

***Графика. Структура на
графиките. Структура на
вградените графични функции.
Комбиниране на графични
елементи. Графика на функция.***

Структура на графиките

Всички графики в *Mathematica* са конструирани от обекти, които се наричат основни елементи, като ***точки, линии, кръгове, полигони***. Всички графики в *Mathematica* представляват съвкупност от основни елементи. А графичен обект е завършено графично изображение. Всички функции за графики генерират графични обекти.

Всяка графика се състои от три съставни части:

- *основни елементи*
- *директиви*
- *опции*

Основни елементи

точки, линии, кръгове, полигони.

Например, единичен кръг центриран в началото на координатната система. Използваме вход. който създава основен елемент - кръг

```
In[20]:= Circle[{0, 0}, 1]
```

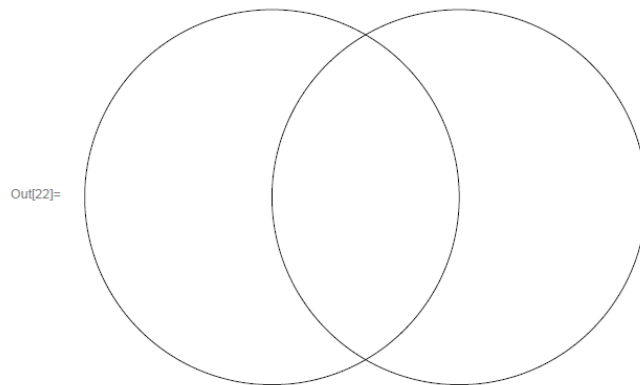
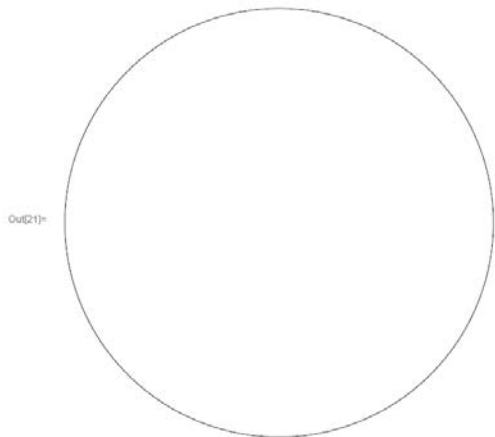
```
Out[20]= Circle[{0, 0}, 1]
```

За да се покаже **двумерна графика на основен елемент**, се използва `Graphics`.

За да се покажат повече от един основен елемент, се оформя списък.

```
In[22]:= Graphics[{Circle[{0, 0}, 1], Circle[{1, 0}, 1]}]
```

```
In[21]:= Graphics[Circle[{0, 0}, 1]]
```

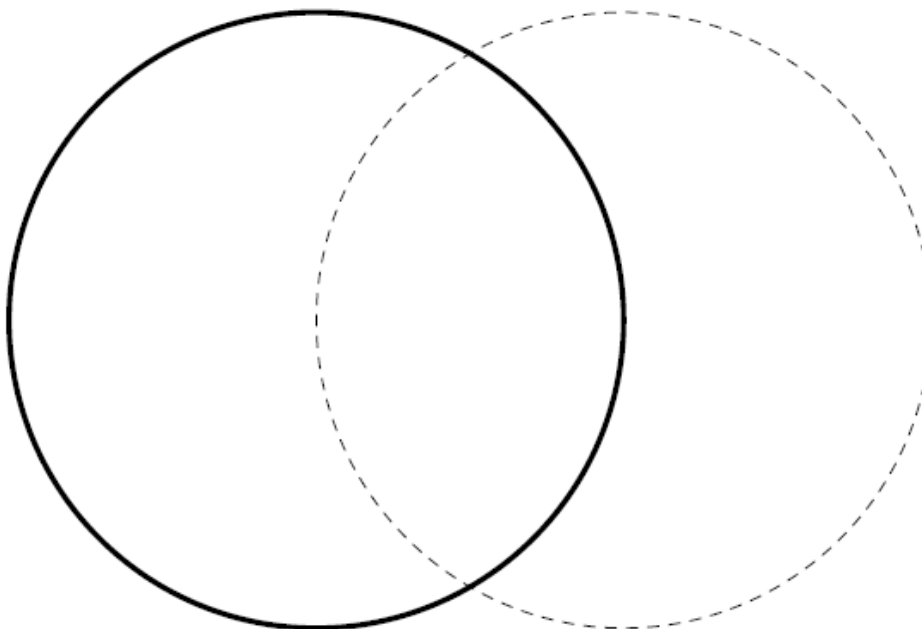


Графични директиви

Графичните директиви указват как да бъдат визуализирани основните графични елементи. Те се използват, за да се модифицират основните елементи. Например, единия кръг може да е плътен (Thick), а другия - пунктиран (Dashed).

```
In[23]:= Graphics[{{Thick, Circle[{0, 0}, 1]}, {Dashed, Circle[{1, 0}, 1]}}]
```

Out[23]=



Опции

Използват се като настройки, за да модифицират цялата графика

Цялата графика се оформя чрез използването на опциите. Ако лисват параметри за настройки(опции), то те са **по подразбиране**. Ако искаме допълнителна настройка, то е необходима **собствена настройка**, която се задава с опциите. Всяка опция има име. Те трябва да са след елементите или списъците от основни елементи, дадени като аргументи на функцията Graphics.

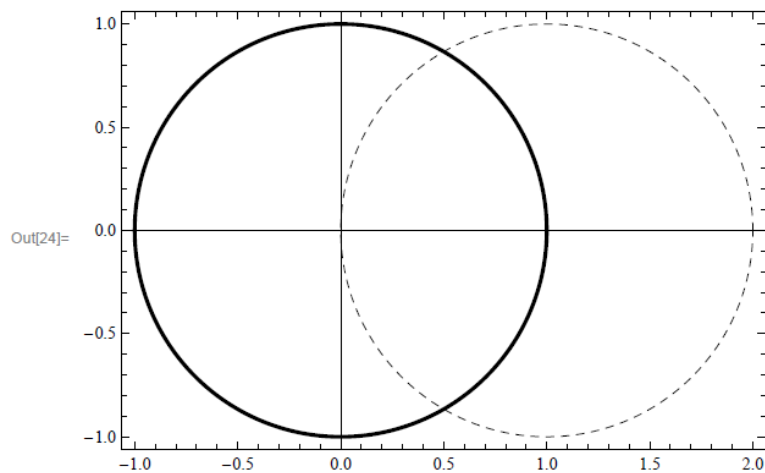
Задаването на потребителски настройки става чрез поредица от правила от вида

име->стойност.

Опции

Например, да добавим координатните оси (чрез Axes) и да сложим графиката в (чрез Frame)

```
In[24]:= Graphics[{{Thick, Circle[{0, 0}, 1]}, {Dashed, Circle[{1, 0}, 1]}},  
  Axes → True, Frame → True]
```



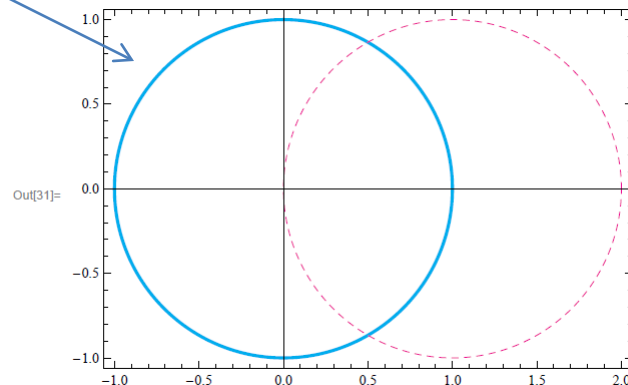
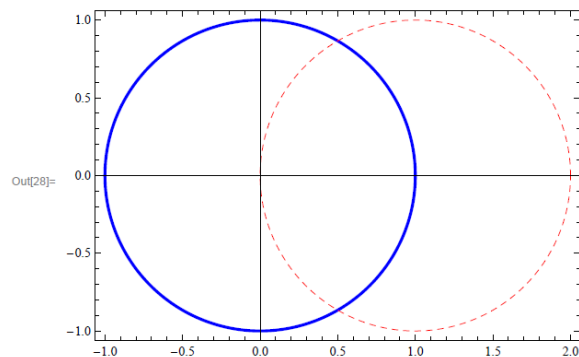
Смяна на цвят

Това е отново директива на графиката и става с помощта на **RGBColor[{r,g,b}]** – променя цвета на червен, зелен, син, като яркостта му се изразява с число от 0 до 1.

CMYKColor[{c,m,y,k}] променя цвета на синьо-зелен, червен, жълт, черен, като яркостта му се изразява с число от 0 до 1.

```
In[28]:= Graphics[{{RGBColor[0, 0, 1], Thick, Circle[{0, 0}, 1]},  
  { RGBColor[1, 0, 0], Dashed, Circle[{1, 0}, 1]}}, Axes → True, Frame → True]
```

```
In[31]:= Graphics[{{CMYKColor[{1, 0, 0, 0}], Thick, Circle[{0, 0}, 1]},  
  { CMYKColor[{0, 1, 0, 0}], Dashed, Circle[{1, 0}, 1]}}, Axes → True, Frame → True]
```



Тримерни графики

За 3D графика се използва вградената функция Graphics3D, която изобразява основните елементи.

Например, двете сфери като основни елементи са в Red директиви, а цилиндъра е в Blue директивата.

Използвани са две опции:

- FaceGrids добавя мрежата към всяка от стените на кутийката и
- ViewVertical е използвано за да смени вертикалната посока на изображението

```
In[25]:= Graphics3D[{{Red, Sphere[{0, 0, 0}], Sphere[{2, 2, 2}]},  
  {Blue, Cylinder[{0, 0, 0}, {2, 2, 2}], 0.2}}},  
  FaceGrids -> All, ViewVertical -> {1, 0, 0}]
```

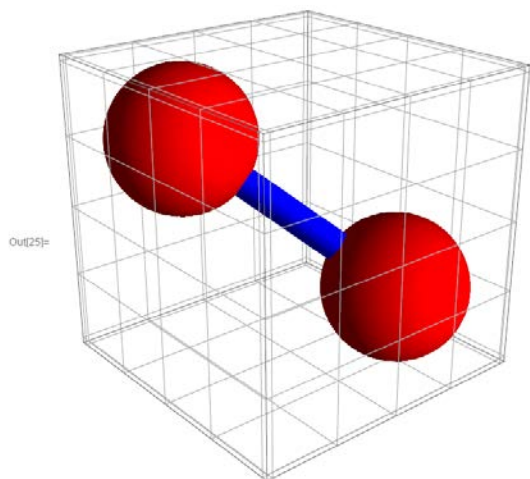


Таблица на основните графични елементи

<code>Point[{x, y}]</code>	Точка с дадени координати
<code>Line[{ {x₁, y₁}, {x₂, y₁}, ... }]</code>	Права през дадените точки
<code>Rectangle[{ {x_{min}, y_{min}}, {x_{max}, y_{max}} }]</code>	Плътен правоъгълник
<code>Polygon[{ {x₁, y₁}, {x₂, y₂}, ... }]</code>	Плътен полигон
<code>Circle[{x, y}, r, {θ₁, θ₂}]</code>	Дъга с радиус r
<code>Disk[{x, y}, r]</code>	Плътен диск
<code>Raster[{ {x₁₁, x₁₂, ...}, {x₂₁, x₂₂, ...}, ... }]</code>	Правоъгълна стрелка
<code>Text[expr, {x, y}]</code>	Текст, центриран в дадената точка
<code>Arrow[{pt₁, pt₂}]</code>	Стрелка,

Двумерна графика на функции

Използва се командата **Plot**, която има 2 аргумента, отделени със ,
Plot[функция, {x,xmin,xmax}]

Първият аргумент е функцията, чиято графика се чертае, а втория се нарича итератор, оформен като списък. Той описва интервала от стойности на аргумента, в които се чертае графиката: от xmin до xmax.

Командата Plot може да чертае и няколко графики едновременно на една и съща координатна система като в този случай има вида

Plot[{функция1, функция2,}, {x,xmin,xmax}]

Всеки път когато се използва **Plot**, Mathematica решава по подразбиране къде да сложи осите, и те НЕ ВИНАГИ се пресичат в началото (0,0). Ако искате да използвате осите, които се пресичат в началото, то трябва допълнително да се окаже това.

Например, да начертаете графиката на функцията $100\cos(x) e^x$ в интервала от -10 до 10

```
In[32]:= f[x_] := 100 * Cos[x] * Exp[x]
```

```
In[33]:= Plot[f[x], {x, -10, 10}]
```

Out[33]=



В този случай се оказва, че графиката е отрязана по y-оста, и е необходимо да се разшири множеството от стойности на функцията, дадени на графиката

Промяната на вертикалната скала става с **PlotRange**.

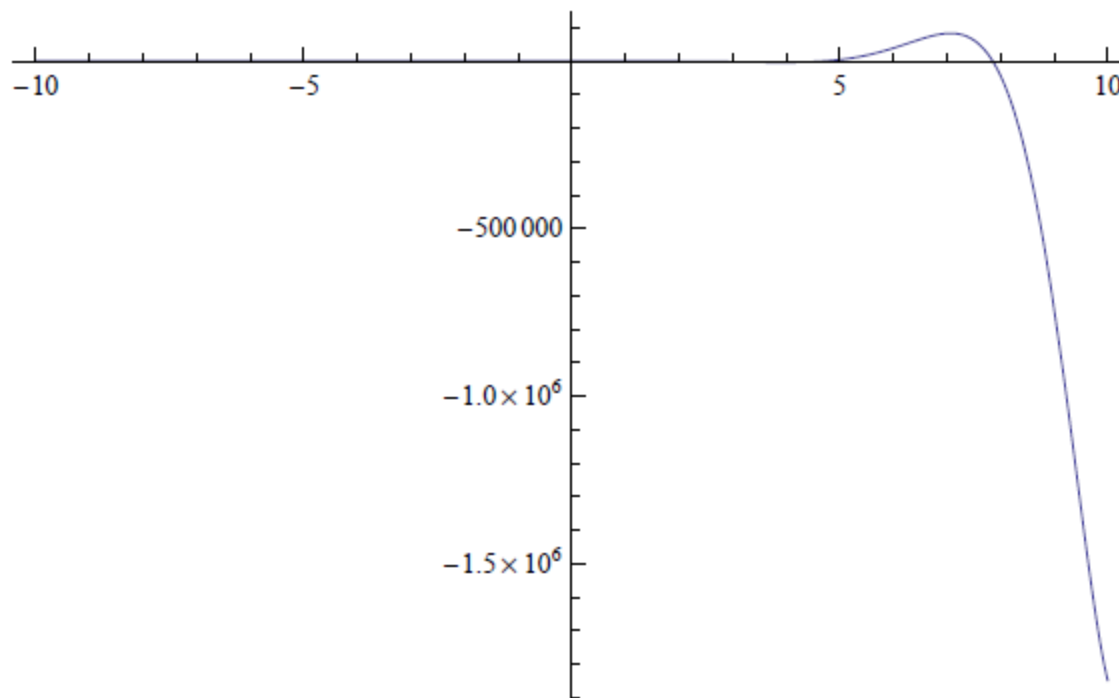
Тази вградена функция се слага като допълнителен аргумент на **Plot**. Трябва да се сложи след двата задължителни аргумента и е от вида

PlotRange ->Full или **PlotRange ->{a,b}**

Ще използваме **PlotRange ->Full**

```
In[34]:= Plot[f[x], {x, -10, 10}, PlotRange -> All]
```

Out[34]=

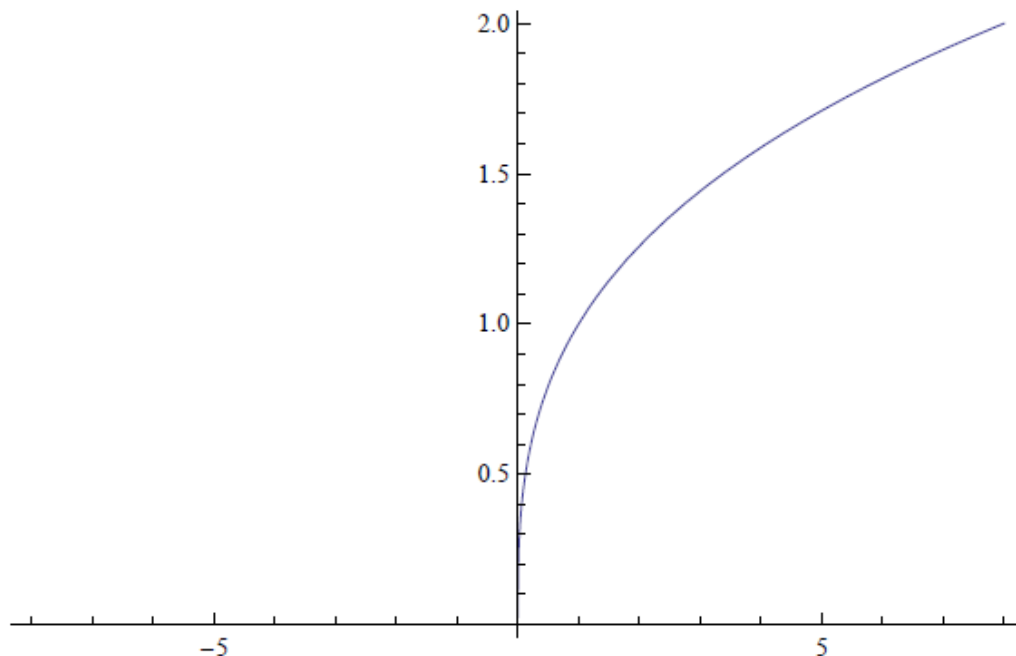


Някои куриозни примери:

Да начертаям графиката на кубичен корен от x

```
In[37]:= Plot[x^(1/3), {x, -8, 8}]
```

Out[37]=



Няма ли кубични корени от отрицателни числа. Ами кубичен корен от -8 е -2? А защо го няма на графиката? Причината е в представянето на корена като комплексни числа.

Основни функции за чертаене 2D графика на функции

Графичните команди зависят от вида на задаване на функцията и са систематизирани в следната таблица

Вид на задаване	функция	Пример	Графична команда
обикновен	$F(x)$	$\sin x$	Plot
параметричен	$X=f(t), y=g(t)$	$x=\sin(3t), y=\cos(2t)$	ParametricPlot
неявен	$f(x,y)=0$	$x^4 + y^5 = 0$	ContourPlot
Неявен неравенство	$f(x,y)>0$	$x^4 + y^5 > 0$	RegionPlot
Полярни координати	$r=f(\theta)$	$R=3\cos(5\theta)$	PolarPlot

ПРИМЕРИ

Да се начертаят графиките на следните функции

$$f(x) = \sin(x)/x$$

$$x = \sin(3t), y = \cos(4t)$$

$$x^4 - (x^2 - y^2) = 0$$

$$x^4 + (x - 2y^2) > 0$$

$$r = 3 \cos(6\theta).$$

`Plot[Sin[x]/x, {x, 0, 10 Pi}]`

`ParametricPlot[{Sin[3 t], Cos[4 t]}, {t, 0, 2 Pi}]`

`ContourPlot[x^4 - (x^2 - y^2) == 0, {x, -1, 1}, {y, -1, 1}]`

`RegionPlot[x^4 + (x - 2 y^2) >= 0, {x, -2, 2}, {y, -2, 2}]`

`PolarPlot[3 Cos[6 t], {t, 0, 2 Pi}]`

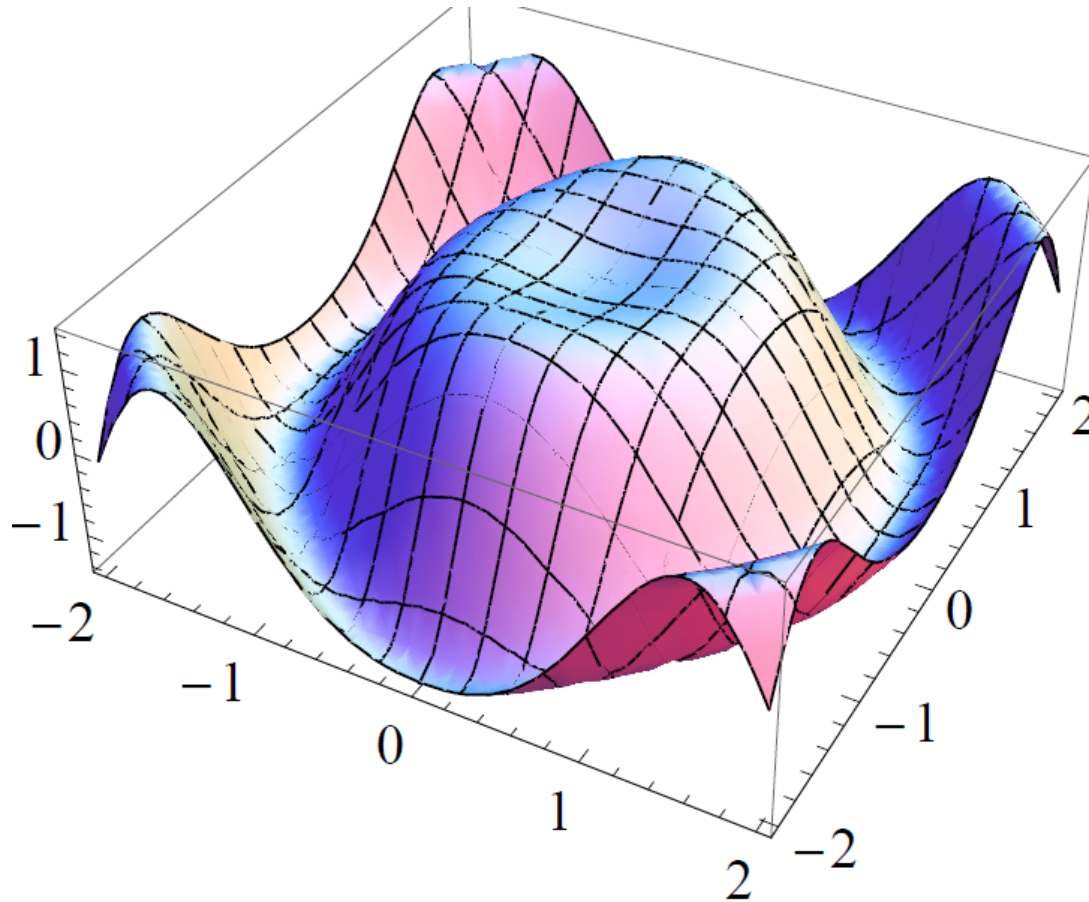
Основни функции за чертаене 3D графика на функции

Графичните команди зависят от вида на задаване на функцията и са систематизирани в следната таблица

Вид на задаване	функция	Пример	Графична команда
обикновен	$f(x,y)$	$\sin(x+y)$	Plot3D
параметричен	$X=f(t), y=g(t), z=h(t)$	$x=\sin(3t), y=\cos(2t), z=\sin(5t)$	ParametricPlot3D
неявен	$f(x,y)=0$	$x^4 + y^5 - 2xz = 0$	ContourPlot3D
Неявен неравенство	$f(x,y)>0$	$x^4 + y^5 - 2xz > 0$	RegionPlot3D

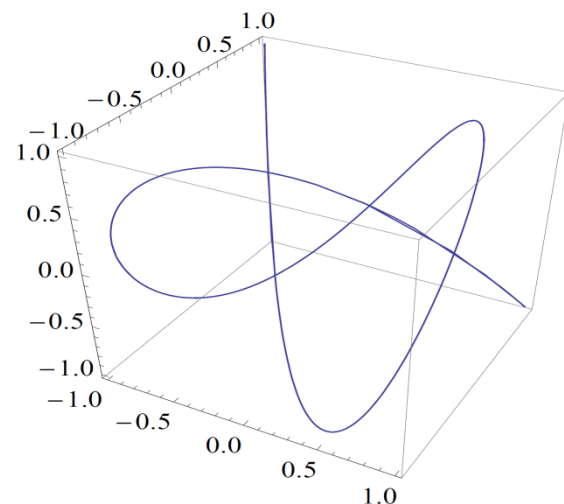
Пример. Начертайте графиката на функцията $\sin(x^2 + y^2)e^{-x^2} + \cos(x^2 + y^2)$ когато x и y се изменят от -2 до 2.

```
f[x_,y_]=Sin[x^2+y^2]*Exp[-x^2]+Cos[x^2+y^2];  
Plot3D[f[x,y],{x,-2,2},{y,-2,2}]
```



Пример. Начертайте графиката на $x=\sin 3t$, $y=\cos 4t$, $z=\sin 5t$ в интервала $[-\pi, \pi]$

`ParametricPlot3D[{Sin[3 t],Cos[4 t], Sin[5 t]},{t,-Pi,Pi}]`



За да получим по добра пространствена графика добавяме нова опция

`ParametricPlot3D[{Sin[3 t],Cos[4 t], Sin[5 t]},{t, -Pi, Pi}, PlotStyle->Tube[0.05]]`

Настройки на графиката

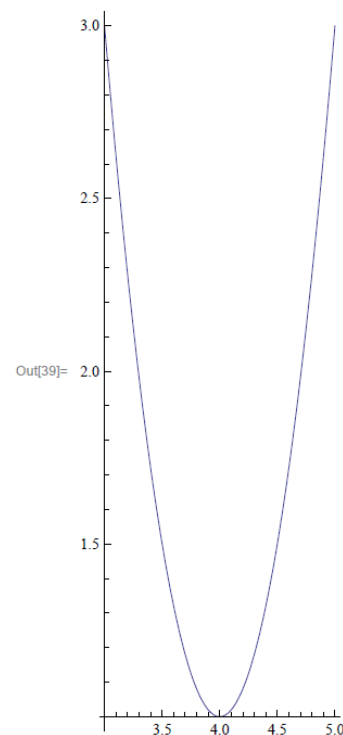
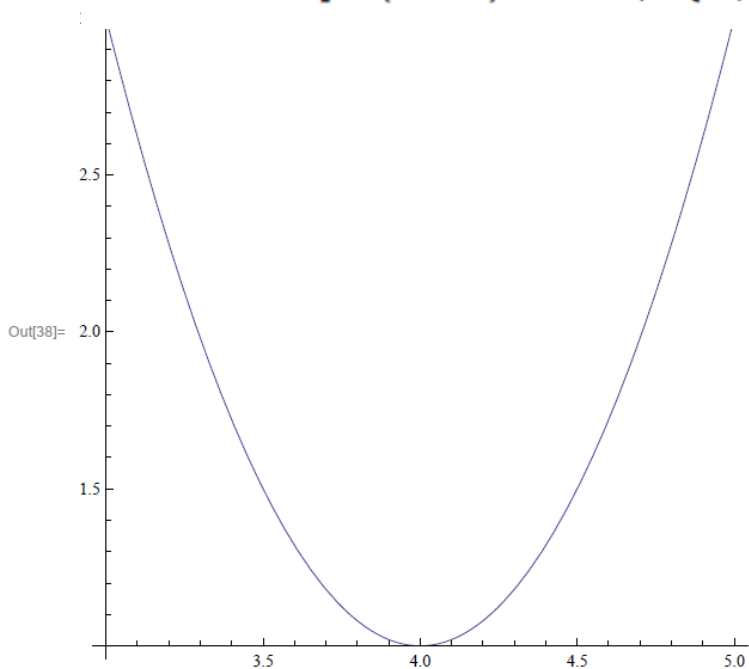
Как да получим една и съща скала по двете оси

AspectRatio->Automatic

```
In[38]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, AspectRatio -> Automatic]
```

Може да използваме и **AspectRatio->** числена стойност, например **AspectRatio->3** ще чертае графика на която височината е 3 пъти по голяма от ширината.

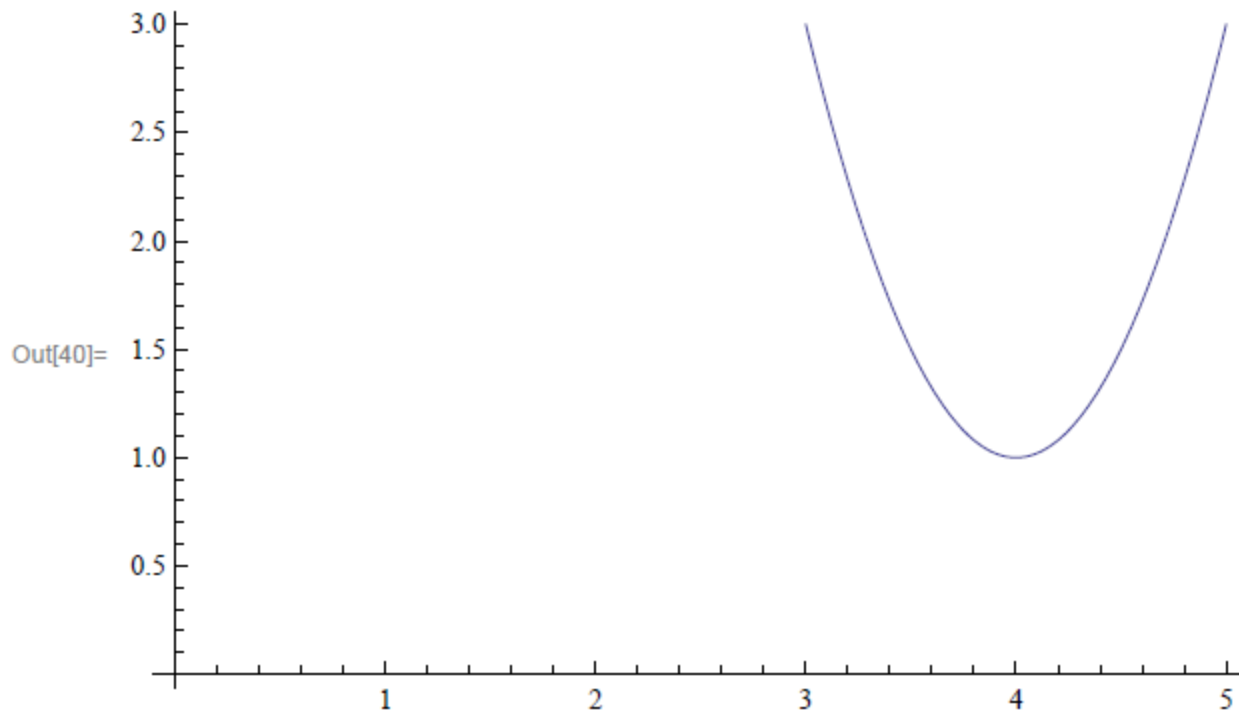
```
In[39]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, AspectRatio ->
```



Как да получим графика, на която двете оси се пресичат в началото

AxesOrigin -> {0, 0}

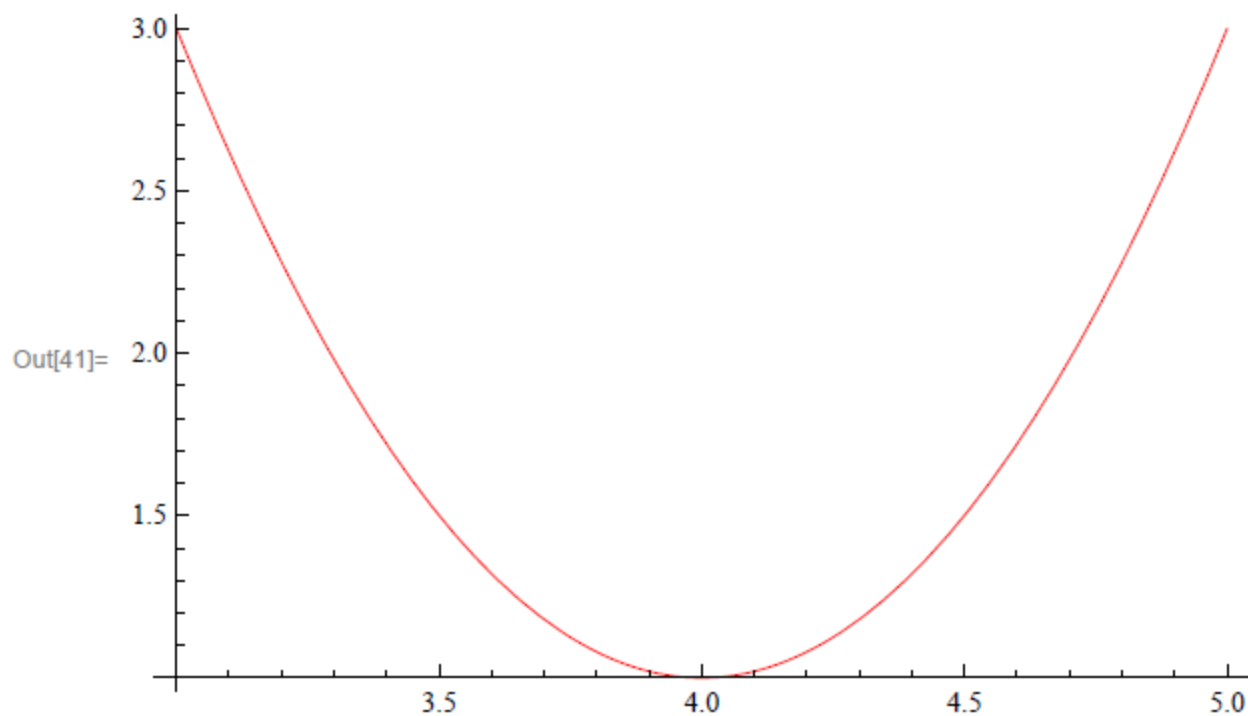
```
In[40]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, AxesOrigin -> {0, 0}]
```



Как да използваме директивите на графиката

ЦВЯТ: това става с **PlotStyle**

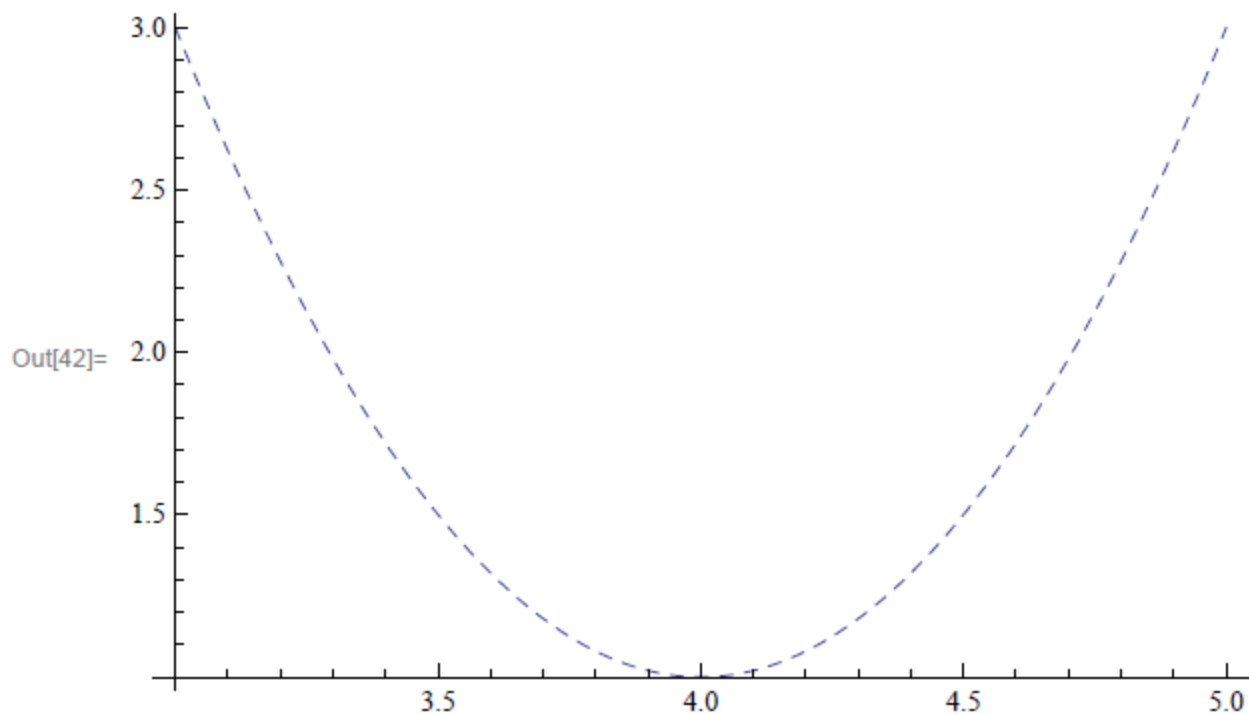
```
In[41]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, PlotStyle -> Red]
```



PlotStyle дава възможност да се използват и други графични директиви като **Thick**, **Gray**, **Dashed**, които трябва да бъдат сложени в **Directive**

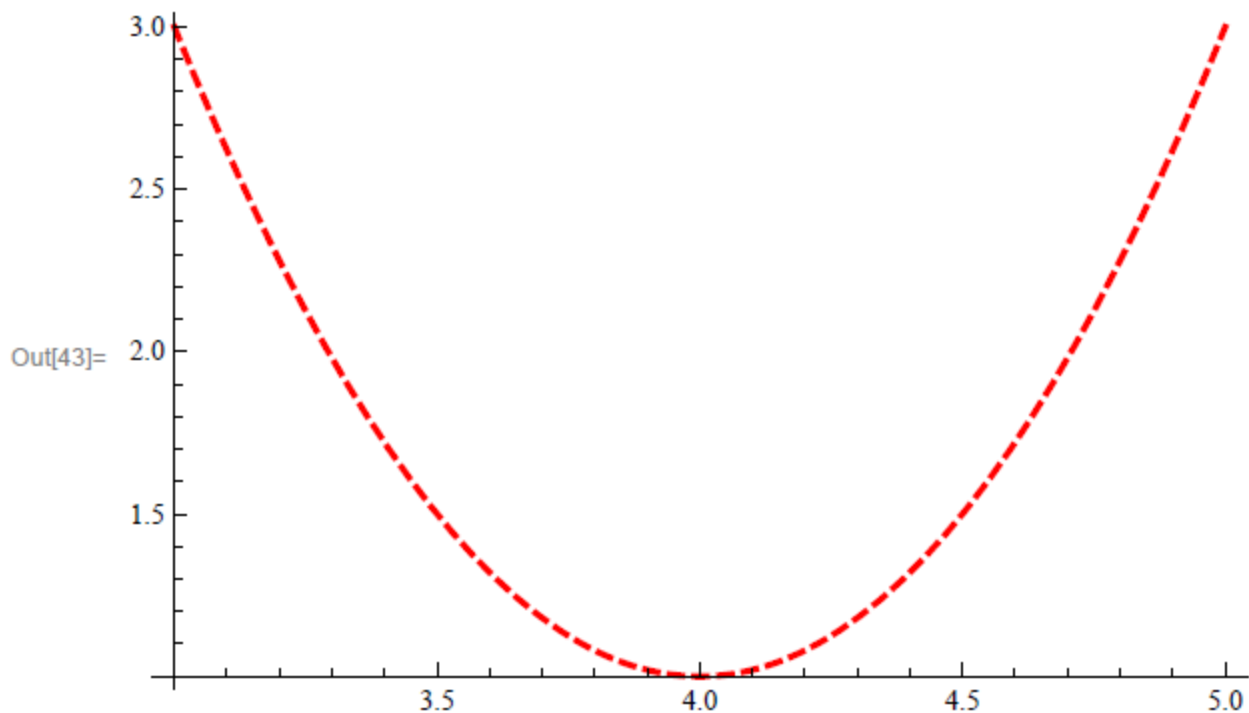
Например, графиката може да бъде пунктирна

```
In[42]:= Plot[2 (x - 4) ^ 2 + 1, {x, 3, 5}, PlotStyle -> Directive[Dashed]]
```



Например, графиката може да бъде червена, пунктирна и удебелена

```
In[43]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, PlotStyle -> Directive[Thick, Red, Dashed]]
```



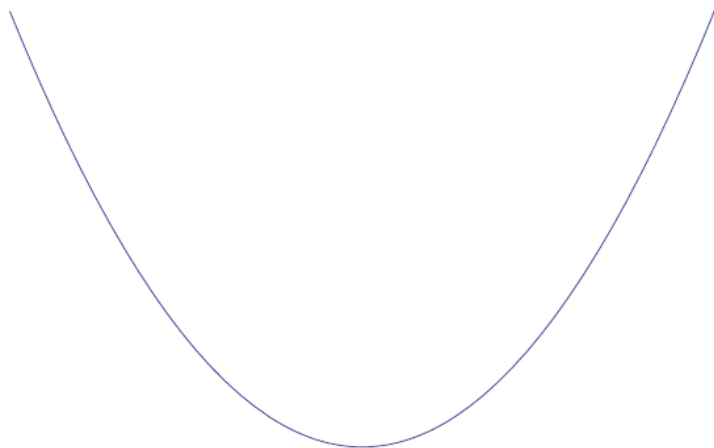
Как да махнем координатните оси или да добавим рамка

За да премахнем осите, използваме **Axes -> False**

За да сложим графиката в рамка, използваме **Frame -> True**:

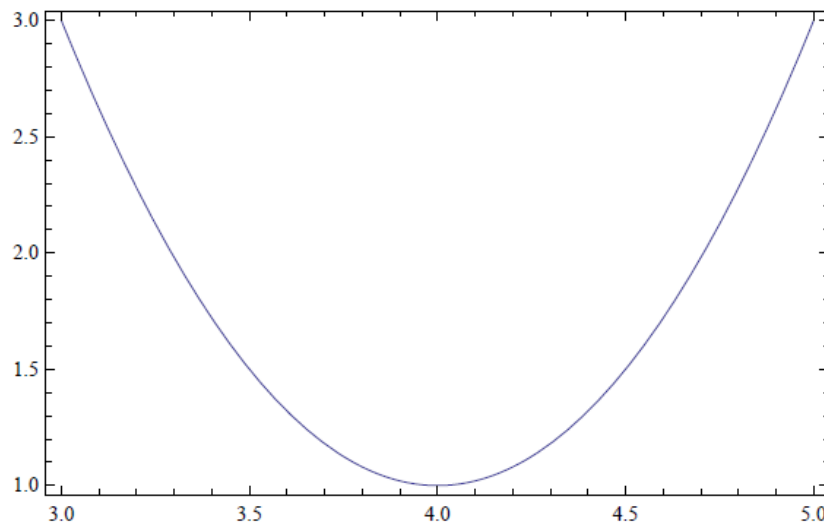
```
In[44]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, Axes -> False]
```

Out[44]=



```
In[45]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, Frame -> True]
```

Out[45]=

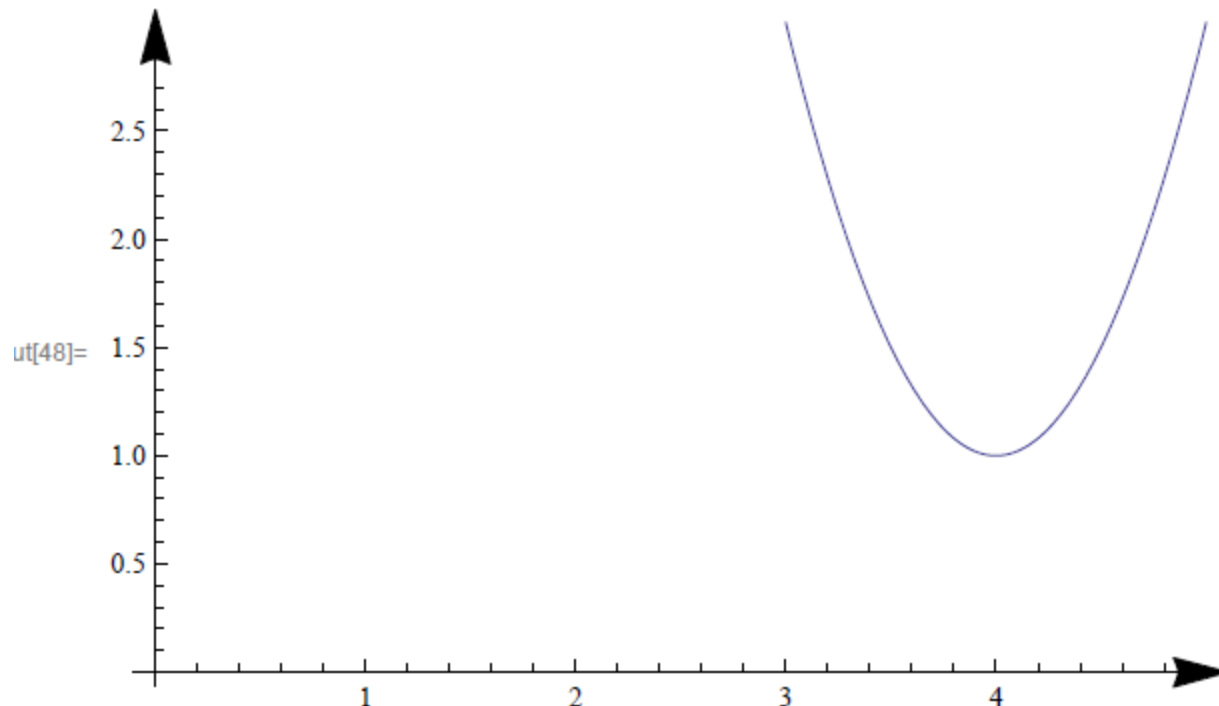


Как да сложим стрелки на координатните оси

Използваме **AxesStyle -> Arrowheads[0.05]** за да поставим стрелки само нагоре и надясно

Използваме **AxesStyle -> Arrowheads[{-0.05, 0.05}]** за да поставим стрелки във всички посоки

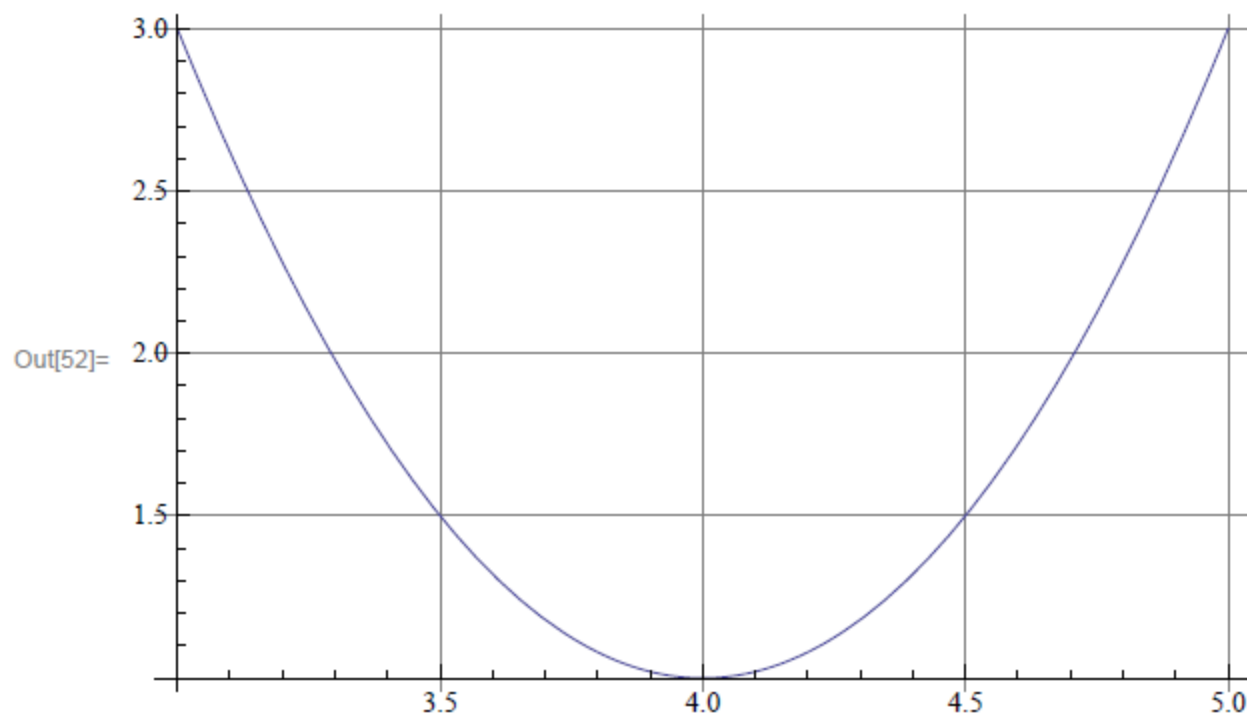
```
Plot[2 (x - 4) ^ 2 + 1, {x, 3, 5},  
  AxesOrigin -> {0, 0}, AxesStyle -> Arrowheads[0.05]]
```



Как да добавим координатна мрежа

GridLines -> Automatic

```
In[52]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, GridLines -> Automatic]
```

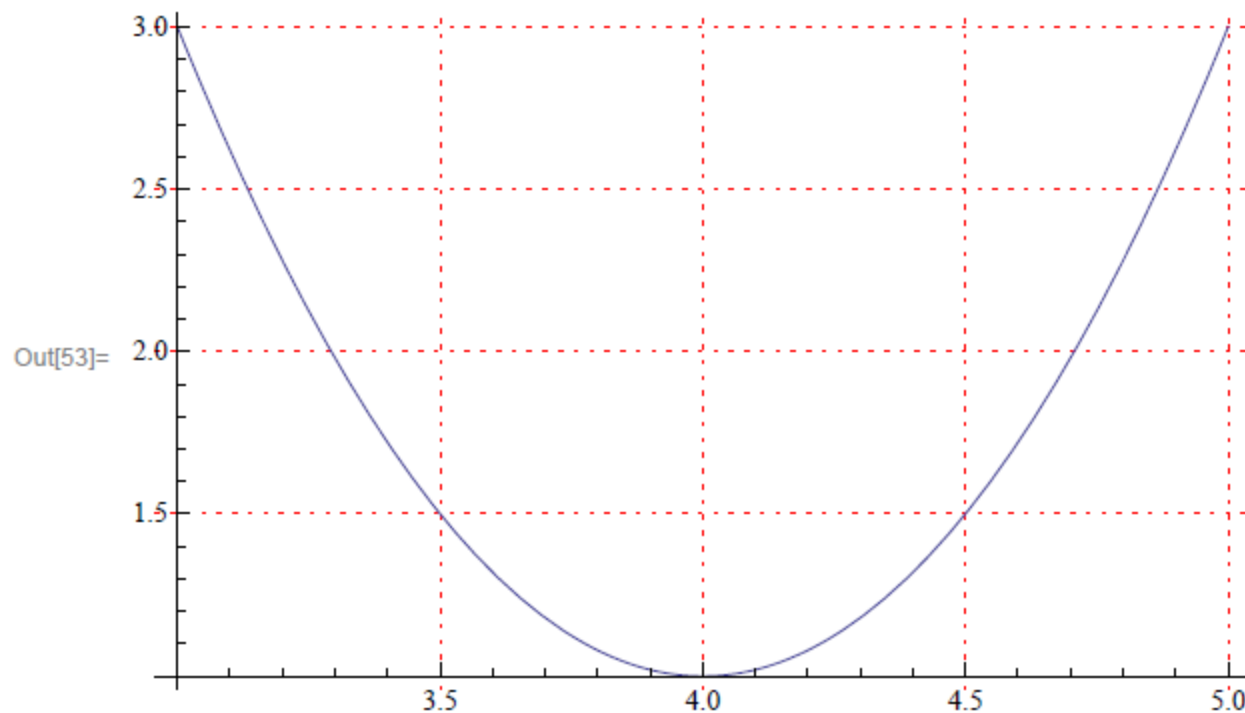


Съчетаване на отделните възможности за чертаене на графики

Например, координатната мрежа може да е синя пунктирна и тънка.

Това става с използването на **GridLinesStyle** и на **Directive**

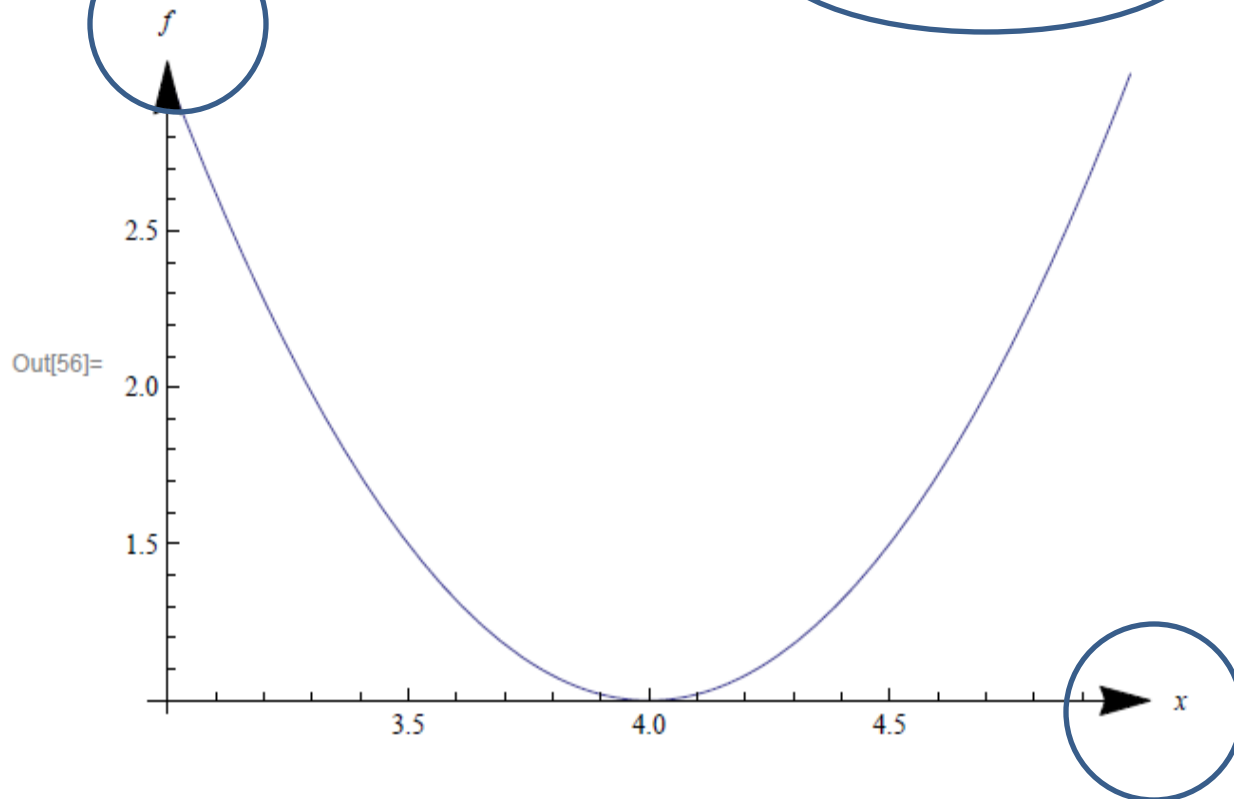
```
In[53]:= Plot[2 (x - 4) ^2 + 1, {x, 3, 5}, GridLines -> Automatic,  
GridLinesStyle -> Directive[Thin, Red, Dotted]]
```



Как да добавяме етикети

Етикети на координатните оси могат да се добавят с **AxesLabel**

```
In[56]:= Plot[2 (x - 4) ^ 2 + 1, {x, 3, 5}, AxesLabel -> {x, f}, AxesStyle -> Arrowheads[0.05]]
```

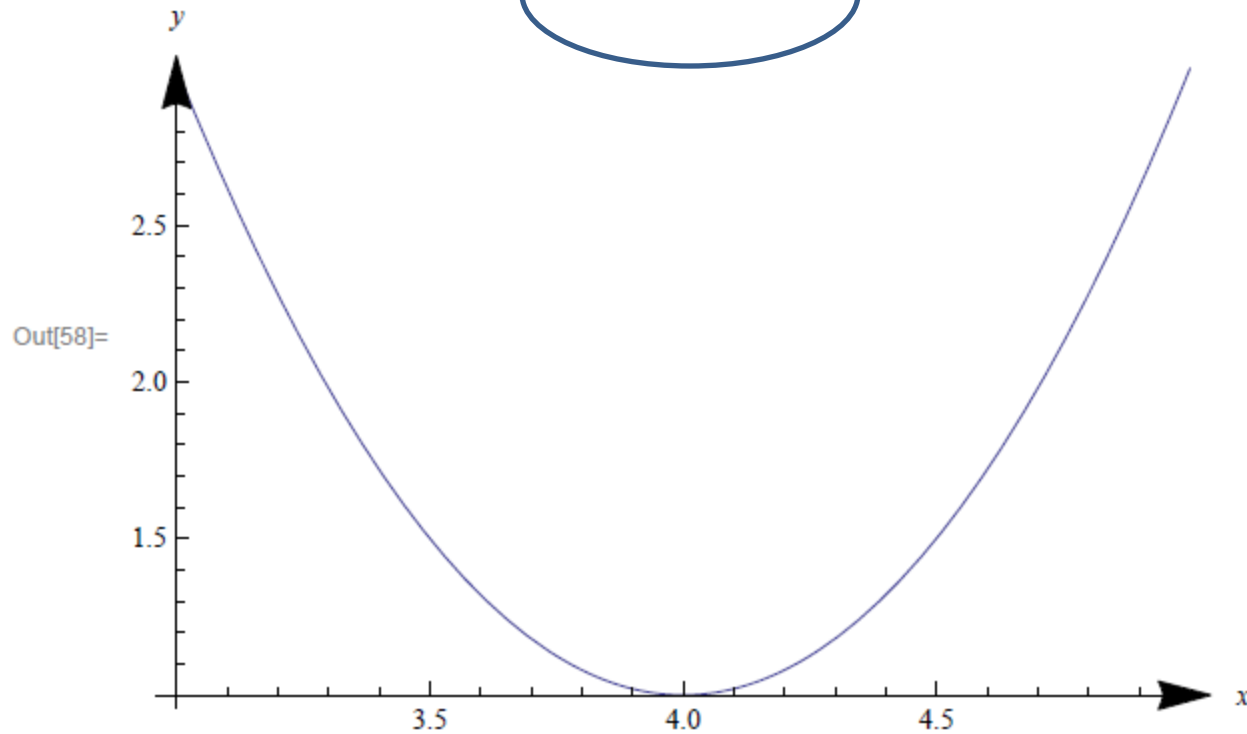


Етикети, които съдържат оператори(като =), или съдържат повече от една дума трябва да се оформят като стринг, т.е. да се поставят в кавички. В този случай, текстът между кавичките се възпроизвежда точно както е написан.

Забележка. Ако текстът е написан на кирилица, при запазването в пдф формат може да се получи грешка.

```
In[58]:= Plot[2 (x - 4)^2 + 1, {x, 3, 5}, AxesLabel -> {x, y},  
PlotLabel -> "y=2 (x-4)^2 + 1" AxesStyle -> Arrowheads[0.05]]
```

$$y=2(x-4)^2+1$$



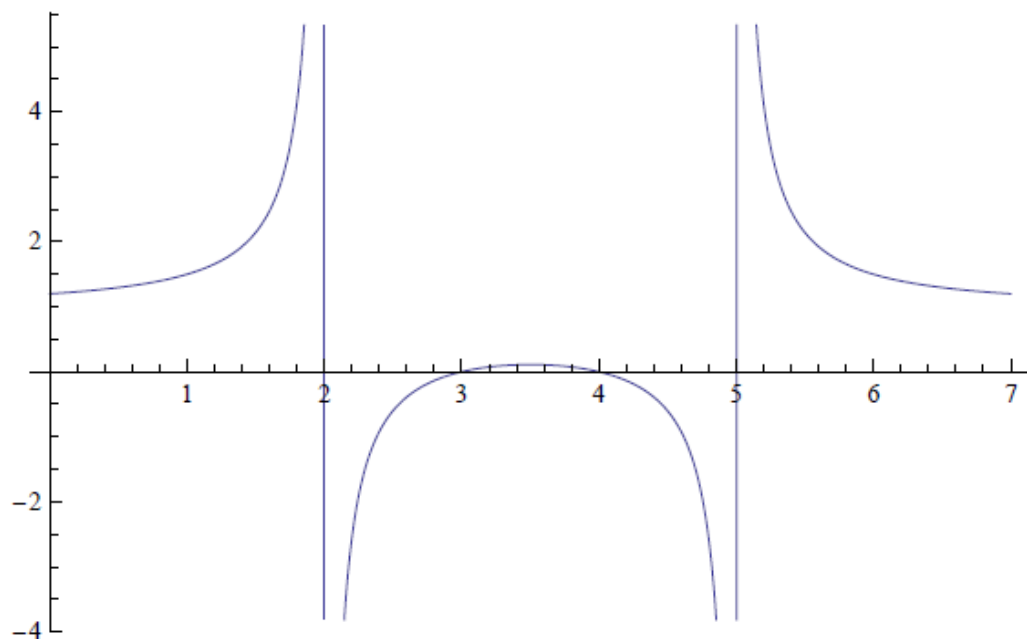
Исключвания и вертикални асимптоти

Например, да начертаете графиката на функцията $f(x) = \frac{(x-3)(x-4)}{(x-2)(x-5)}$

Дефинираме функцията

`f[x_] := (x - 3) * {x - 4} / ((x - 2) * (x - 5))` И начертаваме графиката ѝ

```
Plot[f[x], {x, 0, 7}]
```



Функцията има вертикални асимптоти, които също се изчертават.

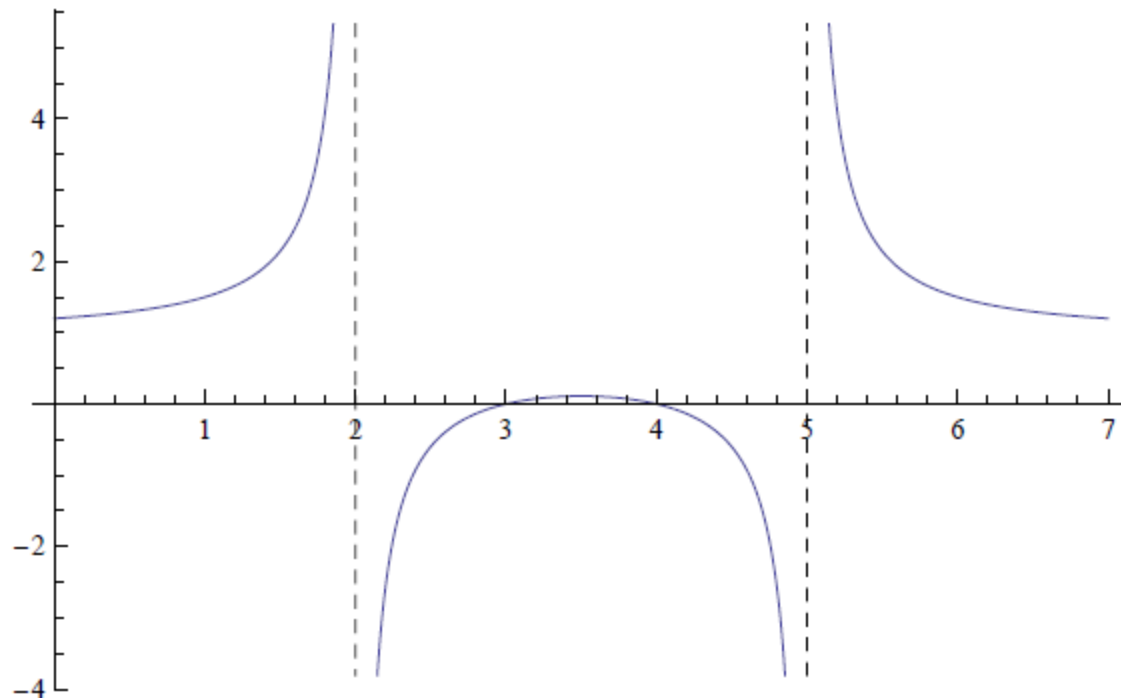
Ако искаме да не се чертаят вертикални асимптоти, то изключваме точките 2 и 5, и да използваме

Exclusions $\rightarrow \{x == 2, x == 5\}$

Ако искаме асимптотите да са пунктирани, то използваме

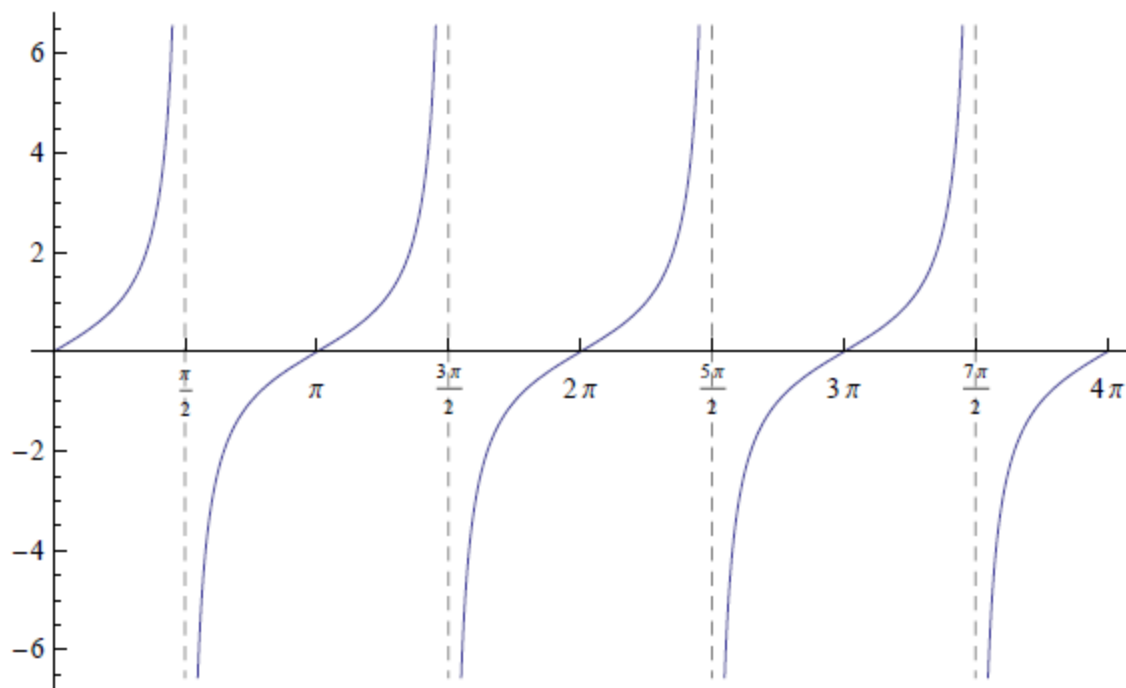
ExclusionsStyle $\rightarrow$ Dashed

```
Plot[f[x], {x, 0, 7}, Exclusions  $\rightarrow$  {x == 2, x == 5}, ExclusionsStyle  $\rightarrow$  Dashed]
```



Предимството на изключванията се вижда особено добре на графиката на функцията тангенс, където тя има много асимптоти в разглеждания интервал и това са точките за които косинуса се анулира.

```
Plot[Tan[x], {x, 0, 4 Pi}, Exclusions -> {Cos[x] == 0},  
ExclusionsStyle -> Directive[Gray, Dashed],  
Ticks -> {Range[0, 4 Pi, Pi / 2], Automatic}]
```



Работа с частично непрекъснати функции

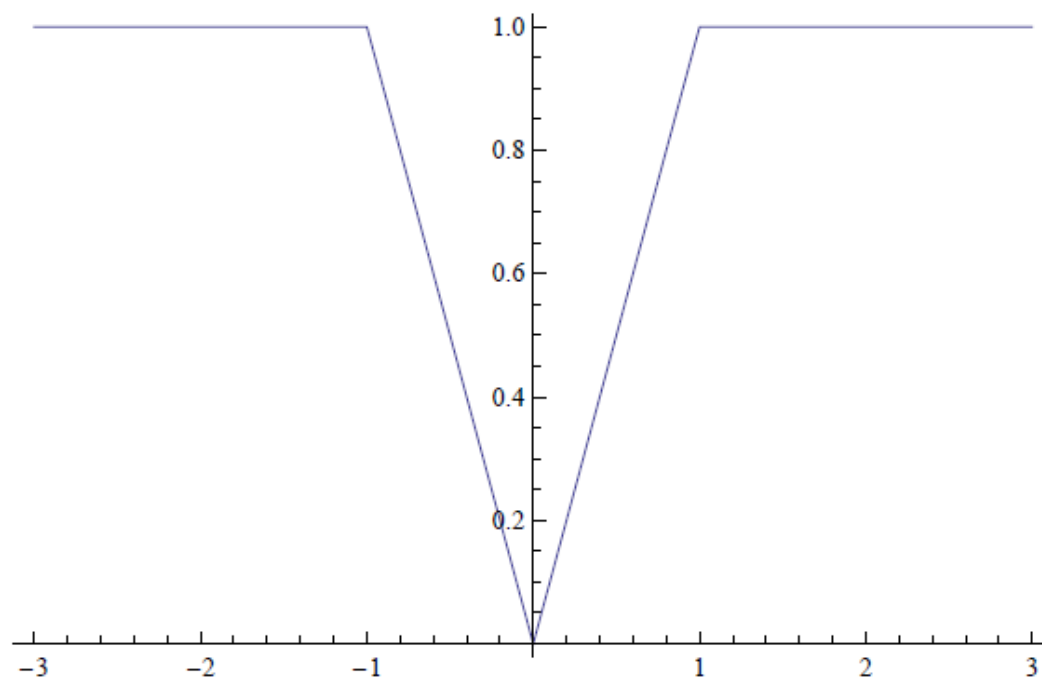
$$f(x) = \begin{cases} x & 0 \leq x \leq 1 \\ -x & -1 \leq x < 0 \\ 1 & \text{в противен случай} \end{cases}$$

За да дефинираме функцията използваме вградената функция **Piecewise**

```
In[2]:= f[x_] := Piecewise[{{x, 0 ≤ x ≤ 1}, {-x, -1 ≤ x < 0}, {1, x < -1}, {1, x > 1}}]
```

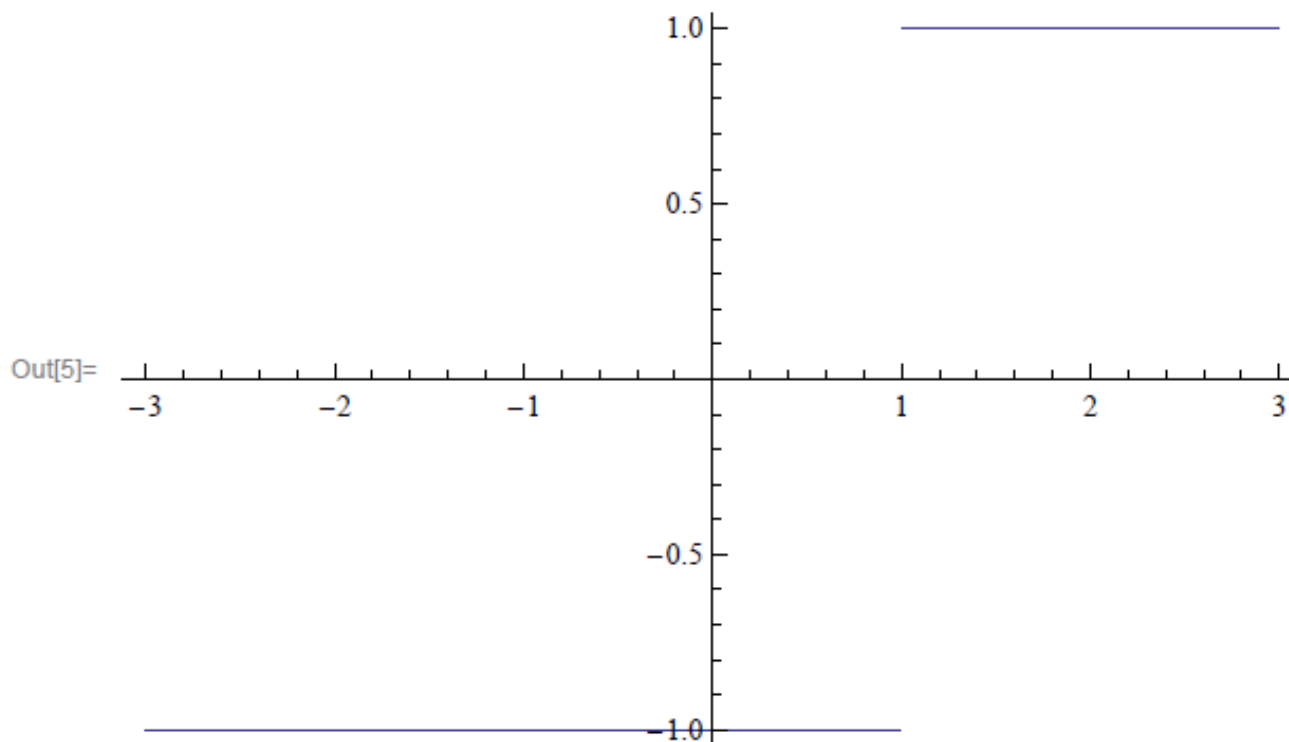
```
In[3]:= Plot[f[x], {x, -3, 3}]
```

Out[3]=



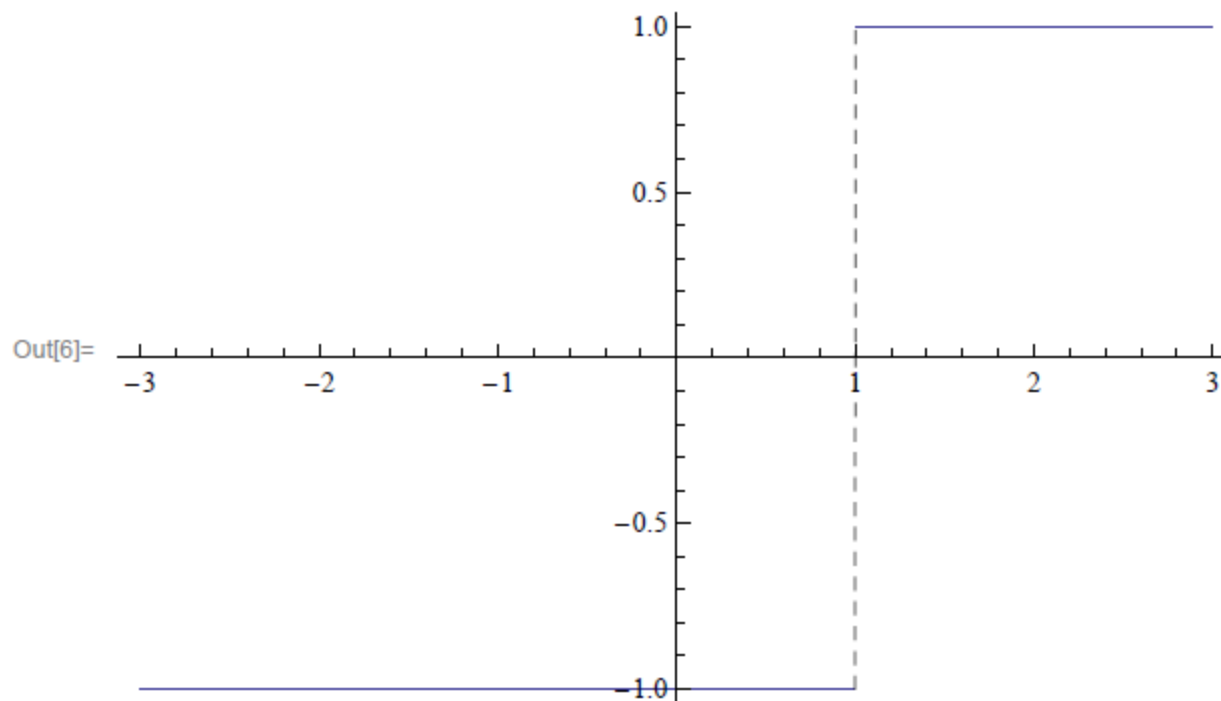
Да разгледаме прекъснатата функция, която приема стойност -1 при $x < 1$ и 1 в останалите случаи. Тогава графиката се чертае с

```
In[5]:= Plot[Piecewise[{{-1, x < 1}, {1, x ≥ 1}}], {x, -3, 3}]
```



А може да се начертае и пунктира

```
In[6]:= Plot[Piecewise[{{-1, x < 1}, {1, x ≥ 1}}], {x, -3, 3}, ExclusionsStyle → Dashed]
```



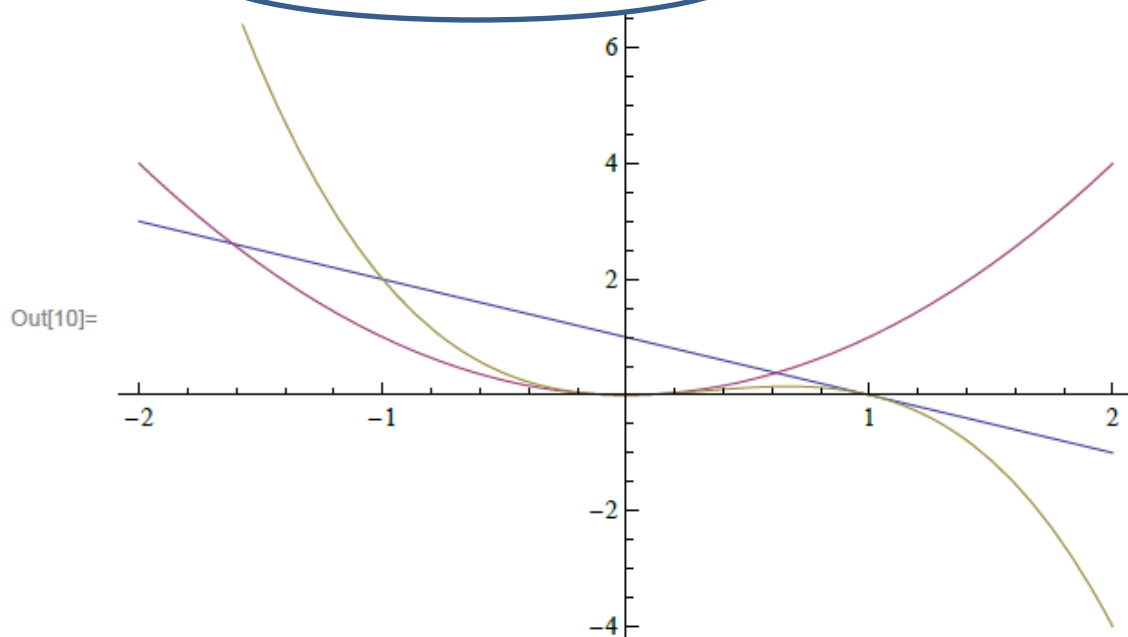
Комбиниране на графики

Понякога се налага да се начертаят в една и съща координатна система две и повече графики. Тогава съставяме списък от тези функции като първи аргументи на командата **Plot**

```
In[8]:= f[x_] := 1 - x;
```

```
In[9]:= g[x_] := x^2
```

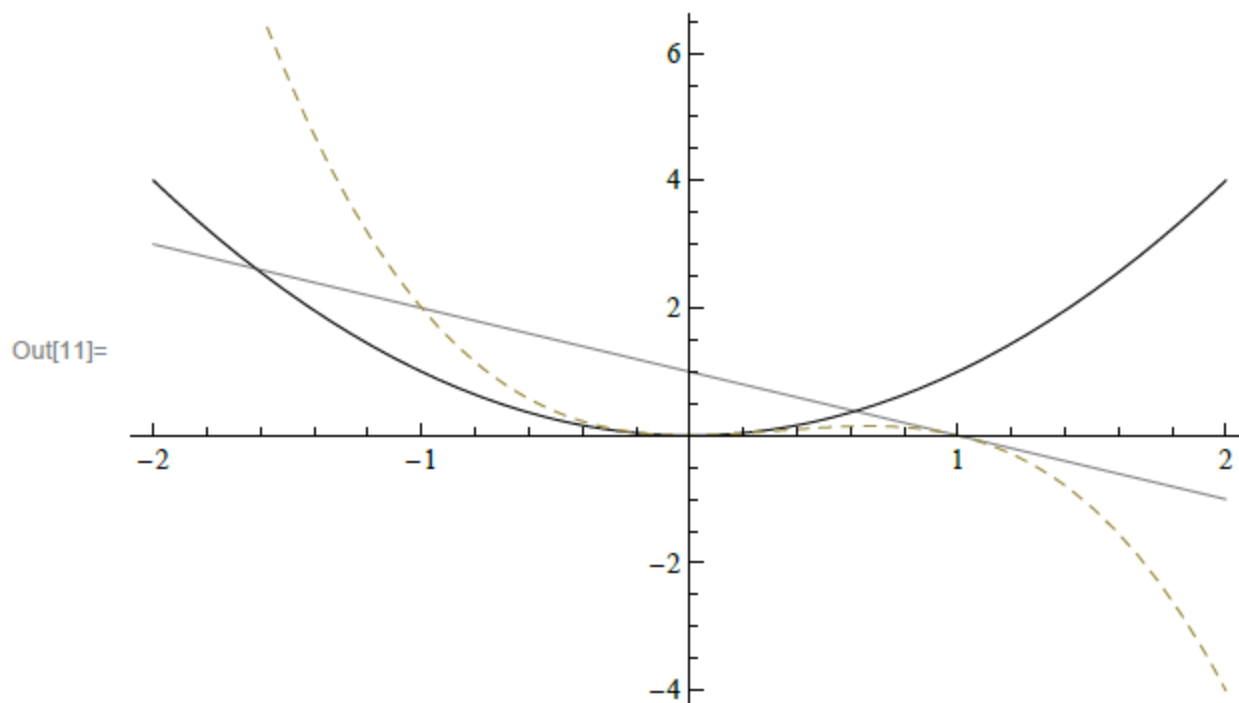
```
In[10]:= Plot[{f[x], g[x], f[x] * g[x]}, {x, -2, 2}]
```



На екрана трите графики са дадени с различни цветове

Може да използваме **PlotStyle** за да сменим вида на трите функции. Особено полезно е това когато се принтират графиките в сиво-черна гама. Поставете PlotStyle към списъка на трите директиви. Те ще се приложат към функциите в реда, които са зададени.

```
In[11]:= Plot[{f[x], g[x], f[x] * g[x]}, {x, -2, 2},  
PlotStyle -> {Gray, Black, Dashing[0.01]}}
```



Добавяне на легенда към графиката

Първоначално трябва да заредим пакета **PlotLegends** .

Използваме командата **Needs**, като не забравяйте кавичките “ и обратния акцент `

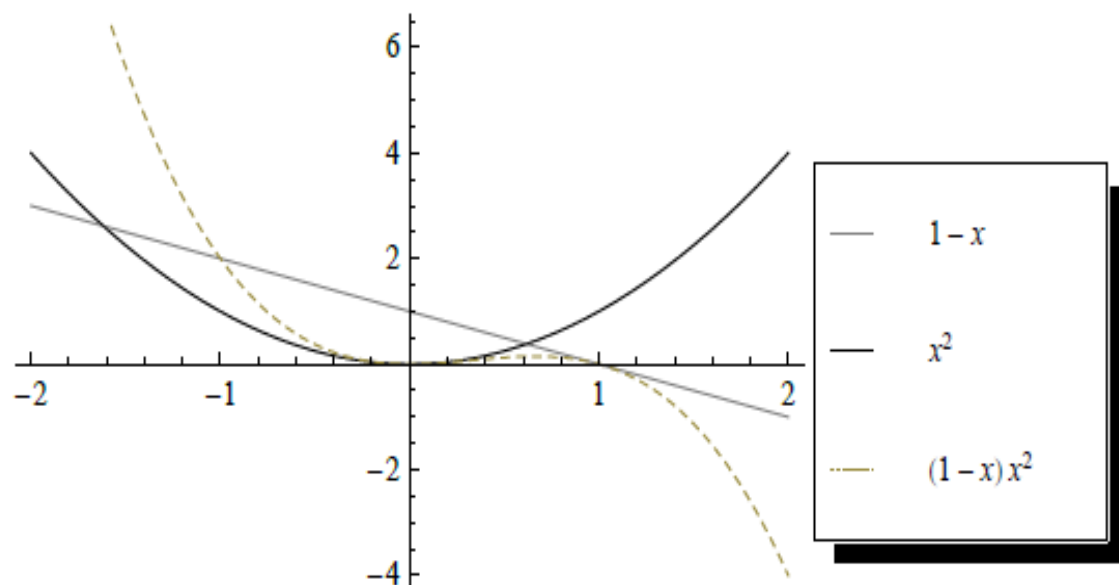
```
In[18]:= Needs["PlotLegends`"]
```

```
In[19]:= Plot[{f[x], g[x], f[x] * g[x]}, {x, -2, 2},  
  PlotStyle → {Gray, Black, Dashing[0.01]},  
  PlotLegend → {f[x], g[x], f[x] * g[x]}, LegendPosition → {1, -0.5}]
```

Командата **PlotLegend** дава възможност да се избира позицията на легендата, по-точно тя определя къде е средата на лявата страна на правоъгълника с легендата. Ако в **LegendPosition** е поставено $\{-1, -1\}$, то легендата се поставя в горния ляв ъгъл, а $\{1, 1\}$ поставя легендата в горния десен ъгъл..

```
In[19]:= Plot[{f[x], g[x], f[x] * g[x]}, {x, -2, 2},  
PlotStyle -> {Gray, Black, Dashing[0.01]},  
PlotLegend -> {f[x], g[x], f[x] * g[x]}, LegendPosition -> {1, -0.5}]
```

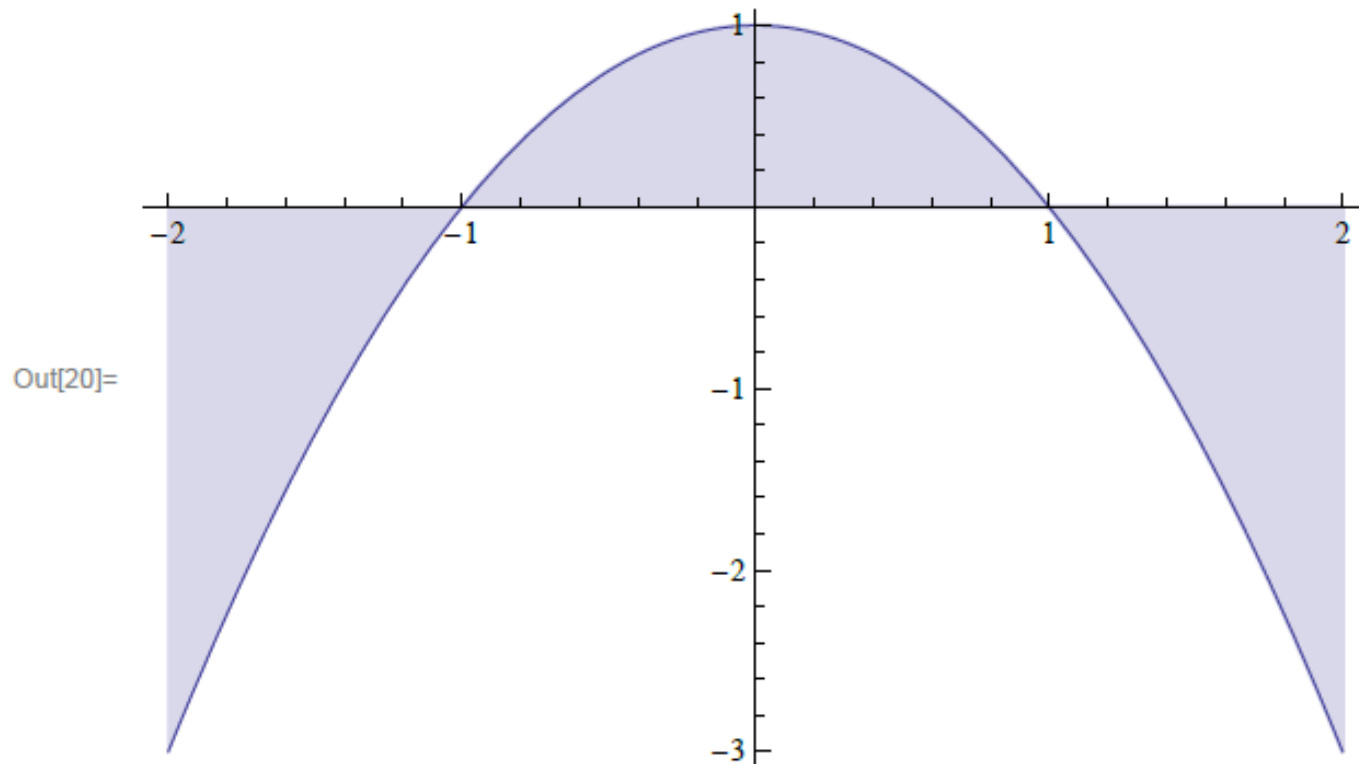
Out[19]=



Затъмняване на част от графиката

Затъмняване на областта между графиката и x-оста

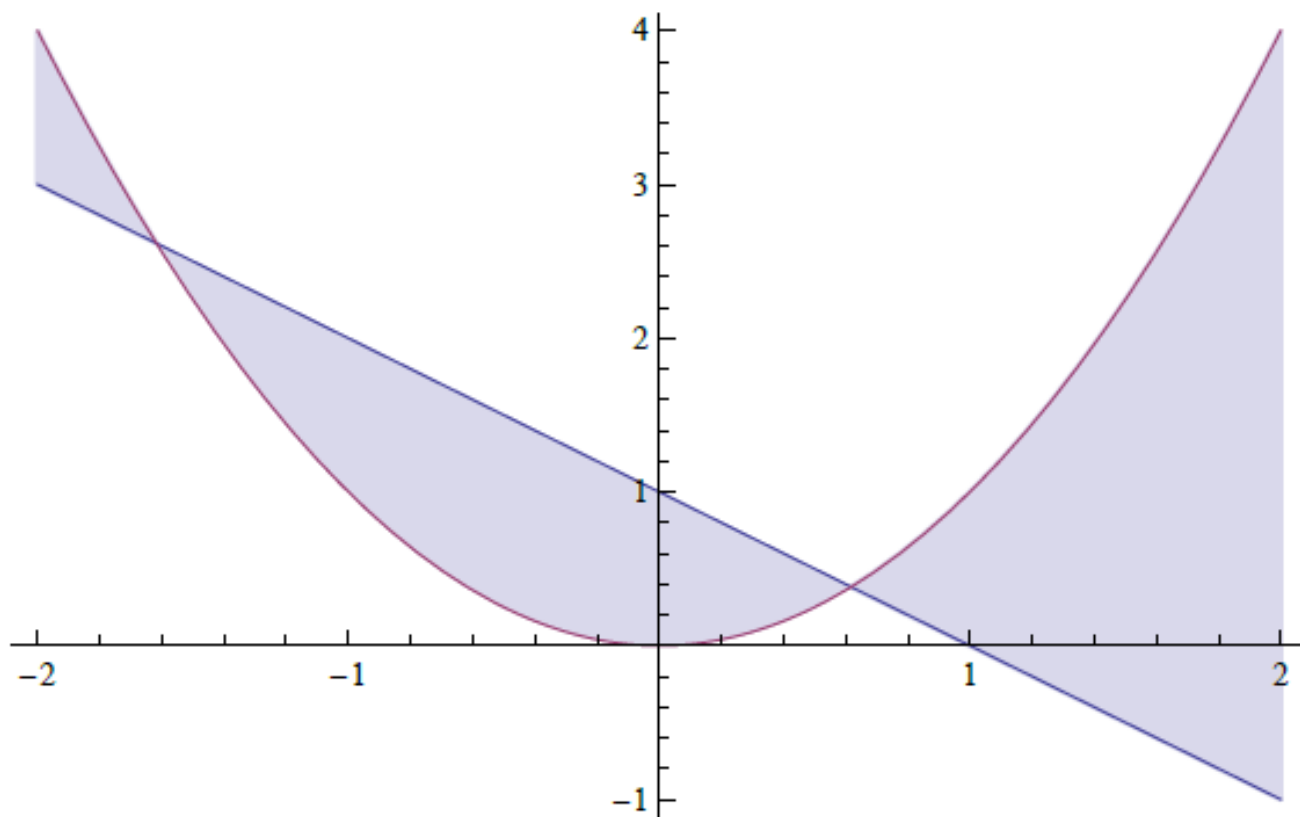
```
In[20]:= Plot[1 - x^2, {x, -2, 2}, Filling -> Axis]
```



Запълване областта между две графики

In[21]:= `Plot[{f[x], g[x]}, {x, -2, 2}, Filling -> {1}]`

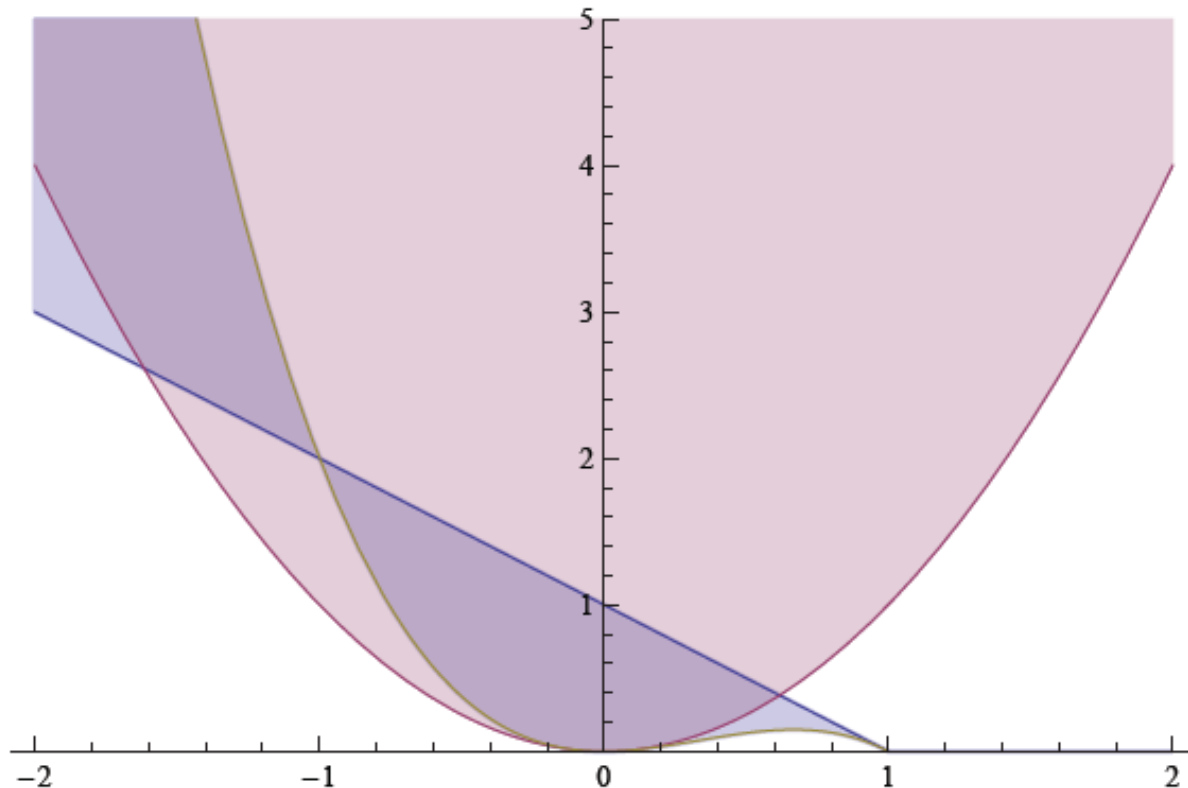
Out[21]=



Когато имаме повече от 2 графики, има различни начини за запълване на области.

```
In[22]:= Plot[{f[x], g[x], f[x] * g[x]}, {x, -2, 2},  
PlotRange -> {0, 5}, Filling -> {1 -> {3}, 2 -> Top}]
```

Out[22]=



КРАЙ

Задача за самостоятелна работа

Начертайте графиката на функцията, като използвате подходящо самостоятелни настройки

$$\frac{\cos(15x)}{1+x^2}$$