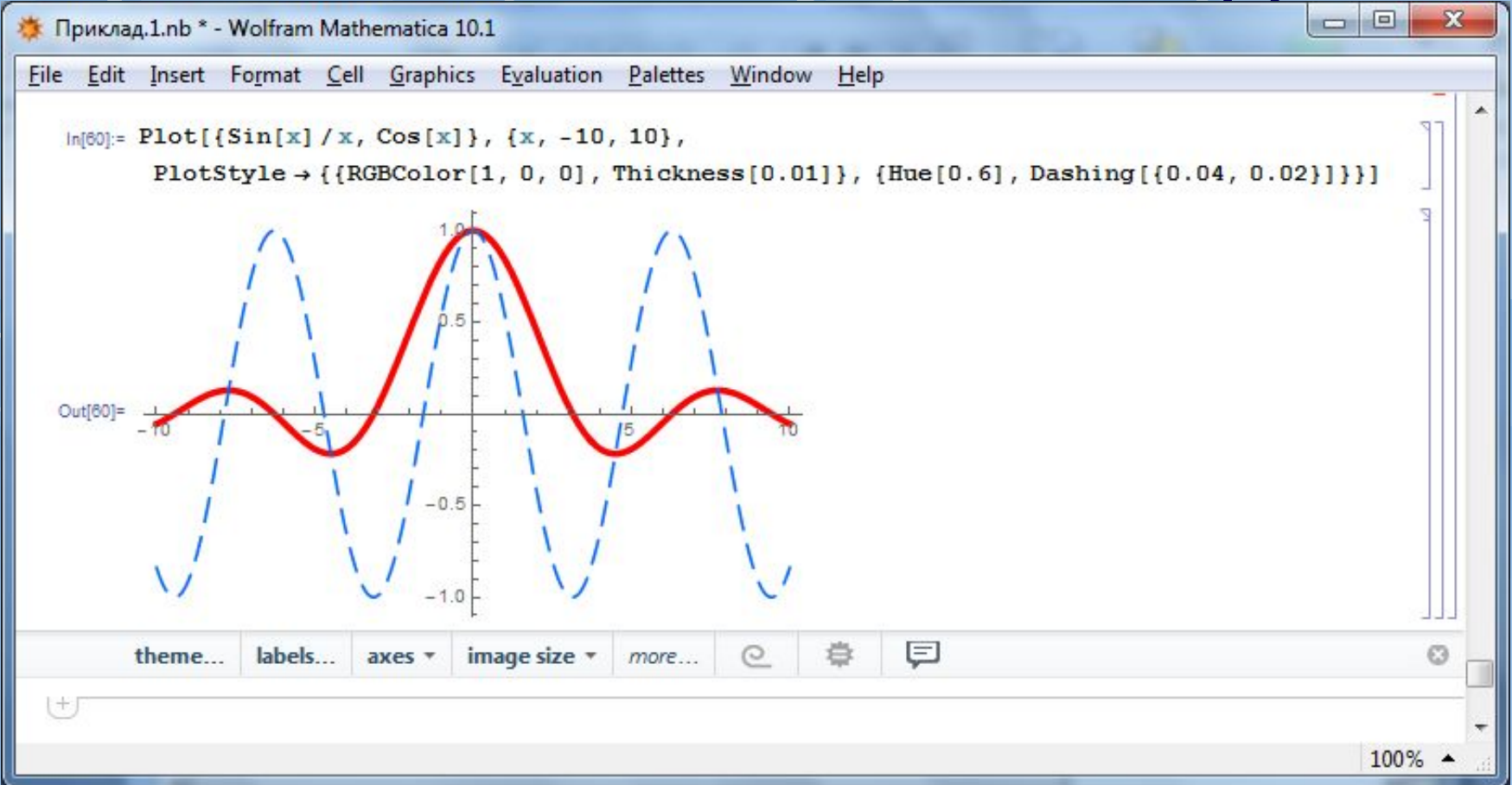
◆ Приклад ◆ Приклад_7 ◆ Приклад_7

```
Plot[Sin[x]/x, {x, -10, 10},  
GridLines -> {{-10, -9, -8, -7, 7, 8, 9, 10}, Automatic}]
```



◆ Приклад ◆ Приклад_8 ◆ Приклад 8, **Стиль ліній**
Plot[{Sin[x]/x, Cos[x]},{x,-10,10},
PlotStyle -> {{**RGBColor**[1, 0, 0], **Thickness**[0.01]},
{**Hue**[0.6], **Dashing**[{0.04, 0.02}]}}

Перша лінія
Друга

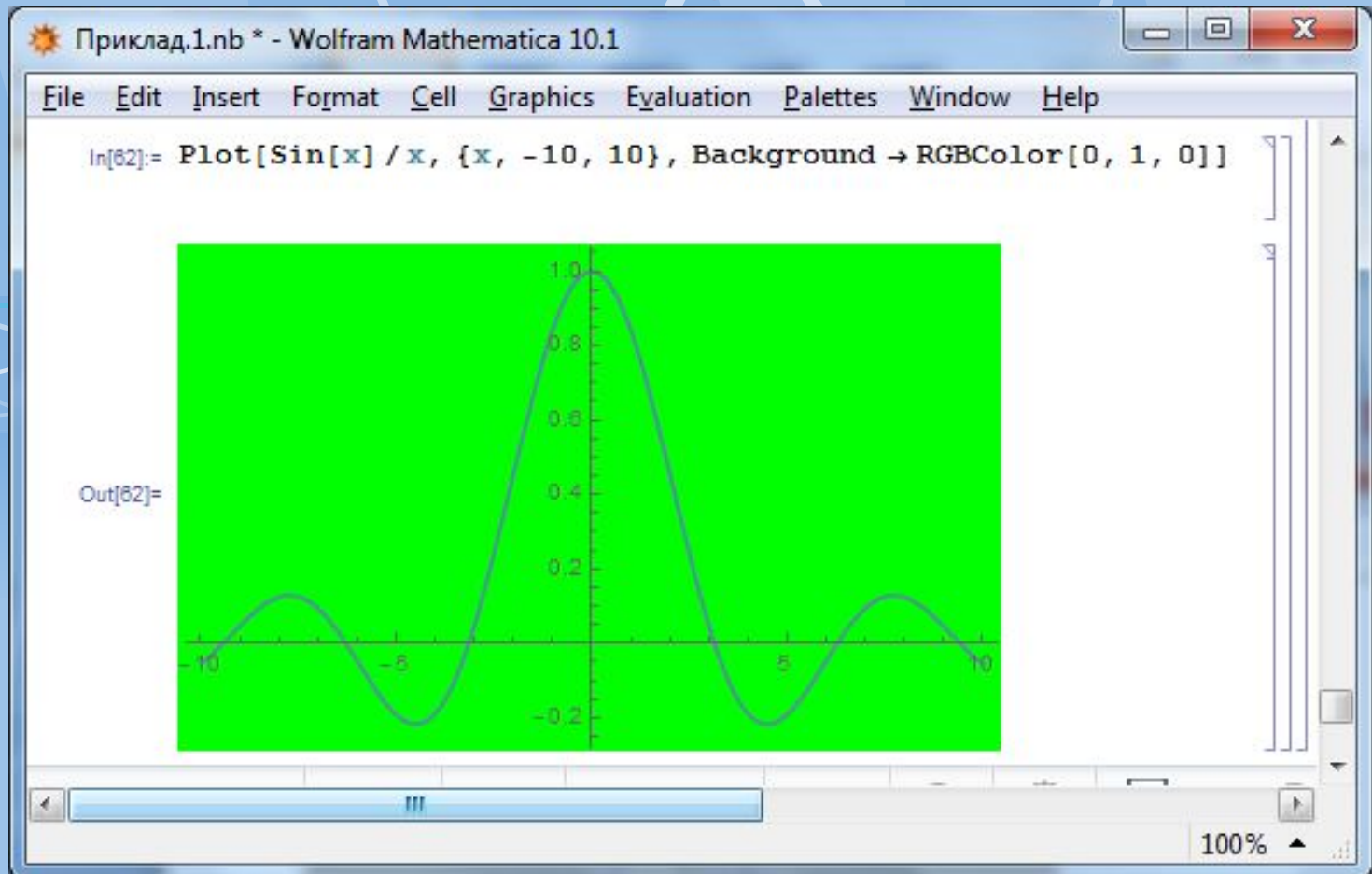


◆ Приклади:

```
Plot[Sin[x]/x,{x,-10,10}, Background -> GrayLevel[0.7]]
```



```
Plot[Sin[x]/x,{x,-10,10}, Background -> RGBColor[0, 1, 0]]
```



Також часто виникає необхідність побудови графіка по точках. Це забезпечує вбудована в ядро графічна функція **ListPlot**:

- **ListPlot** [$\{y_1, y_2, \dots\}$] – виводить графік списку величин.

Координати x приймають значення 1, 2, ...;

- **ListPlot** [$\{\{x_1, y_1\}, \{x_2, y_2\}, \dots\}$] – виводить графік списку величин з зазначеними x і y координатами.

У найпростішому випадку (рис. 7) ця функція сама задає значення координати $x = 0, 1, 2, 3, \dots i$ буде на графіку точки з координатами (x, y) , вибираючи y послідовно зі списку координат.

Функція **ListPlot**, особливо в її другій формі (із заданими координатами x і y), зручна для виведення на графік експериментальних точок

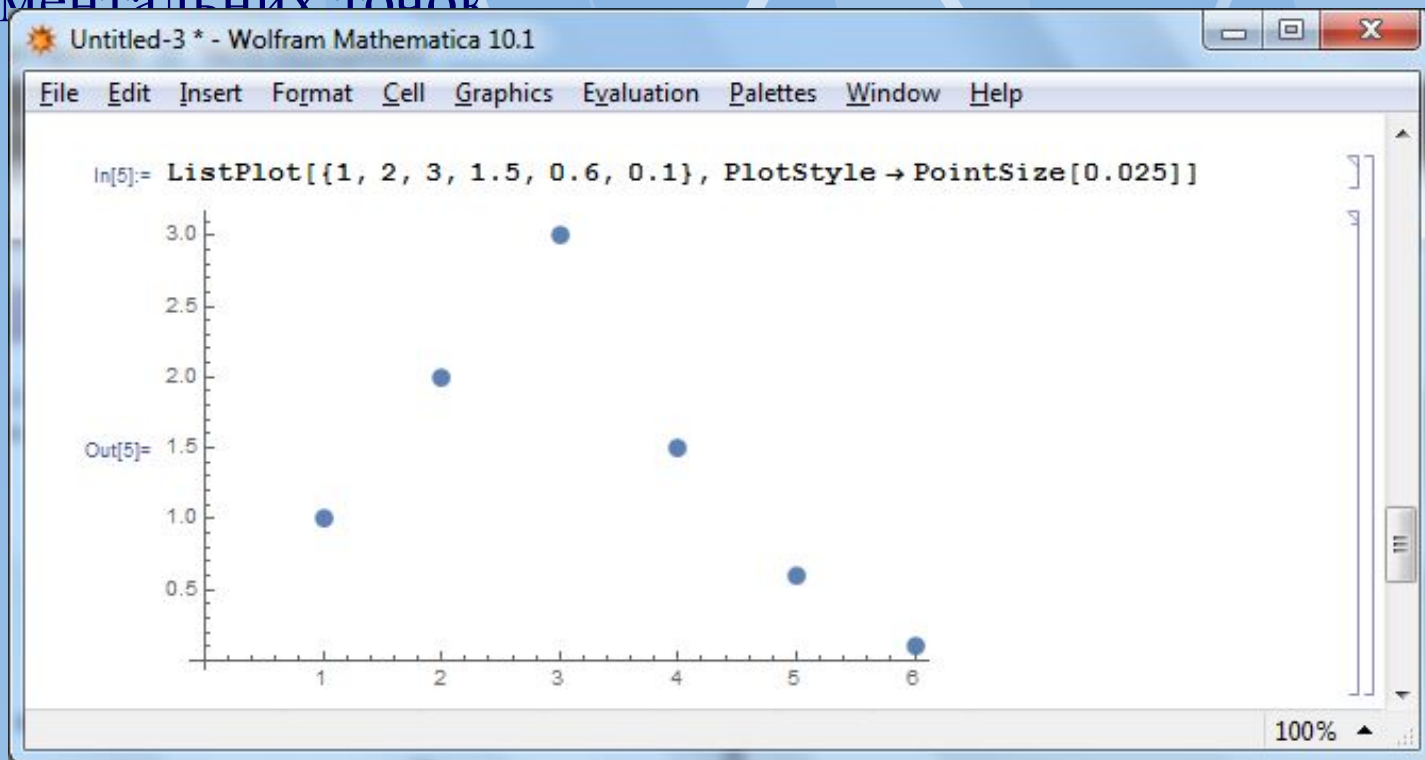


Рис.7. Приклад Побудови графіка по точках

Для побудови параметрично заданих функцій використовуються наступні графічні засоби:

- **ParametricPlot** [$\{fx, fy\}$, $\{t, tmin, tmax\}$] – будує параметричний графік з координатами fx і fy (відповідними x і y), одержуваними як функції від t ;
- **ParametricPlot** [$\{\{fx, fy\}, \{gx, gy\}, \dots\}$, $\{t, tmin, tmax\}$] – будує графіки декількох параметричних кривих.

Функції fx , fy можуть бути як безпосередньо вписані в список параметрів, так і визначені як функції користувача.

На рис. 8 показано побудову параметрично заданої фігури Ліссажу.

Вона задається функціями синуса і косинуса з постійним параметром R і аргументами, кратними t .

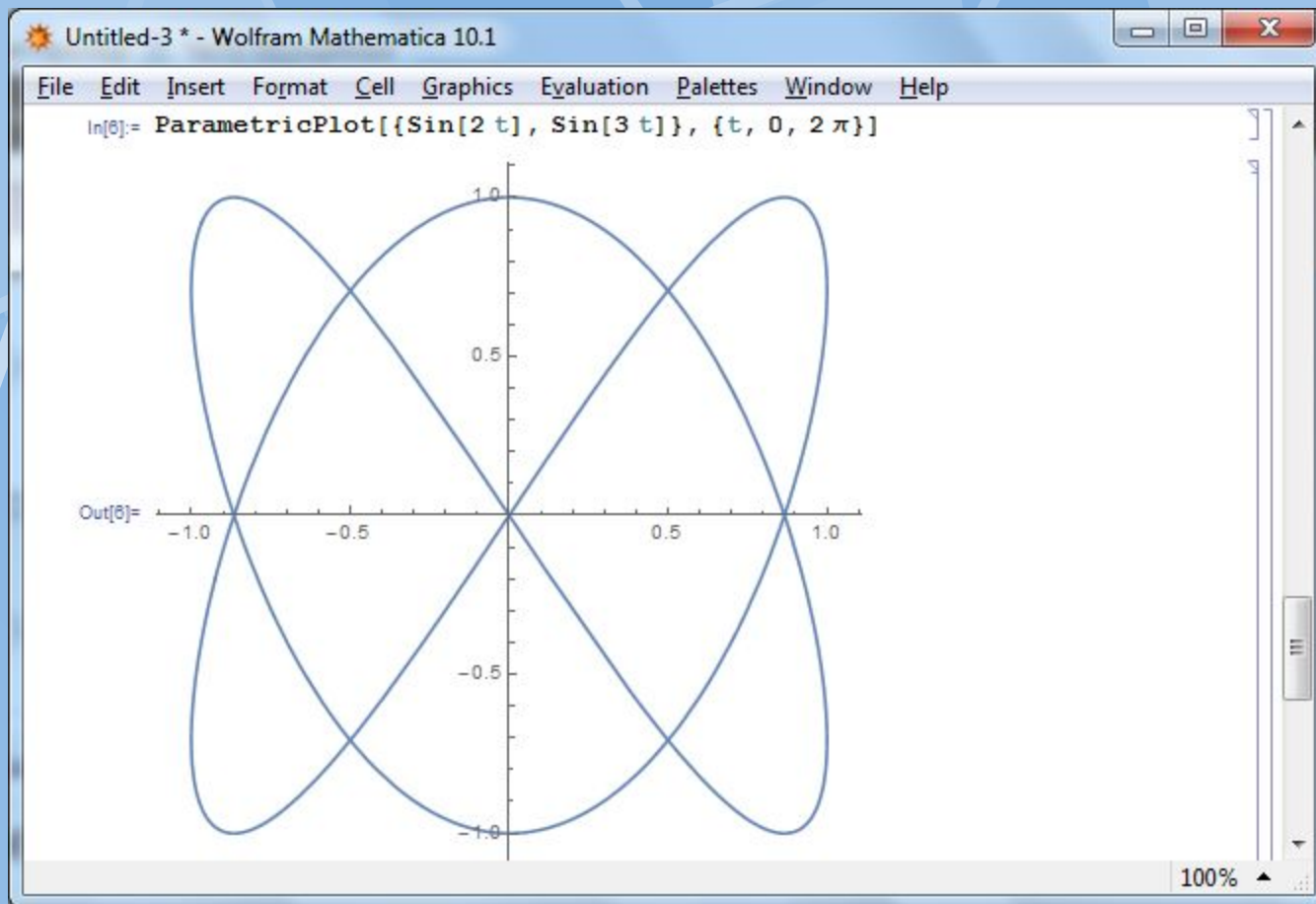


Рис.8. Побудова фігури Ліссажу

Тепер розглянемо спосіб побудови графіків в полярній системі координат (рис. 9).

Для цього використовується функція

PolarPlot:

- **PolarPlot [f, {t, tmin, tmax}]** – будує графік в полярній системі координат.
- **PolarPlot [{f1, f2, f3, ...}, {t, tmin, tmax}]** – будує графіки функцій в полярній системі координат.

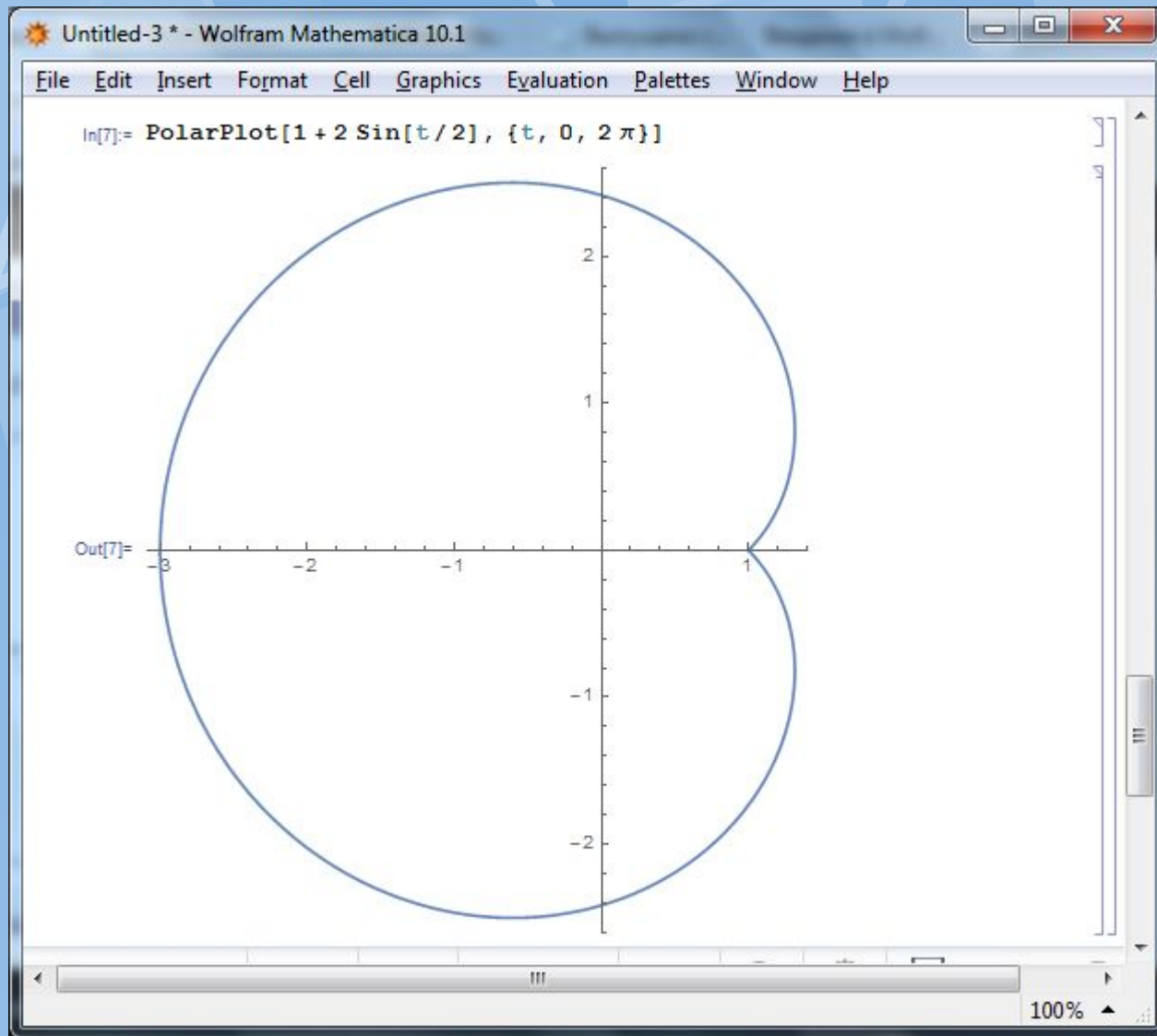


Рис.9.Приклад побудови графіка функції в полярній системі координат

Побудова графіків поверхонь

Функція двох змінних $z = f(x, y)$ утворює в просторі деяку тривимірну поверхню або фігуру. Для їх побудови доводиться використовувати координатну систему з трьома осями координат: x , y і z . Оскільки екран дисплея плоский, то насправді об'ємність фігур лише імітується.

Для побудови графіків тривимірних поверхонь в системі Mathematica використовується основна графічна функція **Plot3D**:

- **Plot3D** [f , { x , x_{\min} , x_{\max} }, { y , y_{\min} , y_{\max} }] – будує тривимірний графік функції $f(x, y)$;
- **Plot3D** [{ f , s }, { x , x_{\min} , x_{\max} }, { y , y_{\min} , y_{\max} }] – будує тривимірний графік, в якому висоту поверхні визначає параметр f , а затінення - параметр s .

На рис.11 показаний приклад побудови поверхні, що описується функцією двох змінних $\cos(xy)$ при x і y , що міняються від -3 до 3.

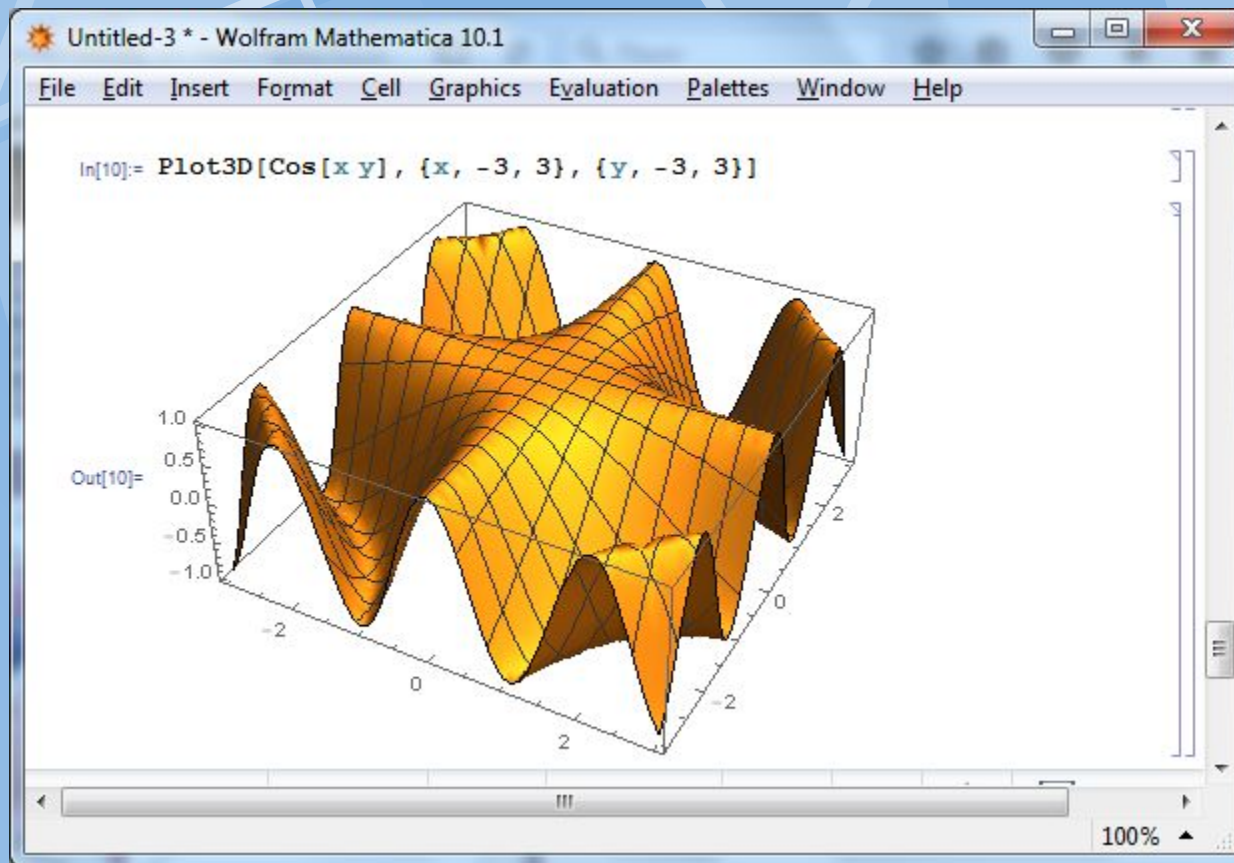


Рис.11. Приклад побудови поверхні

Поверхні також можна будувати по точкам та в параметричній формі використовуючи при цьому відповідні функції **ListPointPlot3D** і **ParametricPlot3D**.

3. Елементи програмування

3.1. Основні арифметичні оператори

2.3. Логічні оператори

3. 3. Оператори порівняння

3.4. Умовні оператори

3.5. Оператори циклу

3.6. Створення модулів

3.7. Приклад

Арифметичні оператори (1)

= Присвоєння

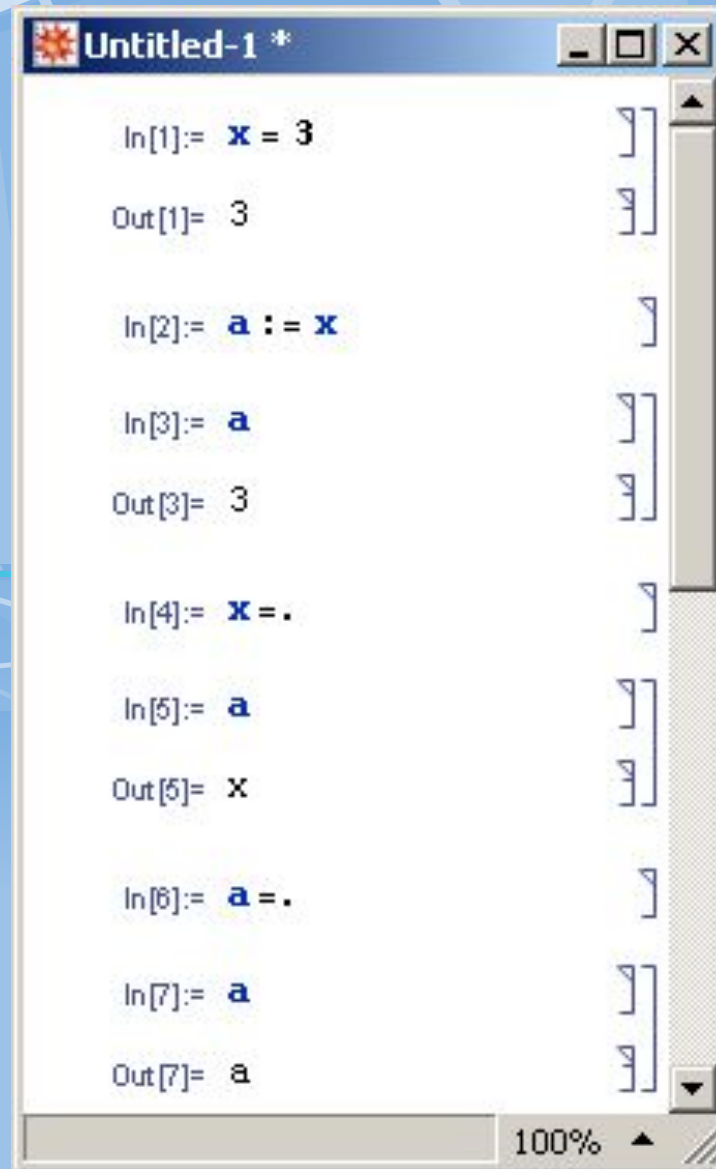
:= Функціональне присвоєння

```
In[8]:= x = 3  
Out[8]= 3  
  
In[9]:= b = x  
Out[9]= 3  
  
In[10]:= b  
Out[10]= 3  
  
In[11]:= x = 5  
Out[11]= 5  
  
In[12]:= b  
Out[12]= 3
```

```
In[13]:= x = 3  
Out[13]= 3  
  
In[14]:= b := x  
  
In[15]:= b  
Out[15]= 3  
  
In[16]:= x = 5  
Out[16]= 5  
  
In[17]:= b  
Out[17]= 5
```

Арифметичні оператори (2)

=. Оператор відмови від значення



```
Untitled-1 *  
  
In[1]:= x = 3  
Out[1]= 3  
  
In[2]:= a := x  
  
In[3]:= a  
Out[3]= 3  
  
In[4]:= x = .  
  
In[5]:= a  
Out[5]= x  
  
In[6]:= a = .  
  
In[7]:= a  
Out[7]= a
```

100%

Арифметичні оператори (3)

++ Оператор інкременту

-- Оператор декременту

```
Untitled-3 *
In[18]:= i = 10
Out[18]= 10

In[19]:= i++
Out[19]= 10

In[20]:= i
Out[20]= 11

In[21]:= ++i
Out[21]= 12

In[22]:= i
Out[22]= 12

100%
```

```
Untitled-3 *
In[23]:= i = 10
Out[23]= 10

In[24]:= i--
Out[24]= 10

In[25]:= i
Out[25]= 9

In[26]:= --i
Out[26]= 8

In[27]:= i
Out[27]= 8

100%
```


Арифметичні оператори (4)

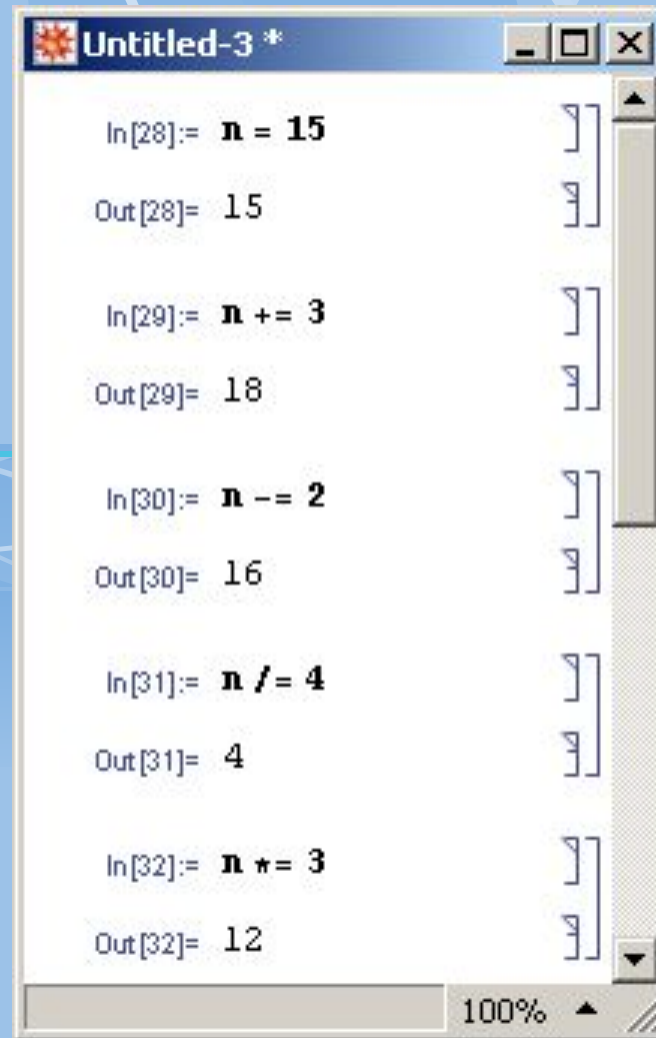
Скорочені форми операторів:

+= додавання

-= віднімання

***=** множення

/= ділення



```
In[28]:= n = 15
Out[28]= 15

In[29]:= n += 3
Out[29]= 18

In[30]:= n -= 2
Out[30]= 16

In[31]:= n /= 4
Out[31]= 4

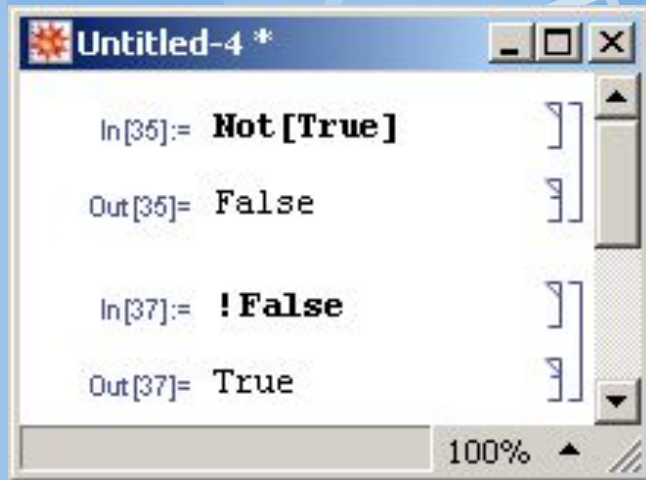
In[32]:= n *= 3
Out[32]= 12
```

The screenshot shows a Mathematica notebook window with the title 'Untitled-3 *'. The notebook contains a series of input-output pairs for arithmetic operations on a variable 'n'. The operations are: n = 15, n += 3, n -= 2, n /= 4, and n *= 3. The corresponding outputs are 15, 18, 16, 4, and 12. The notebook interface includes a scroll bar on the right and a zoom level of 100% at the bottom.

Логічні оператори (1)

Логічне заперечення:

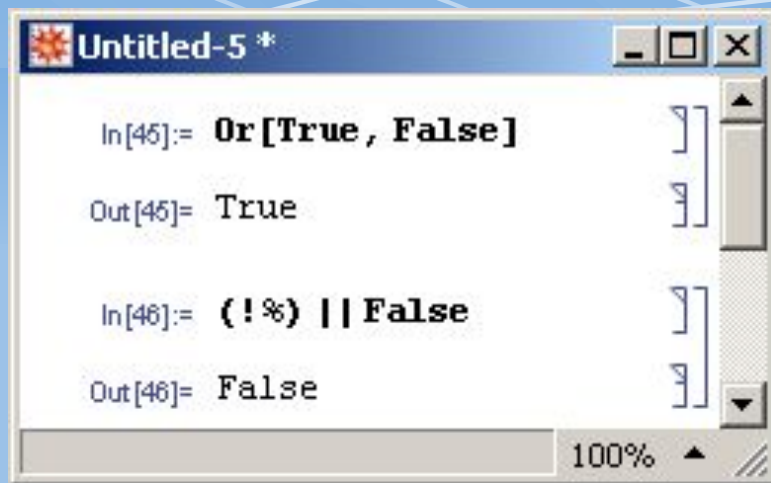
Not[] або **!**



```
Untitled-4 *  
In[35]:= Not[True]  
Out[35]= False  
  
In[37]:= !False  
Out[37]= True
```

Логічне OR:

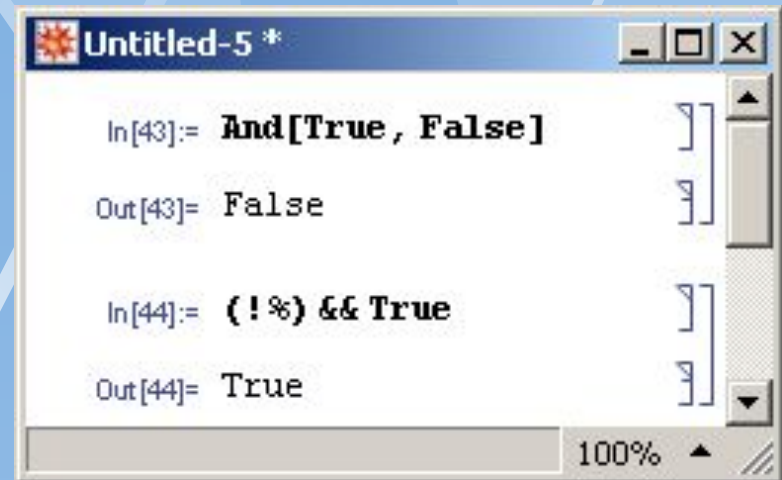
Or[] або **||**



```
Untitled-5 *  
In[45]:= Or[True, False]  
Out[45]= True  
  
In[46]:= (!%) || False  
Out[46]= False
```

Логічне AND:

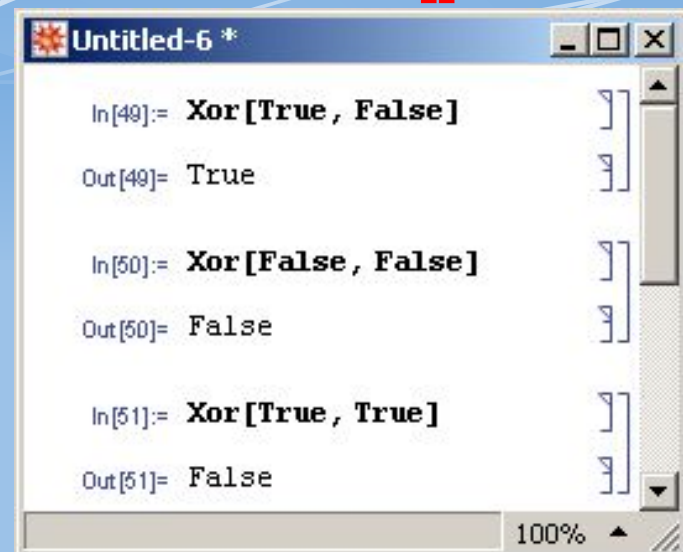
And[] або **&&**



```
Untitled-5 *  
In[43]:= And[True, False]  
Out[43]= False  
  
In[44]:= (!%) && True  
Out[44]= True
```

Логічне XOR:

Xor[]

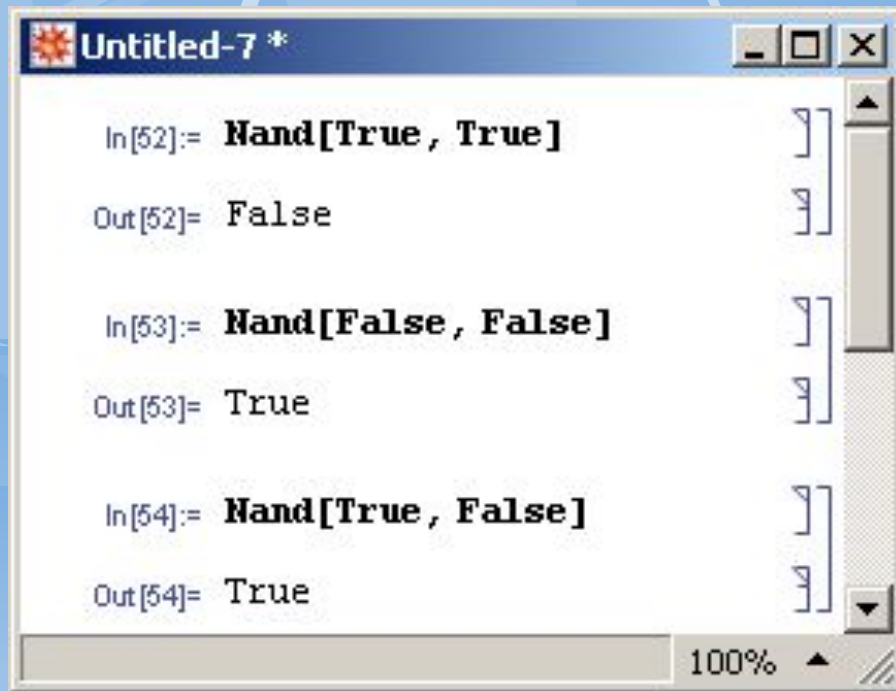


```
Untitled-6 *  
In[49]:= Xor[True, False]  
Out[49]= True  
  
In[50]:= Xor[False, False]  
Out[50]= False  
  
In[51]:= Xor[True, True]  
Out[51]= False
```

Логічні оператори (2)

Логічне заперечуюче AND:

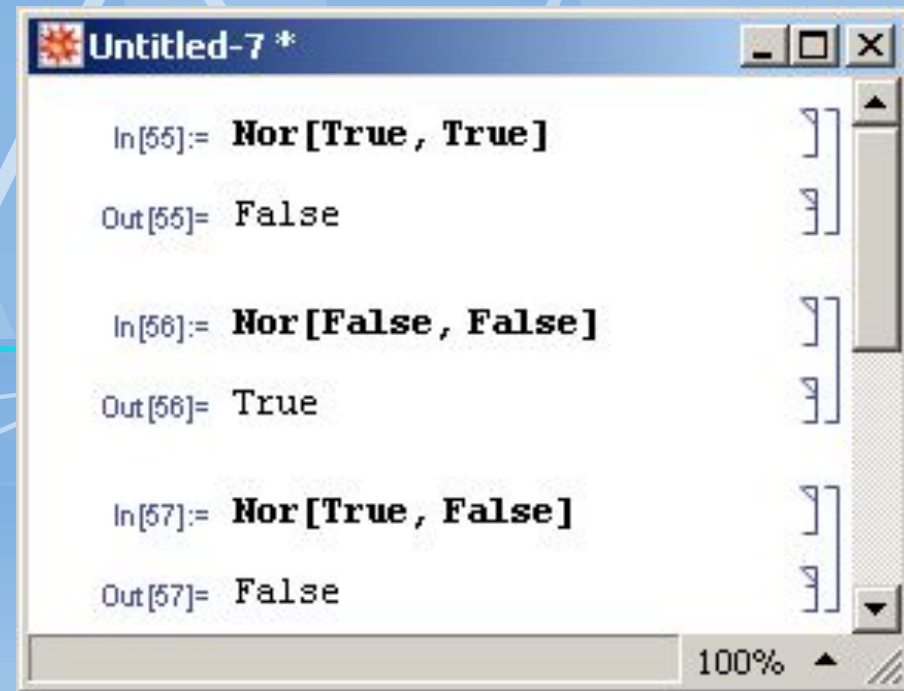
Nand[]



```
Untitled-7 *  
  
In[52]:= Nand[True, True]  
Out[52]= False  
  
In[53]:= Nand[False, False]  
Out[53]= True  
  
In[54]:= Nand[True, False]  
Out[54]= True  
  
100%
```

Логічне заперечуюче OR:

Nor[]



```
Untitled-7 *  
  
In[55]:= Nor[True, True]  
Out[55]= False  
  
In[56]:= Nor[False, False]  
Out[56]= True  
  
In[57]:= Nor[True, False]  
Out[57]= False  
  
100%
```

Оператори порівняння

Оператор

Опис

==

Дорівнює

!=

Не дорівнює

===

Тотожно дорівнює

!==

Тотожно не дорівнює

<

Менше

<=

Менше або дорівнює

>

Більше

>=

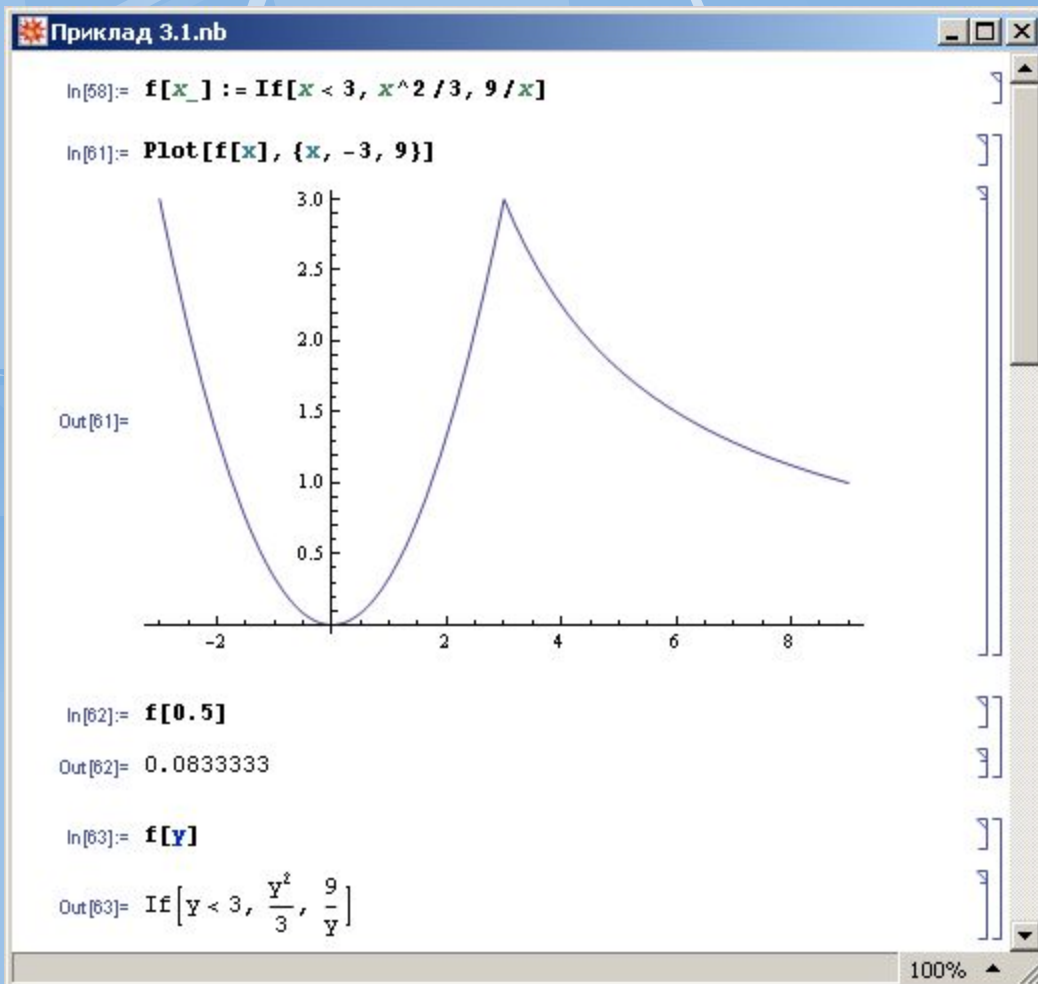
Більше або дорівнює

Умовний оператор If[]

Синтаксис: If [умова, оператор1, оператор2]

або: If[умова, оператор1, оператор2, оператор3]

Приклад 1:



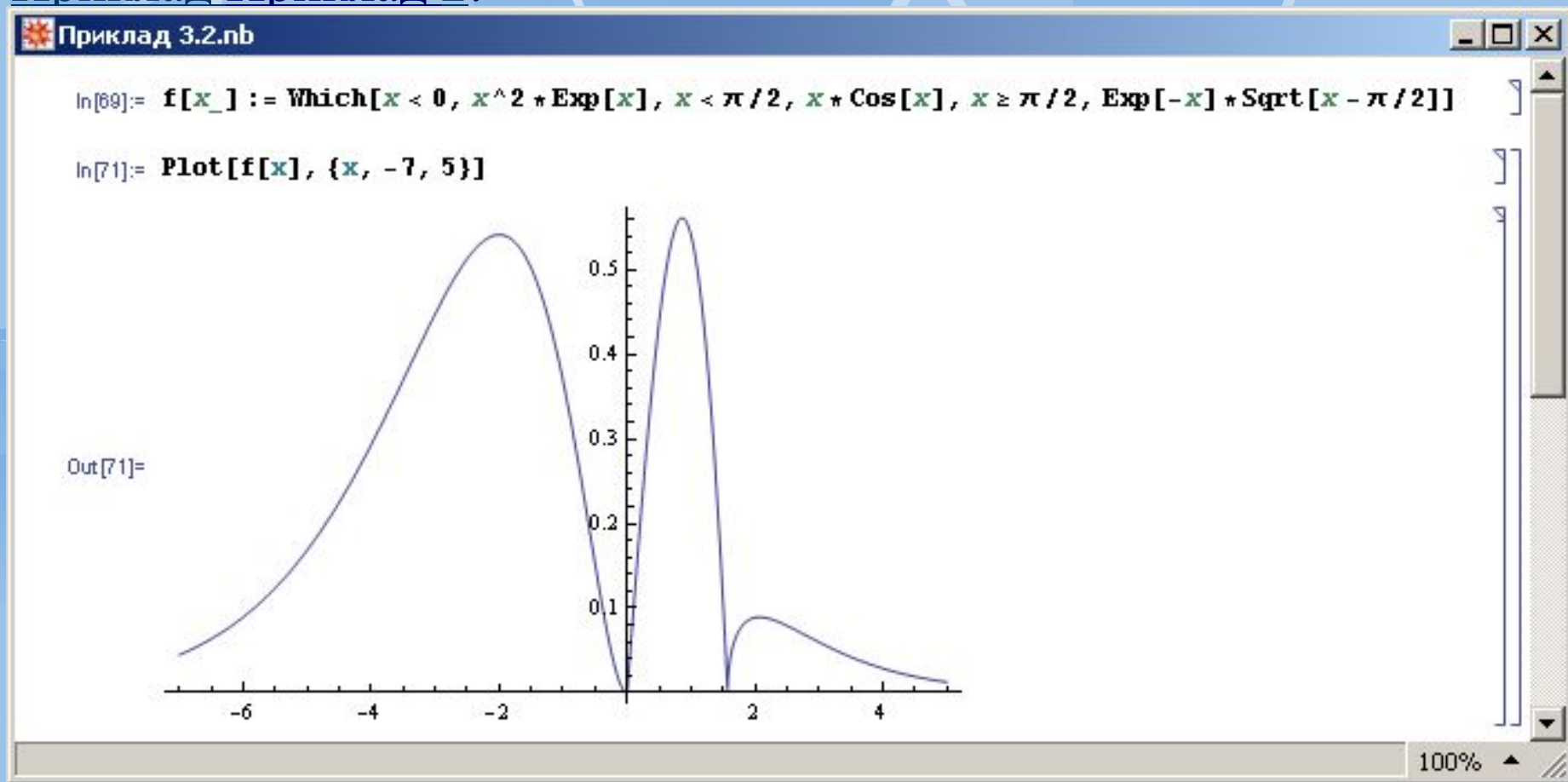
Якщо не може бути
розрахована УМОВА

Умовний оператор Which[]

Синтаксис:

Which[умова1,оператор1, умова2,оператор2,...]

Приклад Приклад 2:

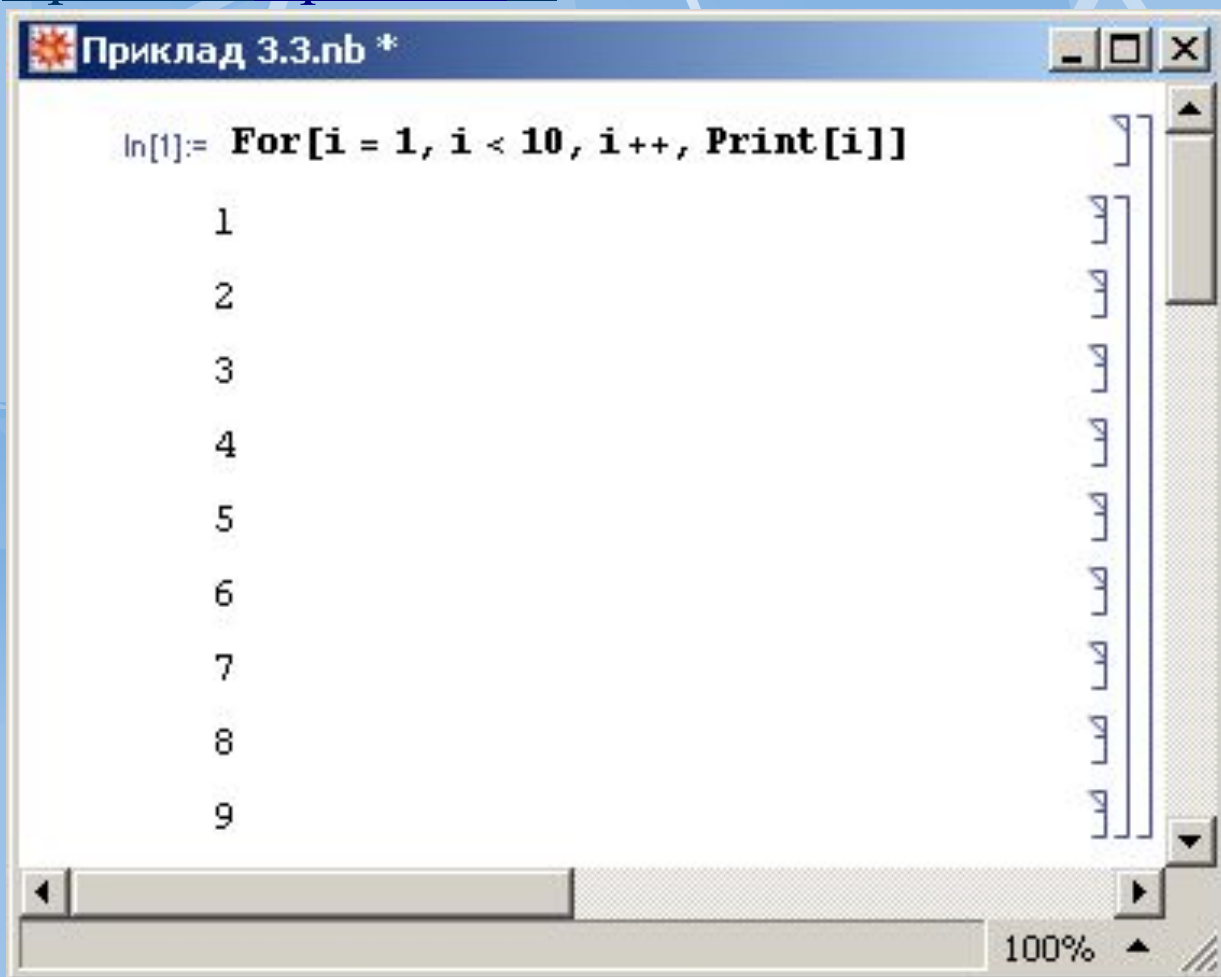


Оператор циклу For[]

Синтаксис:

For[ініціалізація, умова, зміна_параметра_циклу, оператори]

Приклад Приклад 3:



```
In[1]:= For[i = 1, i < 10, i++, Print[i]]
```

1
2
3
4
5
6
7
8
9

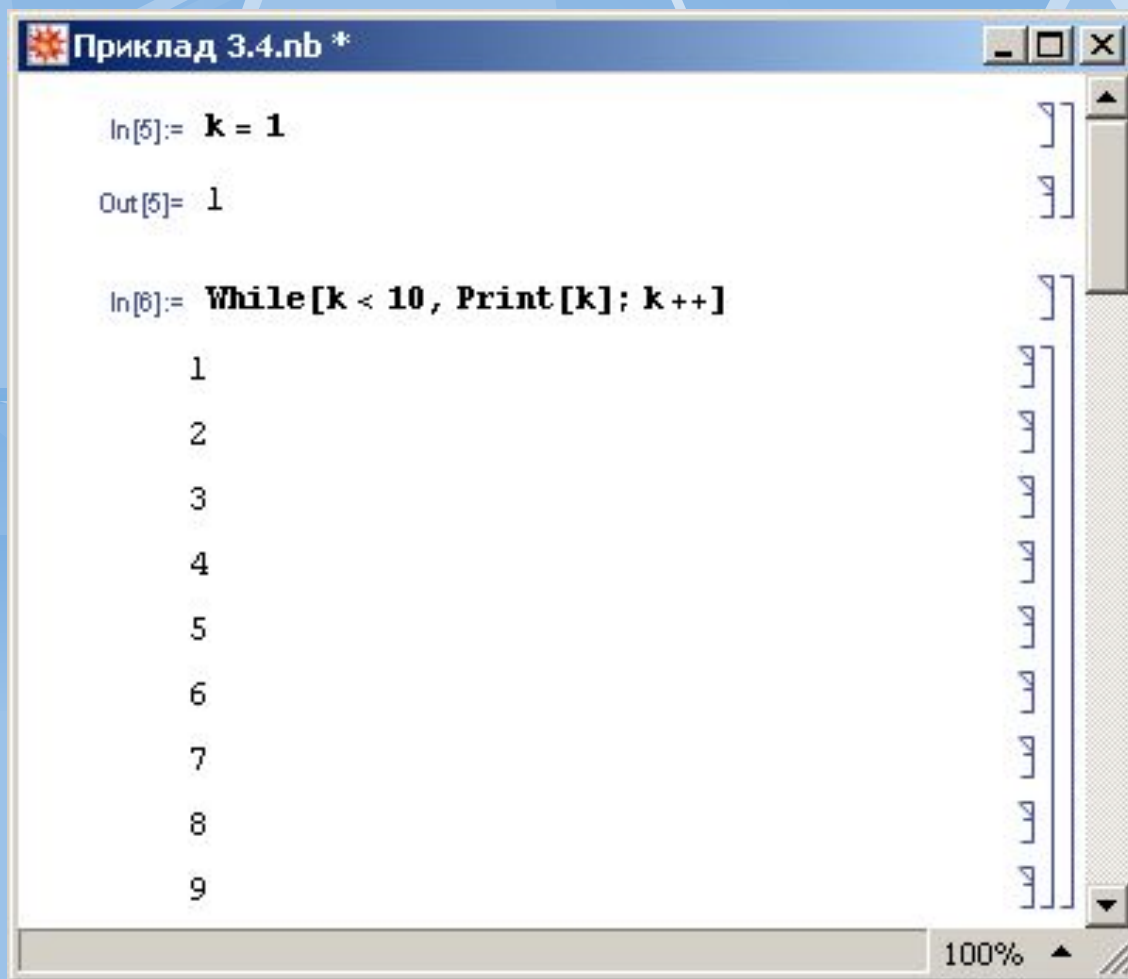
100%

Оператор цикла While[]

Синтаксис:

While[умова,оператори]

Приклад Приклад 4:



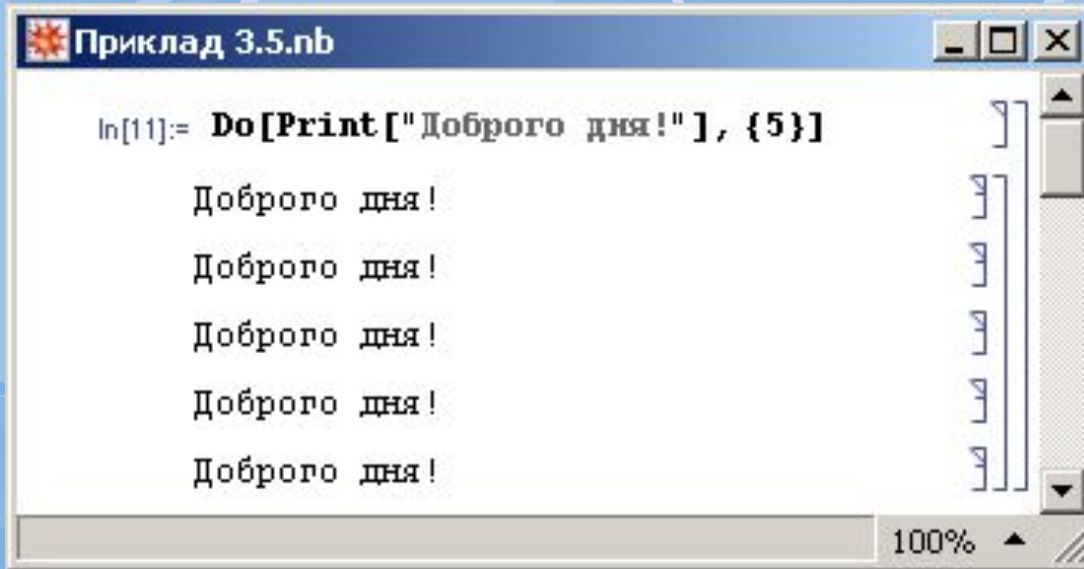
```
Приклад 3.4.nb *  
  
In[5]:= k = 1  
Out[5]= 1  
  
In[6]:= While[k < 10, Print[k]; k++]  
1  
2  
3  
4  
5  
6  
7  
8  
9  
  
100%
```


Оператор циклу Do[]

Синтаксис:

Do[оператори, список_кількість_циклів]

Приклад Приклад 5:



```
Приклад 3.5.nb

In[11]:= Do[Print["Доброго дня!"], {5}]

Доброго дня!
Доброго дня!
Доброго дня!
Доброго дня!
Доброго дня!
```

1. Список з кількістю циклів.
2. Список з індексною змінною і границею натурального ряду
3. Список з індексною змінною, границями діапазону зміни та кроком дискретності

Використання модулів

Функція `Module[]`

Аргументи:

1. Список з локальними змінними модуля
2. Блок операторів модуля
3. Для повернення значення використовують функцію `Return[]`



Приклад 6:

```
Приклад 3.6.nb

Використання модуля:

In[1]:= SQsum[n_] := Module[{s, i}, s = 0; For[i = 0, ++i ≤ n, s += i^2]; Return[s]]

Перевірка коректності роботи процедури:

In[2]:= SQsum[5]

Out[2]= 55

100%
```

Приклад: числа Фібоначчі

Числа: 1, 1, 2, 3, 5, 8, 13 і т.д.

Приклад 7:

```
Приклад 3.7.nb *  
  
Розрахунок чисел Фібоначчі:  
  
In[3]:= Fibon[n_] := (If[n == 1 || n == 2, Return[1], Return[Fibon[n - 1] + Fibon[n - 2]])  
  
In[4]:= Fibon2[n_] := (  
    a = 1;  
    b = 1;  
    For[i = 3, i ≤ n, (b = b + a; a = b - a); i ++];  
    Return[b];)  
  
In[5]:= Fibon[7]  
Out[5]= 13  
  
In[6]:= Fibon2[7]  
Out[6]= 13  
  
In[7]:= Fibonacci[7]  
Out[7]= 13  
  
100%
```

Лабораторна робота №4

Тема: Основи роботи в пакеті Mathematica (10 балів)

Завдання.

1. Записати арифметичні вирази в середовищі Mathematica і обчислити значення при заданому параметрі (x) . (1 бал)
2. Проведіть обчислення функції при заданій умові. (1 бал)
3. Обчисліть значення функції f(x) (1бал)
4. Обчисліть суми чисел (1 бал)
5. Обчисліть значення функції f(x) на відрізку [a; b] з кроком h. Побудуйте графік функції з використанням не менше 5-ти опцій функції Plot. (2 бали)
6. Відповідно до таблиці виконати: (2 бали)

-обчислення точних (використовуючи стандартні функції Mathematica) значень, що відповідають функції в діапазоні зміни аргументу від x_1 до x_2 в m рівномірних точках цього діапазону, включаючи його границі;

- обчислення за вказаними степеневими рядами наближених значень функції в тих самих точках, обмежуючись r першими членами ряду;

- обчислення похибки наближеного значення функції в кожній точці, порівнюючи наближене значення з точним і побудова графіку залежності похибки від аргумента.

7. Використовуючи модулі (функцію Modul[]) створіть власну функцію, яка: (2 бали)
 - o обчислює значення функції із завдання 1.5. Будує графік цієї функції на границях, заданих в завданні 1.5.