

Κεφάλαιο 9ο: Γραφικές Παραστάσεις

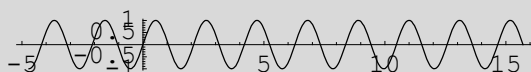
9.1 Διδιάστατες Γραφικές Παραστάσεις

9.1.1 Γραφική Παραστέλιση Καμπυλών

Όταν ορίστε να κάνετε την γραφική παράσταση μιας καμπύλης hV πρέπει κατά αρχή να δώσετε με ποίο τρόπο περιγράφεται η hV . Περιπτώσεις:

1. Η καμπύλη δίνεται ως γραμμή μιας συνάρτησης hV του x . Π.χ $f[x]=\text{Cos}[x]$. Τότε μπορούμε να χρησιμοποιήσουμε την Plot:

```
Plot@Sin@3 xD, {x, -3 Pi/2, 5 Pi/2}, AspectRatio -> AutomaticD;
```



Σκόλια: Πατώντας πάνω στο γραφικό με το ποντίκι ο δείκτης σε χρόνο που έχει ένα σταυρό πάνω και εμφανίζεται ένα πλαίσιο με labels. Με το αριστερό πλήκτρο του ποντικιού μπορούμε να μετακινήσουμε το γραφικό. Με τη V "label" μπορούμε να αλλάξουμε τη διάσταση του γραφικού, συντονίζοντας διάστρον κατακόρυφον.

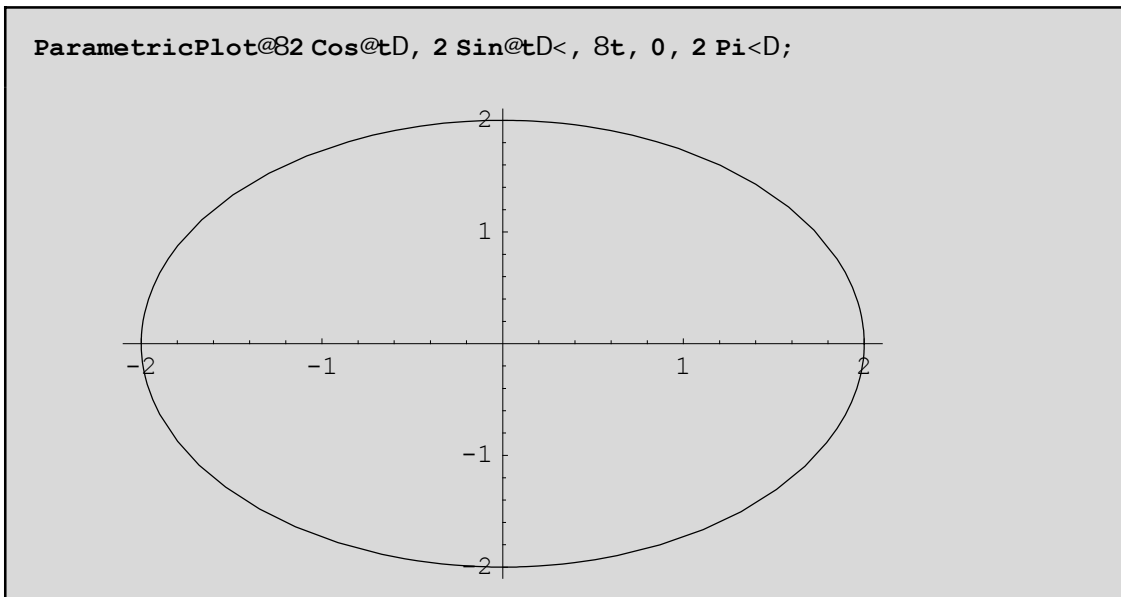
Αν ορίστε να "ζουμάρετε" σε κάποιο κομμάτι του γραφικού μπορείτε να σύρετε κατάλληλα τη V label πατώντας ταυτόχρονα το Ctrl.

Πατώντας πάνω στο γραμμή και στην συνέχεια το Ctrl (συνεχόμενα) και πάτημα κλικ στο γραμμή μπορούμε να δώσουμε σχηματίζεται ένα σταυρό και τη συντεταγμένη οποιαδήποτε σημείου (εκείνου που σχηματίζει το κέντρο του σταυρού) κάτω αριστερά στο παράθυρο.

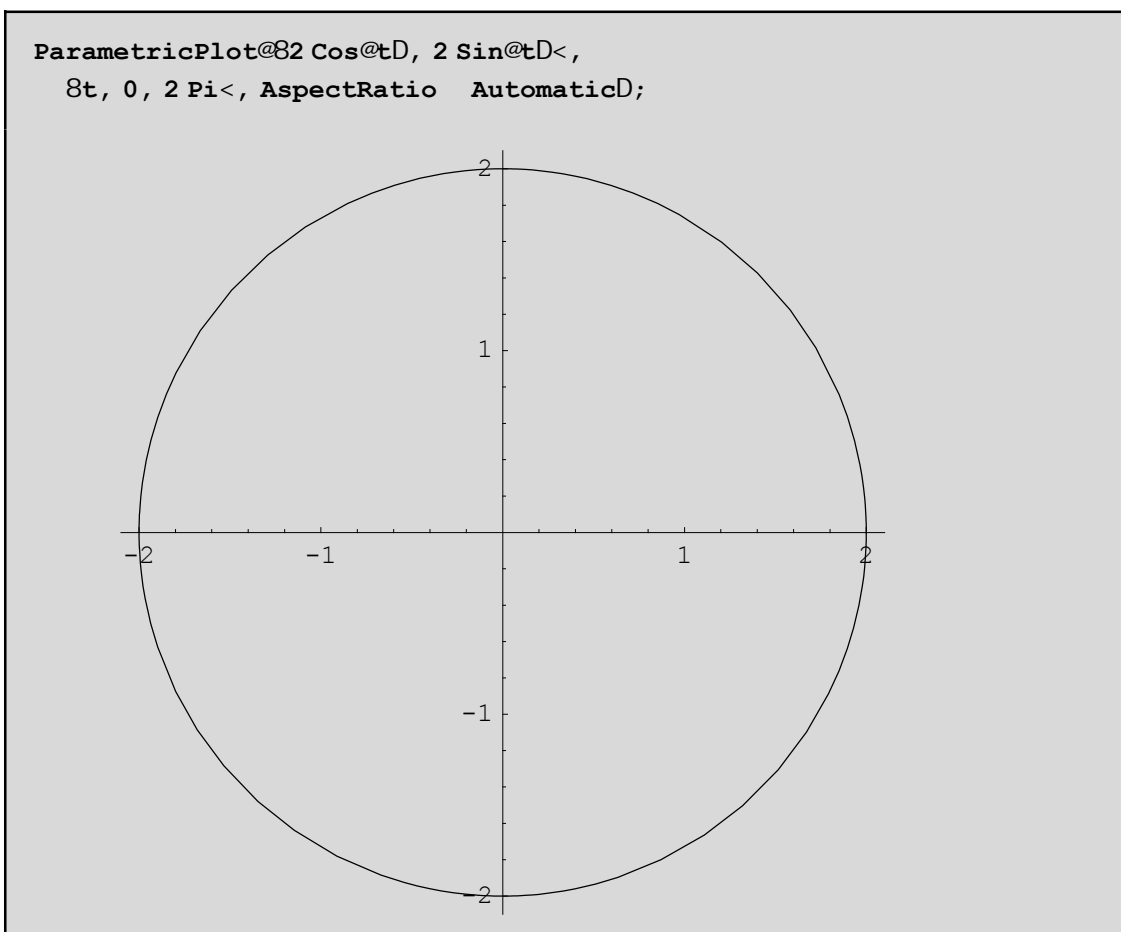
Η Plot παίρνει κάποια σημεία του x στο δοσμένο διάστημα (π.χ $\{x, -3 \text{ Pi}/2, 5 \text{ Pi}/2\}$) και χρησιμοποιώντας τα, υπολογίζει την συνάρτηση hV (εδώ $\text{Sin}[3 x]$) και μεταζεύγη (x, y) κάνει την γραφική παράσταση. Τα σημεία x που επιλέγονται λέγονται PlotPoints και είναι λίγα. Για αυτό υπάρχουν και οι ατέλειες στον σχεδιασμό της καμπύλης. Προσέξτε τη V ατέλειες για παράδειγμα στον σχεδιασμό της $\text{Sin}[3 x]$ όταν ενεργοποιήσετε τα V από τη V label. Όταν $\text{PlotPoints} > 40$ για παράδειγμα, μπορείται να έχετε ένα καλύτερο αποτέλεσμα.

Askhs: Μπορείτε περισσότερα για τη επιλογή Compiled, MaxBend, PlotPoints και PlotDivision και προσπαθήστε να τη εναρμόσετε στο σχεδιασμό της $\text{Sin}[3 x]$ με $\{x, 0, 4 \text{ Pi}\}$. Για να δείτε καλύτερα το πρόβλημα στο σχεδιασμό έχετε στην Plot το AspectRatio -> Automatic και κοιτάξτε την καμπύλη π.χ στα βαρύνοντα. Δεν είναι τόσο ομαλή όσο σε άλλα σημεία της. Για την AspectRatio που νίχστε ανήσων παρακάτω.

2. Όταν η καμπύλη δεν είναι γραμμή συνάρτησης (π.χ όταν η καμπύλη είναι ένα κύκλος) τότε δεν μπορούμε να χρησιμοποιήσουμε την Plot. Αν υποθέσουμε οπότε η καμπύλη ορίζεται με παραμετρικές εξισώσεις. Π.χ οι παραμετρικές εξισώσεις ενός κύκλου με ακτίνα r είναι $x=r \text{ Cos}[t]$, $y=r \text{ Sin}[t]$, με παράμετρο στο διάστημα $[0, 2 \text{ pi}]$. Σε τέτοιες περιπτώσεις μπορούμε τη Plot ή ParametricPlot



Scóliο: Το σcήμα που προκύπτει παύζει να ελλείψη και όci να κύκλσ! Αυτό vj ελ etai sto γεγονόv όti αυτόματα λαφρίζεται (απο την Plot) ο λόγος του ύψους του πλαισίου(που περιβάλλει το γράμμα) πρσ το πλάτος του πλαισίου να είναι 1 al 1/ Crusή Τομή dhl . περίπου 61.8034 % είναι μικρότερο το ύψος από το πλάτος! Αν όλ oune ενj ανίσoune το σcήμα όπωv πρagnatikά είναι den έcoune para na ές oune AspectRatio->1(υος διά πλάτος=1 dhl . τετράγωνο πλαίσιο) ή καλύτερα AspectRatio->Automatic (dhl . h νομόδα nάtrhs hV μήκουV στον Ox = νομόδα nάtrhs hV στον Oy):



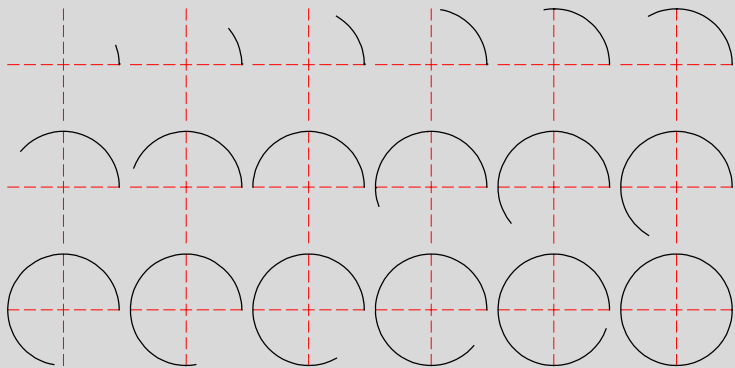
To `AspectRatio` `Automatic` είναι στιν περισσότερη περιπτώσειν αρκετά και ό διότι den paranorj óνει thn kanpól h. Se ne rikéV ónwW peritóséV ótan gia parádeigma ótan écoune negól é dus anal ogía tou mhkouv tou pedou tou x (pedio orisnó) proV to mhkov tou pedou tou y (pedio timón) crhsinopoióne nia sugkdrinmárh timí tou `AspectRatio` óstera écoune éna katarohtó gráj hna.

Askhs h Scediáste thn kanpól h $1/x$ gia $\{x,-1,1\}$ kai `AspectRatio`-> `Automatic`. Ti parathréte. All áxte to `AspectRatio` p.c se $1/2$ ή káti ál l oή aj air ésteto `AspectRatio` gia na bd tiós éteto gráj hna.

Scóli o: KaqóV to negal ónei apo 0 proV to 2 pól o kai negal útero nárov tou kókl ou scediázetai. Gia na dóne tiv diadocikéV "j áséV" nporóne na crhsinopóhs oune thn `entd` ή `Show` ses undas nó ne thn `entd` ή `GraphicsArray` p.c

```

akoloythia = Table@ParametricPlot@82 Cos@tD, 2 Sin@tD<, 8t, 0, n Pi<,
  DisplayFunction Identity, Ticks None, AspectRatio 1,
  AxesOrigin 80, 0<, PlotRange 88-2, 2<, 8-2, 2<<, AxesStyle
  8RGBColor@1, 0, 0D, Dashing@8.1, .05<D<D, 8n, 1 ê 9, 2, 1 ê 9<D;
kommatia = Partition@akoloythia, 6D;
Show@GraphicsArray@kommatiaDD;
    
```



H epil ogé DisplayFunction `Identity` aragkúzei thn `ParametricPlot` na ntn scediásei. H `akoloythia` perícei 18 `graj ikéV` parastáséV st h seirá kai h `Partition` kóbei thn `akoloythia` se `kommatia` twv 6 `graj hnatwn`. Opóte sta `kommatia` qa écoune 3 `grammáV` epi 6 `graj` h náta to kaqéra. H `GraphicsArray` topoqetá ta `graj` h náta tou píraka st h seirá (karé - karé) se ónoia pl aisia kai h `Show` enj anizéi td iká to apotél esna! Gia tiv epil ogéV pou upórcoun nása sto `ParametricPlot` ópwW gia parádeigma `PlotRange` `{{-2,2},{-2,2}}` écoune basikó skopó na kánoun pio katarohtá kai é kustiká ta `graj iká`. Me thn `PlotRange` l éne pciá shneia epiqunóne na enj anizontai st h eikóna. Me `PlotRange`-> `All` enj anizontai ó l a ta shneia thV kanpól hV.

Askhs h Máqete περισσότερα gia tiv parapánw epil ogéV thV `ParametricPlot` nesw tou `Help` kai pros paqís etena páranatis tétenē autéV.

Askhs h Scediásete thn $3-x^2$ gia $\{x,-5,5\}$ dial égotav diaj oretiká `PlotRange` kai `AspectRatio`. P.c qéste `PlotRange`->`All`, `PlotRange`-> $\{-1,1\}$,`All`, `PlotRange`-> $\{All,-1,.02\}$,`PlotRange`-> $\{-4,1\}$, `PlotRange`-> $\{2,10\}$,`{0,.02}` k o k kai diáj oreV epil ogéV gia `AspectRatio` p.c `Automatic`, $1/2$ k o k

3. H kanpól h dñretai wW gráj hna niaV sunúr ths hV, pou orízetai pep l egnána dh l . nás wexís ws hV. P.c gia na kánoune thn `graj iké` parástas h thV parabol ήV $x^2+3 y=5$ sto diást hna $[-3,3]$ qa prépei na l ús oune wW proV y ($y=\sqrt{5-x^2}/3$) kai netá na crhsinopóhs oune thn `Plot`. Kal útera ónwW gia qa h étan na kal és oune thn `entd` ή `ImplicitPlot` cw ríV na créas té na kánoune parapánw dou l éa:

```
ImplicitPlot@x^2 + 3 y ~ 5, 8x, -3, 3<D
```

```
ImplicitPlot@x^2 + 3 y == 5, 8x, -3, 3<D
```

DustucóV den naV ébgal e típota! AV dóne to lógo An patής oune F1 kai dós oune thn l éxh ImplicitPlot qa diapistós oune óti autή brísketai sto pakéto Graphics`ImplicitPlot`. To Graphics éna h diéoush (folder) nása sto opóio brísketai to pakéto ImplicitPlot. Mésa sto pakéto autó écei orisqa h sunárthsh ImplicitPlot. Pakéto l éne éna arcéo ne touV orisnoúV kápoiwv sunartής evn pou écouv skopó na károun nia édikή ergasia p.c na scediásoun nia sunárthsh ótan autή dñetai se pol ikév suntetagnév kok. Apo thn stignή pou kal óne éna pakéto den créazetai na to xarakal és oune parakátwgja dōterh j orá! Era pakéto to kal óne ne thn entolē Needs P.c Needs["Graphics`ImplicitPlot`"]. Pio apl á suníqW gráj oune to dipl ó << sthqishtou Needs p.c.

```
<< Graphics`ImplicitPlot`
```

```
- ImplicitPlot::shdw : Symbol ImplicitPlot appears in multiple
  contexts 8Graphics`ImplicitPlot`, Global`<; definitions in context
  Graphics`ImplicitPlot` may shadow or be shadowed by other definitions.
```

Proséxe anáswW naV ébgal éna mēruna pou l éei óti sunárthsh e thn l éxh ImplicitPlot se dōo diaj oretiká context sto Global kai sto Graphics`ImplicitPlot`. To context éna to ónona tou pakétou. To Global éna éna genikó pakéto kai anógei autónata kápe j orá pou anógoneto Mathematica. Éina "o córov upodochήV" se ópoion npaínei sto Mathematica. Gia parádeigma kápe j orá pou gráj oune nia l éxh (p.c ImplicitPlot) h éna súntod o niaV netabl htēV, apothkeúetai sto Global éktóV kai an enaív tou zhtής oune na to apothkeúsei se ál l o arcéo DustucóV h l éxh ImplicitPlot den écei orisqa sto Global ne kápio trópo. Den upárcei óte sto pakéto tou sustήnatoV dhl. to System (sto System brískontai di orisnoá twv pio basikōv sunartής evn ópW p.c thV prós qshV, thV Plot kok). Etsi to Mathematica qvré thn ImplicitPlot sankáti néo kai den écei idēa peri tínoV prókeitai h poia sunárthsh paristánē! Ótan kal és oune to pakéto <<Graphics`ImplicitPlot` tóte to Mathematica anakal úptē nása s' autó xaná th sunárthsh ImplicitPlot g' autó paraponiétai!!! H ImplicitPlot paranánei nia sunárthsh tou Global epéidh to Global episkíúzei to (eidikó) pakéto Graphics`ImplicitPlot` Etsi l dipón óte tóra qa naV doul éyē h ImplicitPlot paról o pou anixanetos vs to pakéto

```
ImplicitPlot@x^2 + 3 y ~ 5, 8x, -3, 3<D
```

```
ImplicitPlot@x^2 + 3 y == 5, 8x, -3, 3<D
```

Oa prēpē na diagráyoune thn `ImplicitPlot` apo to trécon arcéo (edó to Global) gráj ontaV:

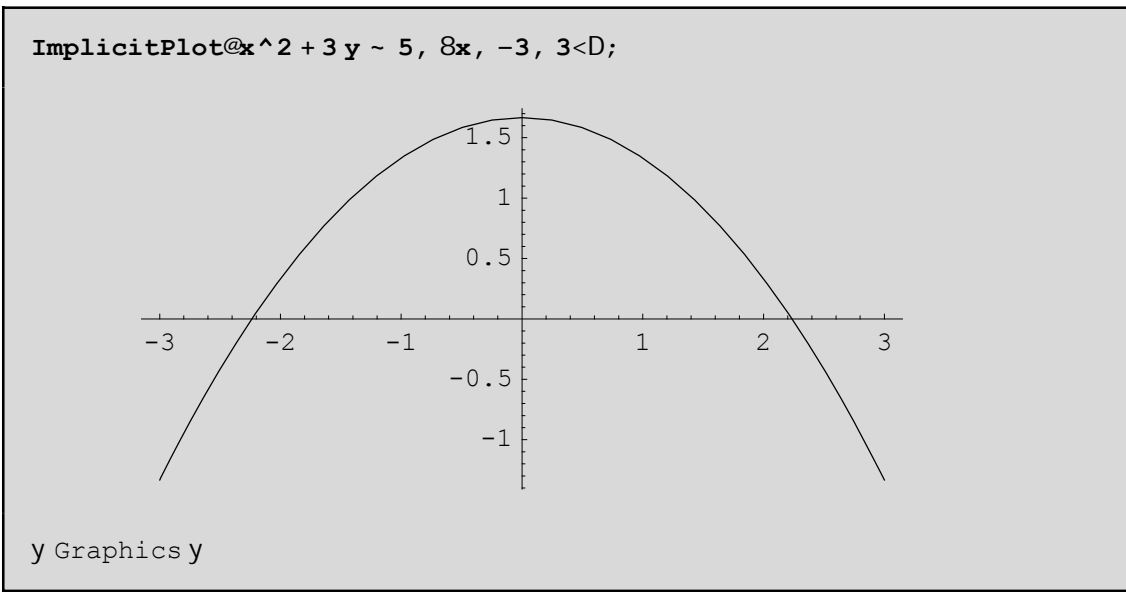
```
Remove@ImplicitPlotD
```

AV rwtής oune tóra poiev sunartής eív kai poia súntod a brískontai akóna nása sto Global

```
?Global`*
```

Global`
[akoloythia](#) [kommatia](#) [n](#) [t](#) [x](#) [y](#)

Όπως βλέπουμε δεν υπάρχει ποιά η `ImplicitPlot` στο `Global` οπότε ας την καλέσουμε ξανά.



Βλέπουμε λοιπόν ότι επειδή δεν βρήκε η `ImplicitPlot` στο `Global`, το *Mathematica* αναγκάστηκε να γυρίσει (με κάποια συγκεκριμένη σειρά - αυτήν που υπάρχει μέσα στο `$ContextPath`) στα άλλα πακέτα που ήδη έχουν ανοικτά δηλ. το `System` και το `Graphics`ImplicitPlot`` και γι' αυτό βρήκε στο δεύτερο πακέτο την σωστή συνάρτηση και την έγραψε.

Σκόλια: για να κερδίσουμε εντάξει δεν δημιουργείται να καλέσουμε κάποιο ειδικό πακέτο για να εκτελεστούν διότι βρίσκονται στο `System`. Εκδώ μέσα βρίσκονται όλες οι γνωστές συναρτήσεις του `Kernel` του *Mathematica*. Γράφοντας `$Packages` μπορούμε να δούμε ποιά πακέτα έχουν ανοικτά τώρα. Με την εντολή `Context[f]` παίρνουμε το όνομα του πακέτου που περιέχει την συνάρτηση `f` ενώ με `$Context` βρίσκουμε το `Context` στο οποίο βρίσκονται αυτή την στιγμή. Με `$ContextPath` παίρνουμε την σειρά προτεραιότητας που θα πρέπει να γυρίσει το *Mathematica* για να βρει τουλάχιστον των συνόλων και των συναρτήσεων. Αρκετά καλύτερα είναι εκείνο που αναζητήθηκε τελευταίο (από εντάξει με το `<<` ή `isw` και από το *Mathematica* κυρίως να το αντιληφθούμε) π.σ.

```
$Packages

8Utilities`FilterOptions`, Graphics`ImplicitPlot`, Global`, System`<
```

```
Context@PlotD
Context@PlusD
Context@ImplicitPlotD
$Context
$ContextPath

System`
```

```
System`
```

```
Graphics`ImplicitPlot`
```

```
Global`
```

```
8Graphics`ImplicitPlot`, Utilities`FilterOptions`, Global`, System`<
```

4. Η κληρονομιά δίνεται ως γραμμή εντολών, που ορίζεται σε πολυκείμενο συντεταγμένων. Το πακέτο Graphics`Graphics` πρέπει να ανοιχτεί κάπου ούρα που δημιουργήσατε την PolarPlot.

```
<< Graphics`Graphics`
PolarPlot@Cos@2 tD, 8t, 0, 2 Pi<, Background RGBColor@0, 0, .5D,
PlotStyle 8RGBColor@1, 0, 0D, Thickness@.01D<D;
```

Με PlotStyle λαμβάνουμε πάλι την εμφάνιση να σχεδιάσει η γραμμή παράσταση ενώ με Background RGBColor[0,0,.5] επιλέγουμε ένα βασικό μπλε για j όντο. Το R(=0) στη RGBColor είναι το κόκκινο το G(=0) το πράσινο και το B(=.5) είναι το μπλε. Αλλά παρατήρηση είναι ότι αυτόματα επιλέχθηκε ένα κίτρινο χρώμα για τον άξονα για λόγους αντίθεσης. Αν δεν να βρούμε το χρώμα αυτό, τότε με AxesStyle->..... μπορούμε να το αλλάξουμε.

Αν τώρα επιλέξουμε να δώσουμε τιμές σε j ως εσείς σε διάστημα 0 <= j <= 2π, τότε η κληρονομιά σε κληρονομιά θα γραφτεί

```
pinakas = Table@PolarPlot@Cos@2 tD, 8t, 0, n Pi<D, 8n, 0.1, 2, 0.1<D;
```

Αν κάπου διπλό κλικ πάρετε εντολή από τι παραπάνω γραμμή, παραστάσεις σε j θα δώσουμε τι j ως εσείς σε διάστημα 0 <= j <= 2π! Επειδή τα σχήματα δεν βγήκαν όμοια κριβώβερινάνα να κάπου κώπιδ/α/α γέ/ς του άξονα

```
Clear@pinakasD
pinakas = Table@PolarPlot@Cos@2 tD, 8t, 0, n Pi<,
PlotRange 88-1, 1<, 8-1, 1<<, Ticks NoneD, 8n, 0.1, 2, 0.1<D;
```

Το ίδιο αποτέλεσμα με το παραπάνω θα πάρουμε με την Do αντί της crήsh thV Table. Δοκιμάστε το παρακάτω

```
Do@PolarPlot@Cos@2 tD, 8t, 0, n Pi<,
  PlotRange 88-1, 1<, 8-1, 1<<D, 8n, 0.1, 2, 0.1<D
```

Αν νικράνουμε το βήμα 0.1 στο {n,0.1,2,0.1} ça έχουμε περισσότερα karé kai ára pio argή kίnhsh pc nporeτέρα antikatastήsete {n,0.1,2,0.01} sto Do kai na dkinήsete xará!

Askhsh: H parametrikή éxish thV proghónhV kanpól hV énai $x[t]=\text{Cos}[2 t] \text{Cos}[t]$ kai $y[t]=\text{Cos}[2 t] \text{Sint}[t]$. Pros paqiste na párete tiv diadocikéV j ás éV crhsinopíontaV thn ParametricPlot antí thV Polar-Plot kai tiv parametrikéV éxish éVanti tiv pol ikéV.

5. H kanpól h dínetai wV éna peperasnéno súnol o shnéwn data pou écouv prokýei apo netrής éV p.c qrnokrasíaV ή piéshV kok.p.c

```
data = Table@9x,  $\frac{1}{x}$ , 8x, 1, 10<E
synarthsh = Interpolation@dataD
Plot@synarthsh@xD, 8x, 1, 10<,
  PlotRange Automatic, AxesOrigin 80, 0<D;
```

```
981, 1<, 92,  $\frac{1}{2}$ , 93,  $\frac{1}{3}$ , 94,  $\frac{1}{4}$ , 95,  $\frac{1}{5}$ ,
96,  $\frac{1}{6}$ , 97,  $\frac{1}{7}$ , 98,  $\frac{1}{8}$ , 99,  $\frac{1}{9}$ , 910,  $\frac{1}{10}$ 
```

```
InterpolatingFunction@881, 10<<, <>D
```

H sunárrthsh Interpolation kúnei parentól ή sta shnéia twv data kai epistréj éi mia onal ή kanpól h pou pernai apota shnéia autá.

Mporóune na crhsinopíhsete kai thn ListPlot kai na enóscune ta shnéia twv data ne euqóγραμμα tmhnata al ía to apotél es na den énai tóso ikanopoihtikó:

```
ListPlot@data, PlotJoined True,
  PlotRange Automatic, AxesOrigin 80, 0<D;
```

H epil ogή PlotJoined@True anagkázé thn ListPlot na enónei ne euqóγραμμα tmhnata ta shnéia pou scedázé! Éinaí pd ó crhsinhs thn períptwsh pou ta data naV den paristáoun shnéia níaV sunárrthshV. P.c ne thn ListPlot nporóune na zwgraj isoune astéV j atsoól éV.

```
koryfes =
  880, 0<, 80.5, 1<, 80.7, 0.5<, 80.5, 0.3<, 80.6, 0.2<, 8.3, 0<<;
fatsoyla = ListPlot@koryfes, PlotJoined True, PlotRange Automatic,
  AxesOrigin 80, 0<, DisplayFunction IdentityD;
mati = Graphics@8PointSize@0.03D, RGBColor@1, 0, 0D,
  Point@80.54, 0.72<D<D;
Show@fatsoyla, mati, DisplayFunction $DisplayFunctionD
```

Με την Show μπορούμε να δούμε πολλά γραφήματα μαζί. Θα μιλήσουμε τώρα για αυτήν παρακάτω. Με DisplayFunction/Identity απούγανε να σχεδιάσουμε το περίγραμμα του κεφαλιού ενώ με DisplayFunction/DisplayFunction επαράγει την duration στο Show να εμφανιστεί της ατσόλα μαζί με το κόκκινο νάτι. Με την Graphics μπορούμε να σχεδιάσουμε σημεία, δίσκους, ευθύγραμμα τμήματα, πολύγωνα, προσέχουμε κέντρο και πολλά άλλα. Ένα κοίταγμα στο Help του Graphics θα σε απείσει...

Askhs: Crhs inopoióntaV thn sunárthsh Polygon gia tiv koryfes na zwgraj ísete xaná thn j atsoí a al l á n eto es wterikó tou ke j al iooú na éna i npl é (kai tonáti kókkino).

6. An h kanpól h dínetai w h sunárthsh pou ikanpoié kápoia diaj oriké éiswsh qa prépi próta na l úsoun ethndiaj oriké éiswsh P.c

```
eqn = y''@xD + 5 Log@y@xDD ~ 0
sol1 = NDSolve@{eqn, y'@D ~ 1, y@D ~ 1<, y@xD, 8x, 0, 4<D
Plot@y@xD ê. sol1, 8x, 0, 4<, PlotStyle RGBColor@1, 0, 0DD;

5 Log@y@xDD + y00@xD == 0
```

```
88y@xD InterpolatingFunction@880., 4.<<, <>D@xD<<
```

9.1.2 Scedízontavnia l ísta apokanpól ev

Με την εντολή Show μπορούμε να συνδυάσουμε γραφικές παραστάσεις που έχουν γίνει με διάφορα πακέτα:

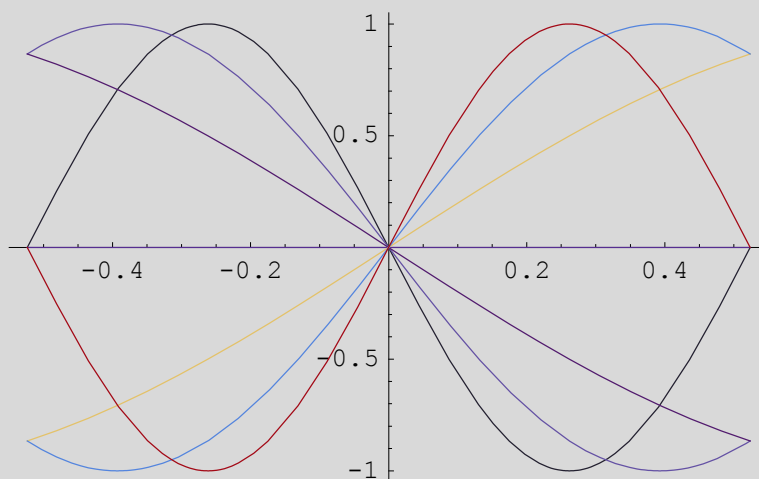
```
trigX@x_D := 88x, -0.03<, 8x, 0.03<, 8x + 0.03, 0<<
trigY@y_D := 88-0.03, y<, 80.03, y<, 80, y + 0.03<<
g1 = Graphics@{RGBColor@1, 0, 0D, Polygon@trigX@1.5DD,
  Polygon@trigY@1DD<, DisplayFunction IdentityD;
g2 = Plot@Sin@3 xD, 8x, -Pi ê 6, Pi ê 6<, DisplayFunction IdentityD;
g3 = PolarPlot@Cos@2 tD, 8t, 0, 2 Pi<, DisplayFunction IdentityD;
Show@g1, g2, g3, Axes True, PlotRange 88-1, 1.53<, 8-1, 1.03<<,
  DisplayFunction $DisplayFunctionD;
```

Με τη βοήθεια της Polygon της Graphics σχεδιάζουμε δύο τριγωνικά ένα στο σημείο x του άξονα Ox και ένα στο σημείο y του άξονα Oy αντίστοιχα. Οι συντεταγμένες του είναι trigX[x] και trigY[y] αντίστοιχα. Η Show συνδυάζει όλα αυτά μαζί και τις γραφικές παραστάσεις.

Αν βέβαια έχουμε γραφικές παραστάσεις που σχεδιάζονται με την ίδια εντολή τότε η χρήση δεν είναι ανάγκη να χρησιμοποιήσουμε την Show.

Ακόμα μπορούμε να παράξουμε τις γραφικές παραστάσεις των Sin[n x] για {n,-6,6,2} δηλ. για n=-6,-4,-2,0,2,4,6 με την Show και χωρίς να την


```
grafika = Table@Plot@Sin@n xD,
  8x, -Pi ê 6, Pi ê 6<, DisplayFunction Identity, PlotStyle
  RGBColor@Random@D, Random@D, Random@DDD, 8n, -6, 6, 2<D;
Show@grafika, DisplayFunction $DisplayFunctionD;
```



Δώσανε ένα τυχαίο χρόνο σε κάθε γραμμή παράσταση για να πάρουμε ένα αποτέλεσμα πιο ελκυστικό. Το `Random[]` να υπενθυμίσουμε επιστέφει ένα τυχαίο πραγματικό μεταξύ 0 και 1 οπότε με `RGBColor[Random[], Random[], Random[]]` δώσανε ένα εντελώς τυχαίο χρόνο στην κανονική ημίσημα `Sin[n x]`.

Αλλά ο τρόπος να γράψουμε την `Show`:

Πρώτα να μαζέψουμε τα ημίσημα σε ένα πίνακα και μετά να κάνουμε την γραμική παράσταση του πίνακα αυτό με την `Plot`:

```
Remove@pinakasD
pinakas = Table@Sin@n xD, 8n, -6, 6, 2<D
Head@pinakasD
style =
  Table@8RGBColor@Random@D, Random@D, Random@DD<, 8n, -6, 6, 2<D;
Plot@pinakas, 8x, -Pi ê 6, Pi ê 6<, PlotStyle styleD;

8-Sin@6 xD, -Sin@4 xD, -Sin@2 xD, 0, Sin@2 xD, Sin@4 xD, Sin@6 xD<
```

```
List
```

```
- Plot::plnr : pinakas is not a machine-size real number at x = -0.523599.
- Plot::plnr : pinakas is not a machine-size real number at x = -0.481117.
- Plot::plnr : pinakas is not a machine-size real number at x = -0.434787.
- General::stop :
  Further output of Plot::plnr will be suppressed during this calculation.
```

Βι έπouνε αυτό den απέδωσε! Ο Ι όγοV είναι ότι η Plot den ηπορέ να σcedιάσει μία λίστα απο sunartήs έV. Άρα qα πρέπει nekάποιο τρόπο να updlogίσoune πρώτα tiV sunartήs έV του pinakas gia ól éV/tiV tináV του {n,-6,6,2}, μία μία xecvrístá, και netá na anal ábei η Plot na zwgraj ísέi netηnsérá, káqenia ap' autέV. Αυτό akribóV kóni η Evaluate. Αl l ázei thnsérá twν entd ów(twn energéów). Parakátwkal ésane kai to paketo Graphics`Legend` gia tí qél oune na eis áqoune kai éra pinaka ne epethghnatikéV etikéteV.

```
<< Graphics`Legend`
etiketes = Table@ToString@Sin@n xDD, 8n, -6, 6, 2<D;
Plot@Evaluate@pinakasD, 8x, -Pi ê 6, Pi ê 6<, PlotStyle style,
PlotRange 88-0.6, 0.6<, 8-1, 1<<, PlotLegend etiketes,
LegendShadow 80, 0<, LegendSize 8.6, .5<,
LegendPosition 80.8, -.6<, LegendBackground GrayLevel@0.8DD;
```

Scólia:

a) Pros éxte nésa sto style qα πρέπει na upárcoun 7 akribóV perigráj éV ósέV kai oi sunartήs έV pou upárcoun sto pinakas. An upήrcenóno μία p.c PlotStyle{RGBColor[1,0,0]} τότε qα bgáinane ól éV/kókkineV! An upήrcan dío perigráj éV p.c PlotStyle{RGBColor[1,0,0],RGBColor[0,1,0]} τότε qα bgál éi tiV "nis éV" ne prásinokai tiV ál l éV nekókkino! Geniká ηπορέtera j tiáxeteto style ópw es éV qél ete P.c ne PlotStyle{RGBColor[1,0,0], Dashing[{.05,.02}], {GrayLevel[0.5]},{Hue[.03]},...} qα pároune thn próth kókkinh kai diakékonárh thn deúterh sunárthsh nía apócrwsh tou gkrízou, thn trítth nía apócrwsh tou Hue kok

b) Anden qél eteton pinaka diagráyte óti carakthristikó periécei thn l éxh Legend. An ómw s aV endiaj éré ti akribóV kóni τότε peiranatistéte ne ta diáj ora carakthristiká thV Legend kai páрте βοήqia apo to Help! Η entd ή ToString[Sin[n x]] epistréj éi thn l éxh "Sin[n x]". Ómw epéidi p.c Sin[-6 x] éinai iso ne -Sin[6 x] gia autó epistréj éi autó pou bl épete sth pinakída. Dokínáste ToString[Sin[ToString[n x]]] anden saVaréséi to paraparónw apotél es na.

Askhsh: Dinetai h él l éyh $x^2+5 y^2=9$, kai h kanpól h pou écei paranetrikή éxíswsh $x[t]:=Sin[t]$ kai $y[t]:=Sin[2 t]$ gia t sto diásthma $\{t,-4,4\}$. Na scedιάsete thn él l éyh se kókkino cróna ne thn ImplicitPlot, thn paranetrikή éxíswsh ne thn ParametricPlot kai ne diakekonnárh gramμή (p.c qéste PlotStyle->Dashing[{.1,.05}]) kai ne thn Show zwgraj íste autέV apo kaimó. Qés te epishV sth Show Background->GrayLevel[.4]

Askhsh: Prospaqéste na scedιάsete thn parágwgo (D[x Cos[x],x]) thV x Cos[x] gia x apo 0 nácri 10. Pros éxte na crhsinopáísete πρώτα thn Evaluate! Αl l ióV qα écete próbl hna. Η Plot den xéré pw na zwgraj ízei thn parágwgoníaVs unárthshV!

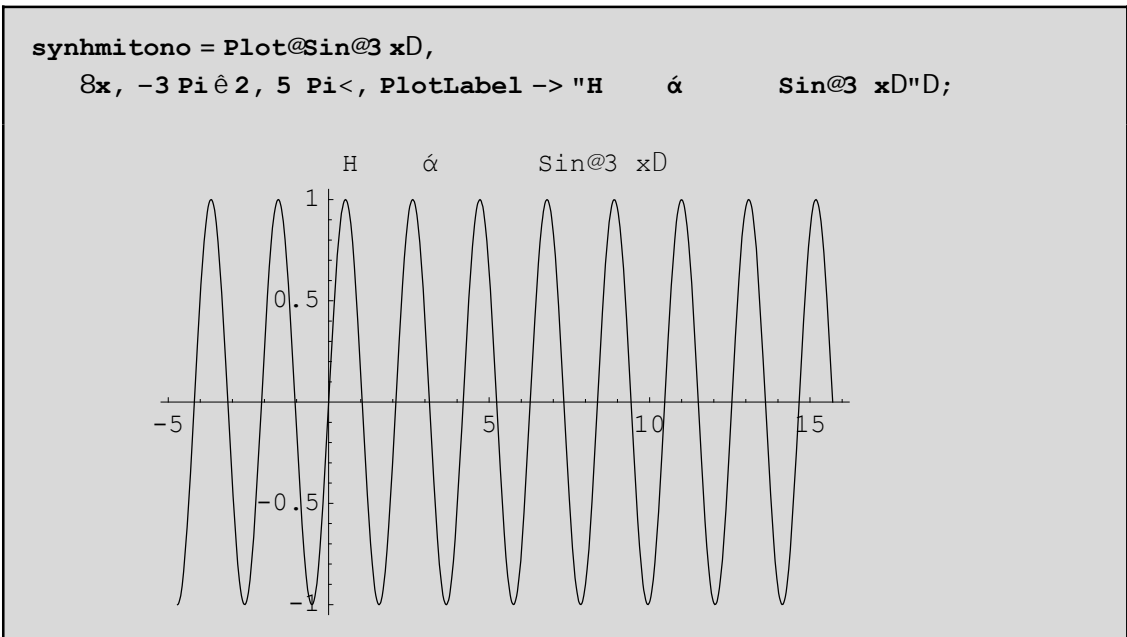
9.1.3 Oi epil ogéV twndi diástatwngraj ikón

Me Options[] ηπορόúne na dóne tiV epil ogéV níaV opias dēpotes unartήshV. P.c

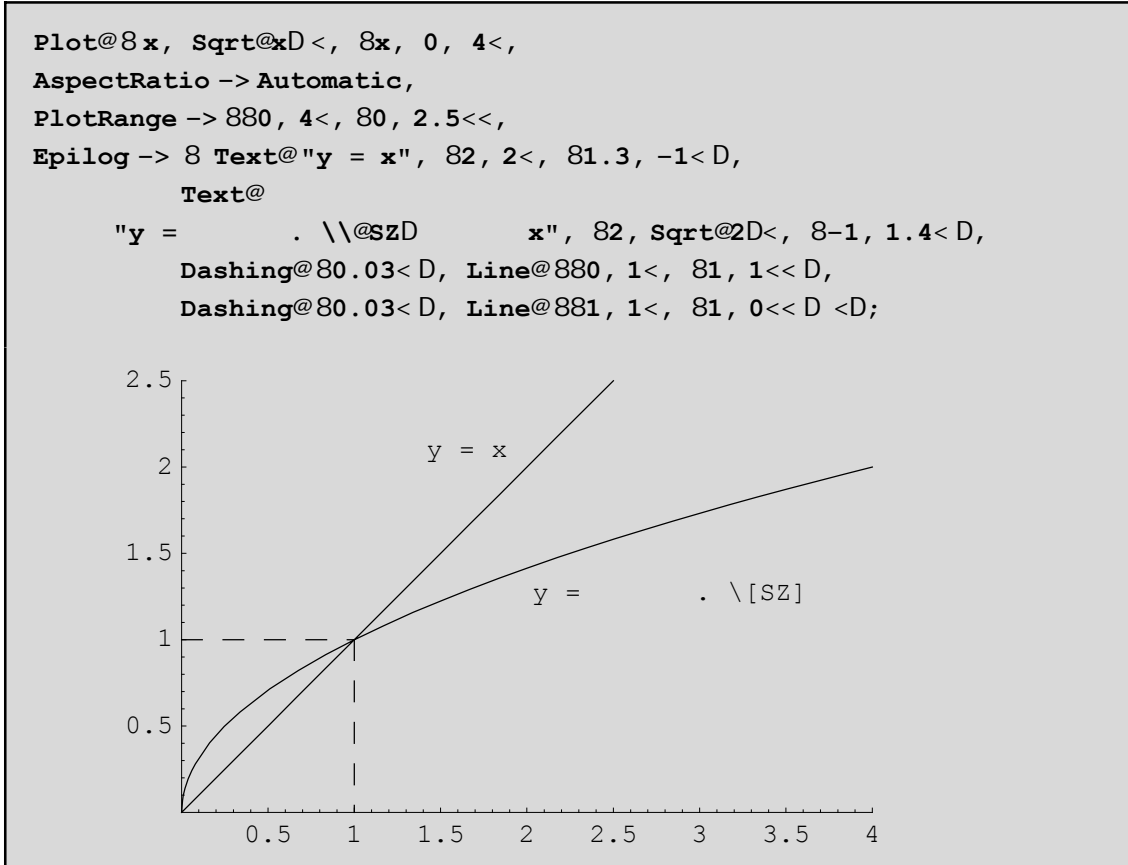
```
Options@PlotD
  AspectRatio  $\frac{1}{\text{GoldenRatio}}$ , Axes Automatic,
  AxesLabel None, AxesOrigin Automatic, AxesStyle Automatic,
  Background Automatic, ColorOutput Automatic, Compiled True,
  DefaultColor Automatic, Epilog  $\<$ , Frame False,
  FrameLabel None, FrameStyle Automatic, FrameTicks Automatic,
  GridLines None, ImageSize Automatic, MaxBend 10.,
  PlotDivision 30., PlotLabel None, PlotPoints 25,
  PlotRange Automatic, PlotRegion Automatic, PlotStyle Automatic,
  Prolog  $\<$ , RotateLabel True, Ticks Automatic,
  DefaultFont f $DefaultFont, DisplayFunction f $DisplayFunction,
  FormatType f $FormatType, TextStyle f $TextStyle=
```

Edw enjarizontai oi proepilognwv tinwv stie epiloggv. P.c h proep. timh thv AspectRatio einai AspectRatio $\frac{1}{\text{GoldenRatio}}$, twv axwnwv einai AxesAutomatic kok Creiazonaste ena biblio gia na analwsone dexodika kate nia apo tiv parapwnw epiloggv. Oa anaj erpwne ne suntonia se katepiv ap' autwv pou den ecoune da eww tora.

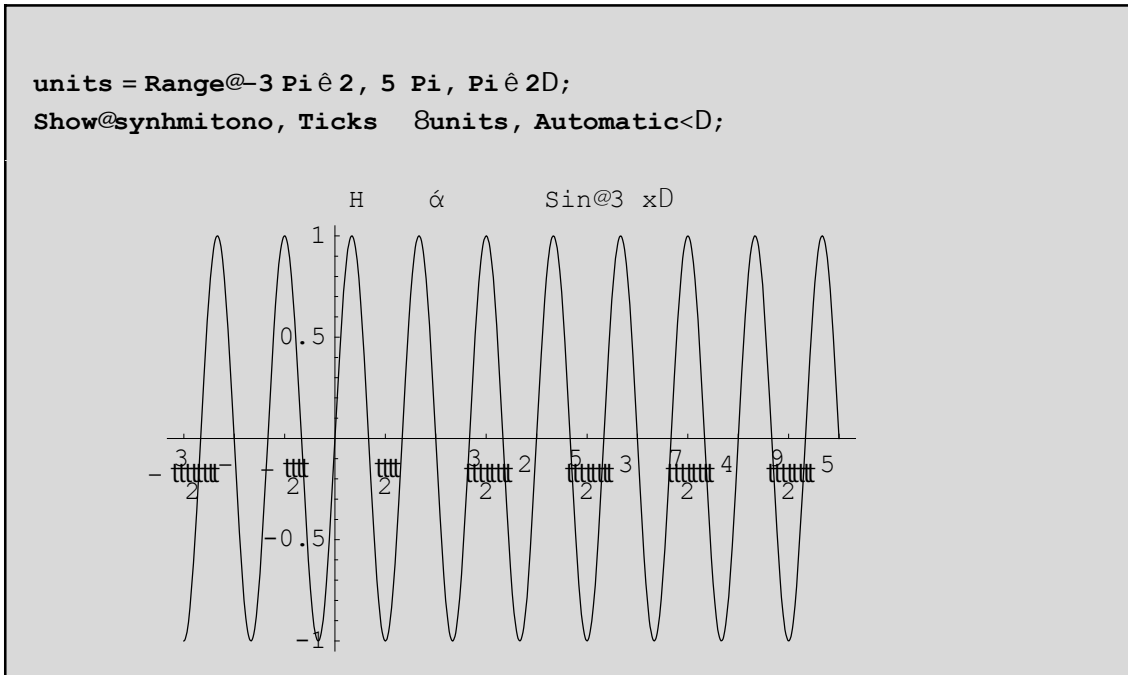
a) Epigraf ew(Labels): Mporwne na proswsone katepiva epigraf h ne thn PlotLabel



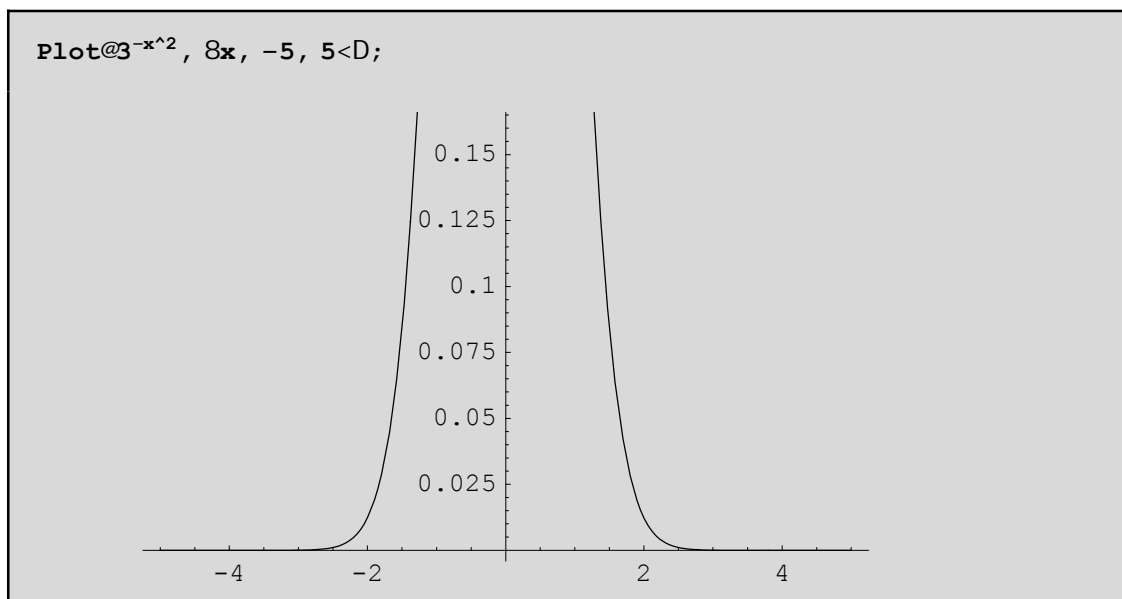
b) Epilog. Mporwne na crhsinwpihs one kai thn FrameLabel h thn AxesLabel (ligo parakwtw qa tiv doone perisswtero analutika) h thn Epilog gia ton idio skopo. H Epilog einai o "epiloggv" sto grafhnaV dhl. katepiva stoicwa (p.c kwnero, etikwta, epiplwron shnwta) pou qel one na npoun aj otou di okl hrwqá h grafh ikh parástash. Sto epwne no parádwna qwsane ta kwnera "y=x", "y=tetr.ríza tou x", kaqwV kai tiv diakekwnwneV proV touV axwneV apo to shnwio (1,1)



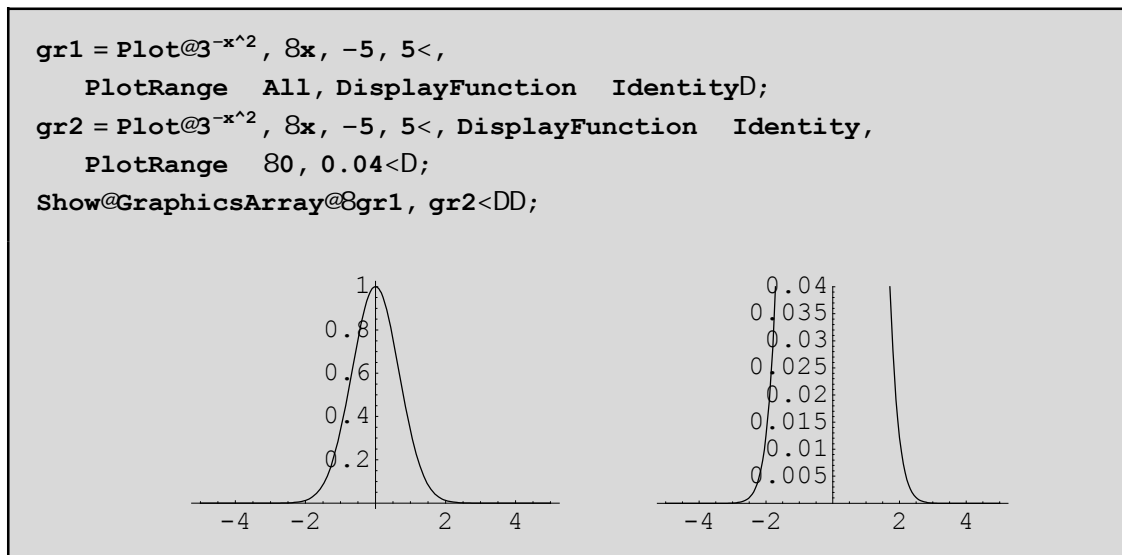
g) Ticks. Μπορούμε να βάλουμε Ticks στα σημεία που ενοφίουμε. Παράκτω βάζουμε στον άξονα Ox Ticks στα σημεία από $-3\pi/2$ έως $5\pi/2$ με βήμα $\pi/2$. Το `Ticks[0{units, Automatic}` σημαίνει ότι στον Oy θα ακολούθησε η προεπιλεγμένη ακολούθησα (η Automatic του Mathematica) των Ticks ενώ στον Ox κατά την διεκία να Vunits



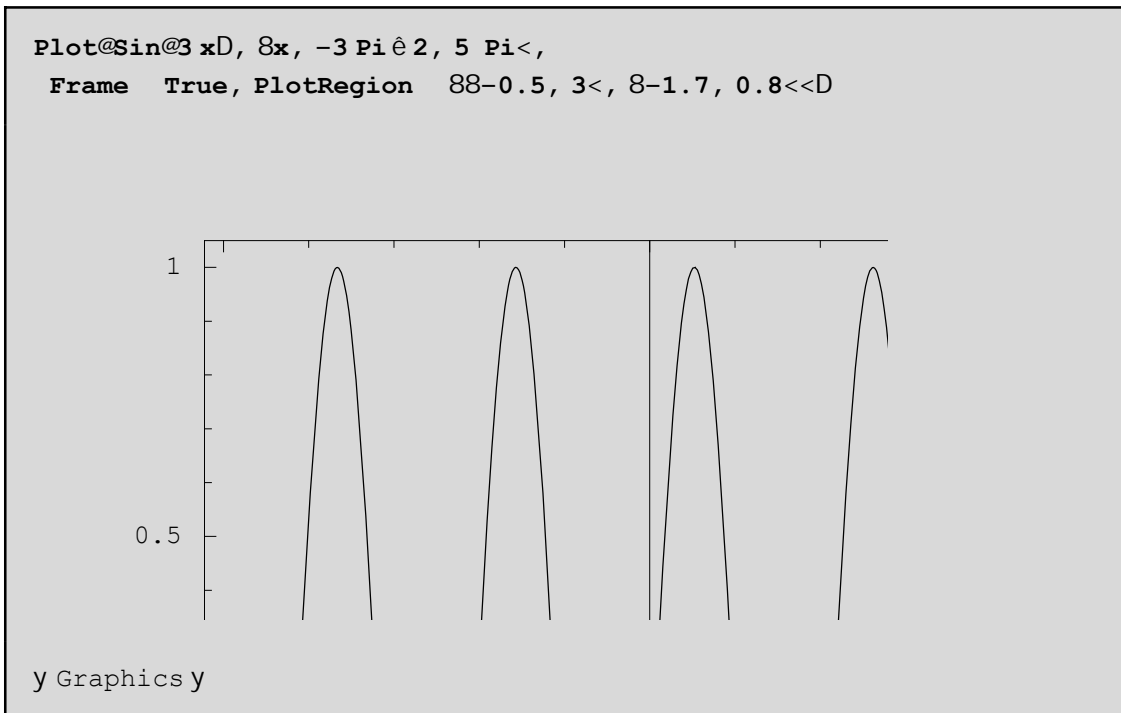
δ) **PlotRange**. Με **PlotRange**->**Automatic** καθορίζεται αυτόματα απο το *Mathematica* ποιές τιμές y της καμπύλης θα εμφανιστούν στο τελικό γράφημα. Σε μερικές περιπτώσεις αυτό δεν πετυχαίνει με αποτέλεσμα να χάνουμε σημαντικές πληροφορίες. Π.χ



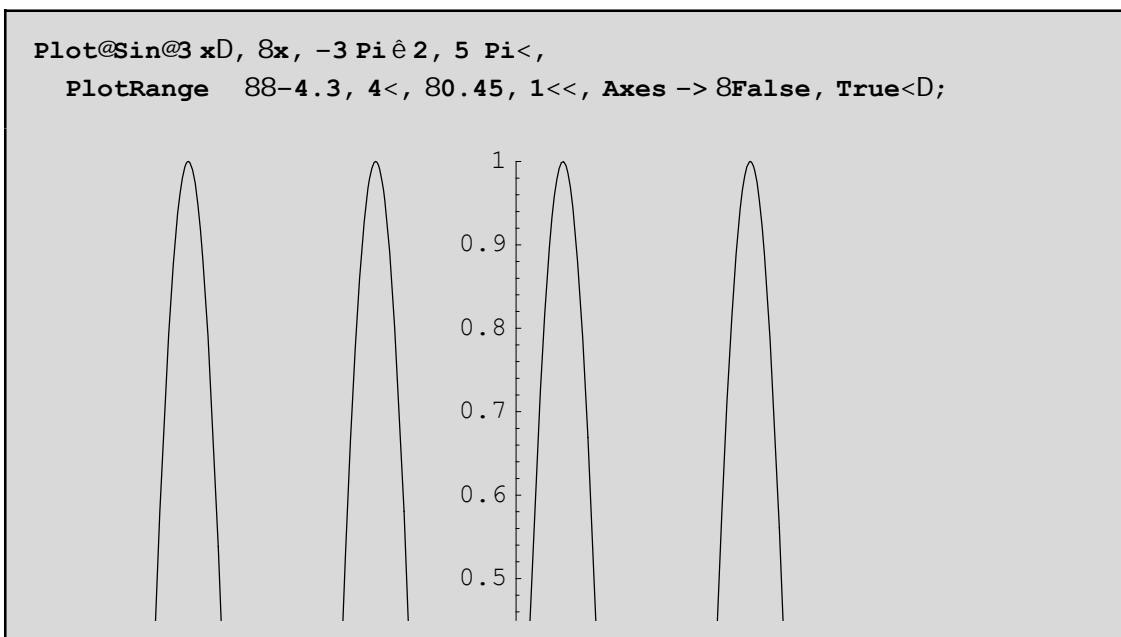
Το σχήμα έχει αποκοπεί στην γειτονιά του 0. Σε τέτοιες περιπτώσεις καλό είναι να βάζουμε **PlotRange**->**All** έτσι ώστε να εμφανίζονται όλες οι τιμές της καμπύλης. Αν πάλι δεν έχουμε το επιθυμητό αποτέλεσμα ή θέλουμε να εμφανίζονται κάποιες συγκεκριμένες τιμές του y , θα μπορούσαμε να βάλουμε ένα συγκεκριμένο διάστημα **PlotRange**->{ y_{min}, y_{max} } π.χ



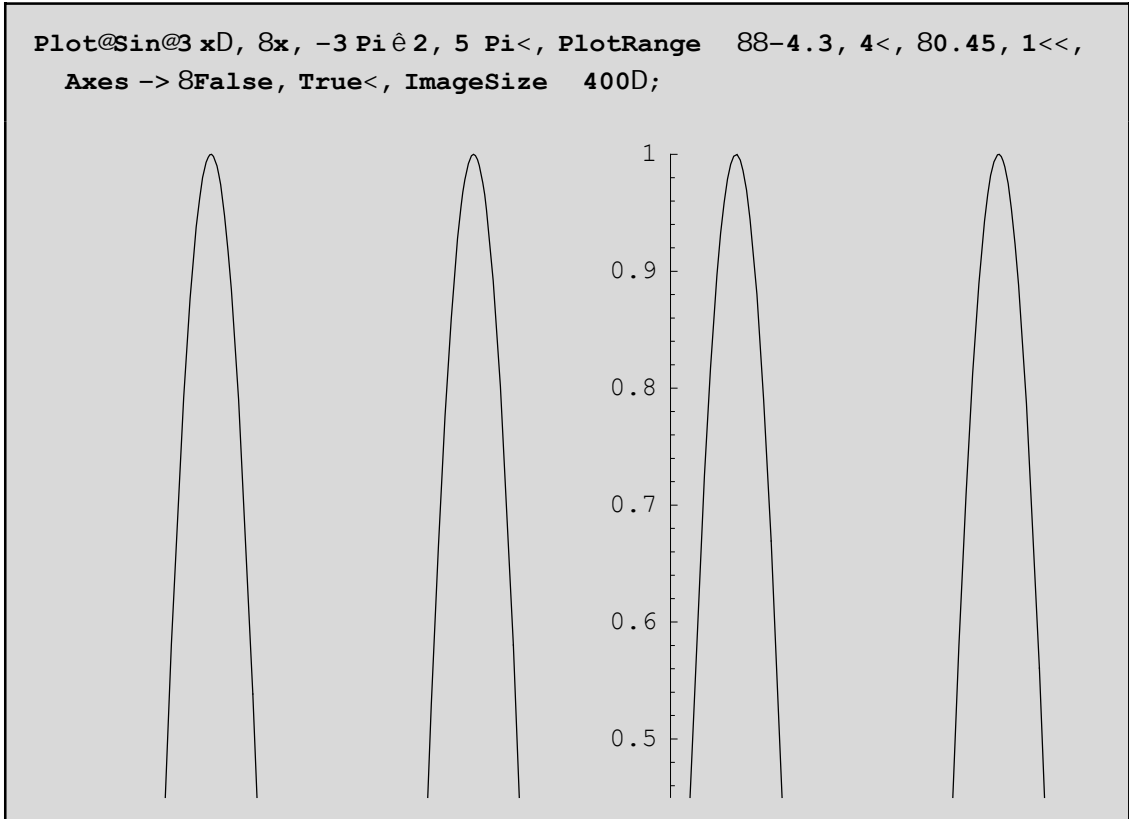
ε). Με **PlotRegion** μπορούμε να φέρουμε περιορία γύρω απο ένα γράφημα. Με κατάλληλη εντολή μπορούμε να στο **PlotRegion** να φέρουμε να ζουν ένα **ep.c**



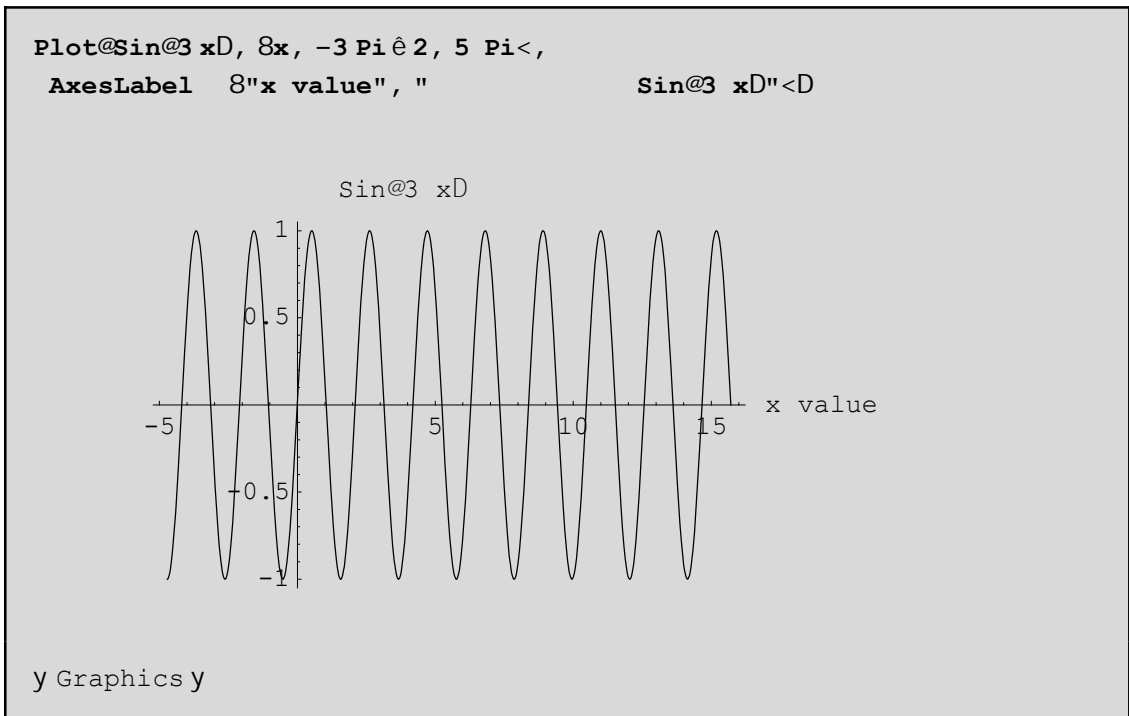
Askhsh Alláxte tiv tináV tou PlotRegion (protínsete tináV netaxó 0 kai 1 p.c Plot-Region $\{\{0.5,1\},\{0,0.5\}\}$ Dátetoapotél es na. Aj airésteto ólotoPlotRegion kai dátetoapotél es na! To $\{0.5,1\}$ sto PlotRegion $\{\{0.5,1\},\{0,0.5\}\}$ shnaínei óti to gráj hna qa katal ábei to diásthna 50% évW 100% sthorizóntia katéqunsh tou pl aisíou(dhl . to íhnis u dexió) enó to $\{0,0.5\}$ shnaínei óti to gráj hna qa katal ábei to diásthna 0% évW 50% sth káqeth katéqunsh tou pl aisíou(dhl . to kátw íhnis u) Mporóne kai na zounároune qétontaV pionikr ó PlotRange. To PlotRange écei scésh na káni neta shneia (x,y) thV kanpól hV, pou epiqunóne na enj arízetai sto gráj hna. To PlotRegion écei scésh nethn tel ikí qésh kai enj árish tou gráj ínatóV nása sto pl aisíou (pou qa enj arízetai sth oqírh naV ótan károune kl ik sto gráj hna)



Για να ζουνόρουνε ακόνα περισσότερο πορόνε να τραβήχουνε προV τα έxw tiV labéV ή να φέσουνε nia
 naóal h timí sto ImageSize p.c ImageSize 400



στ) ΆxωνéV. Me Axes->{False,True} αναγκάσane na nñn scedias té o άxωνάV Ox. Μπορόνε να βάλ cune
 etikéteV stoV άxωνéV

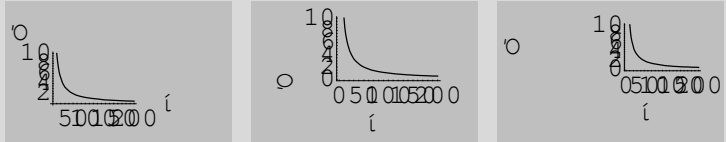


ζ) Frames και GridLines. Μπορούμε να βάλουμε πλαίσια γύρω από την καρδιά ή. Η βασική εντολή είναι Frame->True. Με FrameStyle, FrameLabel, FrameTicks RotatedLabel και GridLines μπορούμε να προσφέρουμε κάποια επιπλέον χαρακτηριστικά στο δικό μας πλαίσιο P.c

```

gr1 = Plot@100 Exp[-x], {x, 0, 2}, AxesLabel -> {"x", "y"},
      Background -> GrayLevel[0.5], DisplayFunction -> Identity;
gr2 = Show@gr1, Frame -> {True, True, False, False},
      FrameLabel -> {"x", "y"};
gr3 = Show@gr2, RotateLabel -> False;
Show@GraphicsArray@{gr1, gr2, gr3}

```



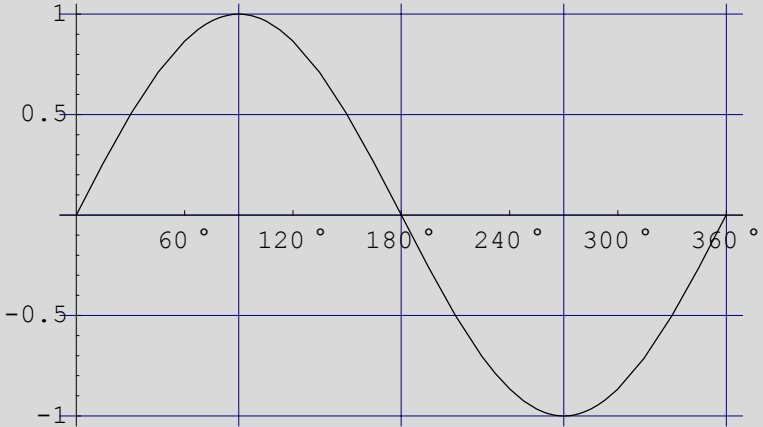
Null⁴

(Τραβήξτε προϋπόθεση από τη Vlab αν χρειαστεί να μεταφράσετε) Με Frame->{True,True,False,False} βάλουμε πλαίσιο μόνο αριστερά και κάτω. Με RotateLabel->False στρίβουμε την ετικέτα οριζόντια. Με RotateLabel->True. Με GridLines μπορούμε να προσφέρουμε πλέγμα στα σχέδια που φτιάχνουμε (με προεπιλογή GridLines->Automatic, το πλέγμα τοποθετείται αυτόματα από το Mathematica σε κάποιο εσωτερικό αλγόριθμο) p.c

```

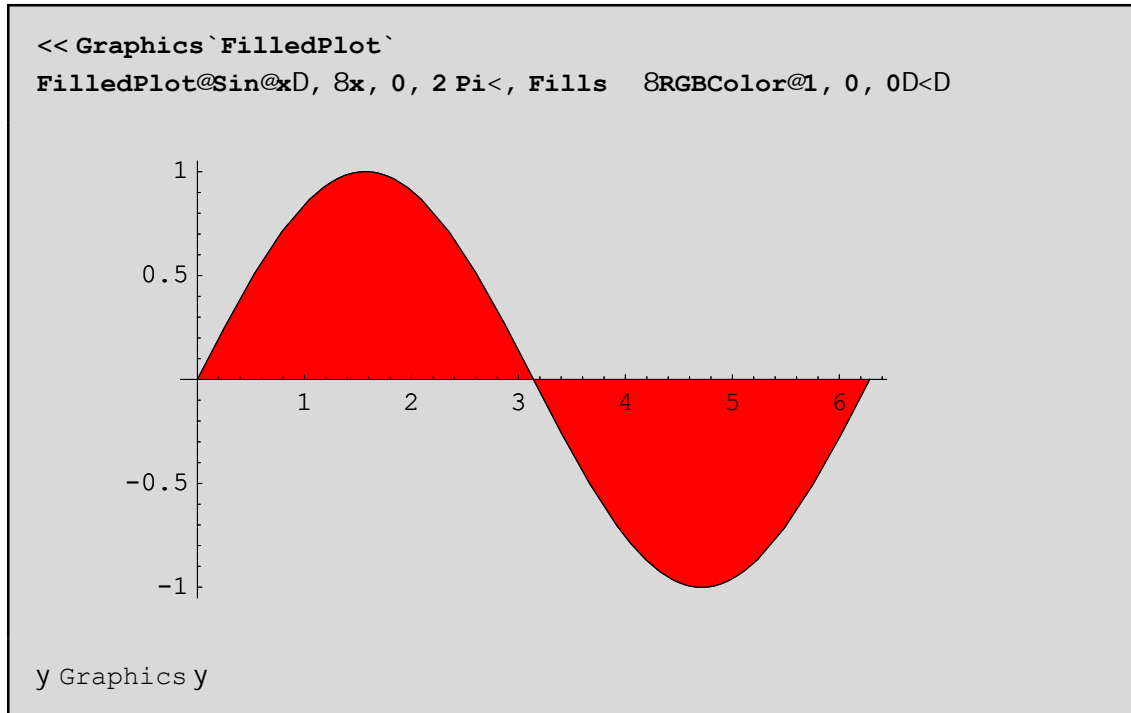
units = Range[0 Degree, 360 Degree, 60 Degree];
Plot[Sin[x], {x, 0, 2 Pi}, Ticks -> {units, Automatic},
      GridLines -> {{Pi/2, 3 Pi/2, 5 Pi/2}, Automatic}

```

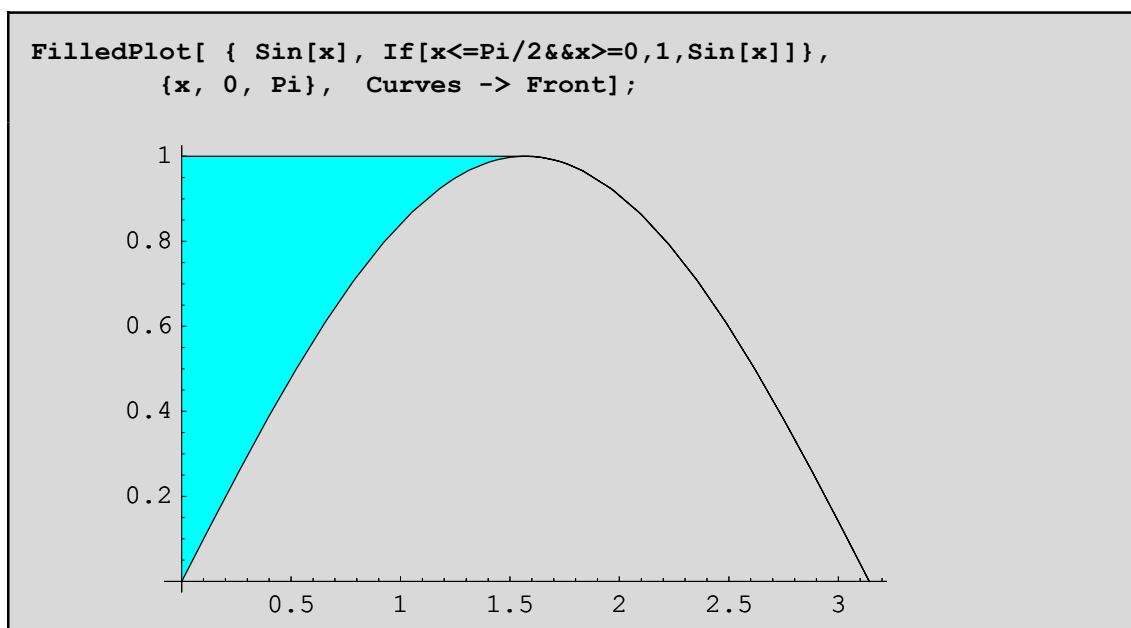


9.1.4 Άλλες δυνατότητες των διαδικαστικών γραμμάτων

Μπορούμε με `FilledPlot` του πακέτου `Graphics`FilledPlot`` να γεμίσουμε με ένα χρώμα το σώμα που βρίσκεται μεταξύ δύο ή παραπάνω καμπών ή καμπών ή μεταξύ μιας καμπής ή και ένα άξονα. Έτσι για παράδειγμα η `FilledPlot[f[x],g[x],{x,xin,xmax}]` γεμίζει με χρώμα τον χώρο μεταξύ της $f[x]$, και $g[x]$ για $\{x,xin,xmax\}$, ενώ με `sketo FilledPlot[f[x],{x,xin,xmax}]` ή `ne FilledPlot[f[x],0,{x,xin,xmax}]` γεμίζεται ο χώρος μεταξύ της $f[x]$ και του O_x . P.c



Στο επόμενο παράδειγμα βλέπουμε χρώμα μεταξύ της $f[x]=\sin[x]$ και της ευθείας $g[x]=1$ και μόνο στο διάστημα $[0,\pi/2]$. Αναγκαστήκαμε να ορίσουμε την $g[x]=1$ για το διάστημα $[0,\pi/2]$ και $f[x]$ στο υπόλοιπο διάστημα έτσι ώστε να βγεί το επιθυμητό αποτέλεσμα



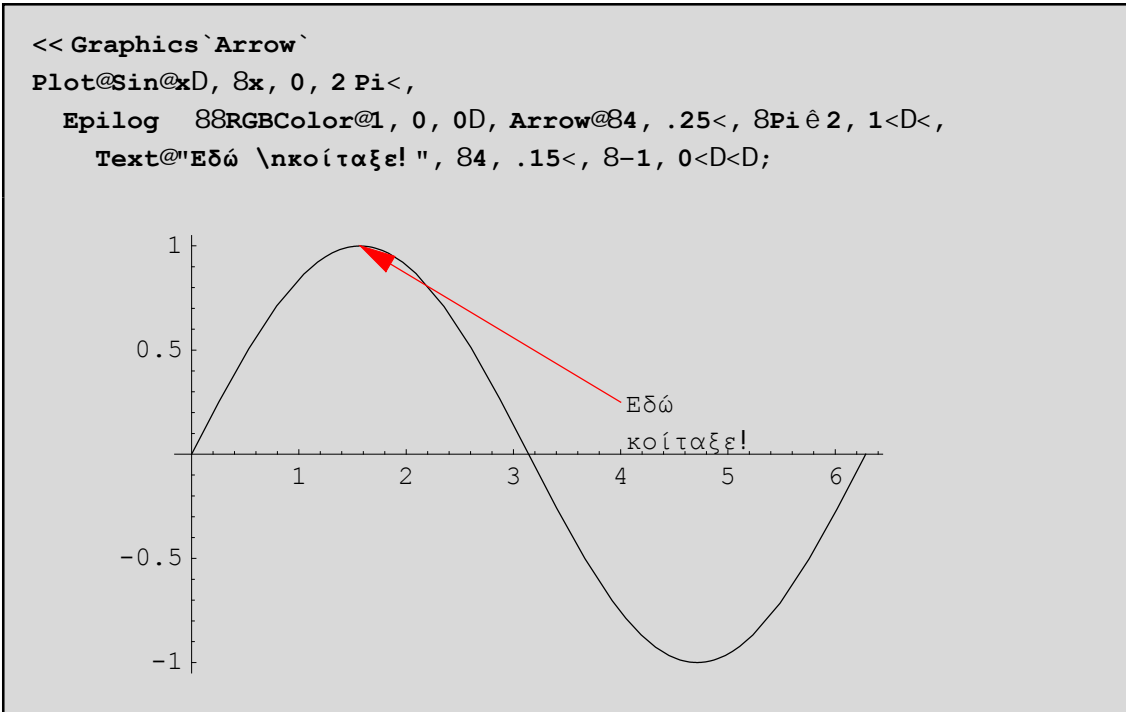
Με `PlotVectorField` του πακέτου `Graphics`Plotfield`` παίρνουμε την γραφική παράσταση ενός διανυσματικού πεδίου ενώ με `CartesianMap` και `PolarMap` του πακέτου `Graphics`ComplexMap``, σχεδιάζουμε ημιγάδικες συναρτήσεις. Με `PlotGradientField[f[x,y],{x,xmin,xmax},{y,ymin,ymax}]` του πακέτου `Graphics`PlotField`` παίρνουμε την κατεύθυνση κατά την οποία η συνάρτηση $f[x,y]$ αυξάνεται με τον ταχύτερο ρυθμό από το σημείο $\{x,y\}$. Δεν πρέπει να ξεχάσουμε και την `Animate` του πακέτου `Graphics`Animation`` που μας δίνει την δυνατότητα της κινούμενης εικόνας. Λίστες σημείων ή λίστα συναρτήσεων. Παράδειγμα `ListPlot`, `Arrow` <<`Graphics`Arrow``,

```
<< Graphics`Animation`
<< Graphics`Graphics`

Animate[ListPlot[Table[A  $\frac{1}{n} + \sin A \frac{n}{2} E + \cos A \frac{n}{2} E$ , {n, m, E}],
  PlotRange -> {{80, 20}, {8-1.5, 1.5}}, AxesOrigin -> {80, 0},
  PlotStyle -> {PointSize->{0.02D, Hue->{.6D<E, 8m, 1, 20, 1<E
```

Πατώντας διπλό κλικ σε ένα από τα παραπάνω γραφήματα έχουμε την ζητούμενη Animation. Προσέξτε και έσane δύο διαδοχικά πακέτα για δύο διαδοχικούς σκοπούς. Και έσane το `Graphics`Graphics`` για την `ListPlot` και το `<<Graphics`Animation`` για το `Animate`. Από την κατασκευή έβρανε ότι η ακολουθία $\frac{1}{n} + \sin A \frac{n}{2} E + \cos A \frac{n}{2} E$ δέce δύο συγkλίνουσες/υπακλούςες.

Τελος με `Arrow[{x0,y0},{x1,y1}]` μπορούμε να σχεδιάσουμε ένα βέλος από το σημείο $A_0 Hx0, y0L$ στο $A_1 Hx1, y1L$. Είναι χρήσιμο όταν θέλουμε να σχεδιάσουμε βέλη ή προσανατολισμένους άξονες ή όταν θέλουμε να τονίσουμε ένα τμήμα του γραφήματος. Η `Arrow` απαιτεί το πακέτο `Graphics`Arrow`` π.χ με την βοήθεια



Με το `\n` μπορούμε να αλλάξουμε γραμμή στο κείμενό μας!