

Asymptote: The Vector Graphics Language

Huỳnh Văn Thơ

Ngày 7 tháng 5 năm 2020

Mục lục

1. Điểm – pair	2
1.1. Khai báo điểm	2
1.2. Vẽ điểm – nhãn điểm	3
1.2.1. Vẽ điểm	3
1.2.2. Điều khiển nhãn	3
2. Đường dẫn – path	5
2.1. Path	5
2.1.1. Path đơn vị	6
2.2. Path controls	7
2.2.1. Một điểm điều khiển	7
2.2.2. Hai điểm điều khiển	7
2.2.3. Đa giác đều n cạnh	7
2.2.4. Hình chữ nhật	8
2.2.5. Đường tròn – cung tròn	8
2.2.6. Một số ví dụ khác về path	9
2.3. Kiểu đường dẫn	10
2.4. Kiểu nối và đầu mút của path	10
2.5. Tạo path kín từ các path	11
2.6. Giao điểm – điểm trên path	11
2.7. Path con – subpath	12
2.8. Điểm trên path	12
2.9. Tiếp tuyến tại điểm thuộc path	14
3. Vẽ	14
3.1. Mũi tên – Arrow – Bar	15
3.2. Nhãn	16
3.2.1. Nhãn cho điểm	16
3.2.2. Nhãn cho path	17
3.3. Ký hiệu cho path	20
3.3.1. Kiểu StickIntervalMarker	20
3.3.2. Kiểu CrossIntervalMarker	21
3.3.3. Kiểu CircleBarIntervalMarker	22
3.3.4. Kiểu TildeIntervalMarker	23

3.4. Vẽ góc	24
3.4.1. Cách tạo các marker để đánh dấu góc	26
4. Màu – tô màu	35
4.1. Màu sắc	35
4.2. Tô màu	36
4.2.1. Đơn màu	36
4.2.2. Nhiều màu	37
4.2.3. Màu kiểu tịnh tiến	37
4.2.4. Màu kiểu trung tâm	38
4.2.5. Tô màu miền giao nhau - evenodd	38
4.2.6. Tô màu cho text	40
5. Phép biến hình	41
6. Pen – font chữ	41
6.1. Pen	41
6.2. Font chữ	42
7. Nền – patterns	48
7.1. Kiểu tile – lưới	48
7.2. Kiểu checker – ô caro	48
7.3. Kiểu brick – lát gạch	49
7.4. Kiểu hatch – gạch sọc	50
7.5. Kiểu crosshatch – 2 đường chéo	50

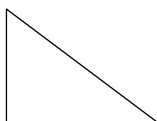
1. Điểm – pair

1.1. Khai báo điểm

Điểm trong Asymptote được định nghĩa theo kiểu dữ liệu **pair** bởi cặp (x,y) trong mặt phẳng tọa độ Oxy .

Code 1: Khai báo điểm \rightarrow chỉ định tọa độ

```
unitsize(1cm);
pair A, B, C, D; // Khai báo các điểm
A=(0,0); // Chỉ định tọa độ
B=(2,0);
C=(4,0);
D=(4,2); //
draw(A--B); // nối A với B
draw(C--D); // nối C với D
```



Code 2: Khai báo điểm và chỉ định tọa độ

```
unitsize(1cm);
pair A=(0,0), B=(2,0), C=(0,1.5);
draw(A--B--C--cycle); // Nối các điểm --> Tam giác ABC
```

Code 3: Khai báo điểm theo mảng

```
unitsize(1cm);
pair [] A;
A[0]=(-1,-1);
A[1]=(1,-1);
A[2]=(1,1);
A[3]=(-1,1);
//--- Nối các điểm được hình vuông
draw (A[0]--A[1]--A[2]--A[3]--cycle);
```



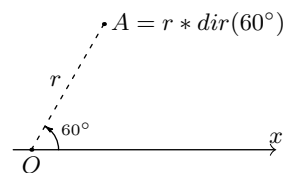
Code 4: Khai báo điểm theo list

```
unitsize(1cm);
pair [] M = {(0,0), (3,0), (3,1.6)};
dot(M); // vẽ các điểm trong list M
// M[0]=(0,0), M[1]=(3,0), M[2]=(3,2)
```

Ngoài ra, Mando cũng có thể khai báo điểm theo tọa độ cực.

Code 5: Định nghĩa điểm theo tọa độ cực

```
unitsize(1cm); dotfactor =3;
defaultpen(fontsize(8pt));
pen p = linetype("3 6");
real r=1.6; pair O =0;
//--- điểm A = (r*cos(60), r*sin(60));
pair A = r*dir(60);
dot("$O$",O,S);
draw(Label("$x$",align=N,EndPoint),(-0.2,0)--(2.7,0),
      Arrow(TeXHead));
dot("$A=r*dir(60^\circ)$",A,E);
draw("\tiny $60^\circ$",arc(O, 0.3,0,60),Arrow(HookHead(
      barb=1.2),size=0.6mm));
draw(Label("$r$",align=LeftSide,Relative(.5)),O--A,p);
```



1.2. Vẽ điểm – nhãn điểm

1.2.1. Vẽ điểm

Để vẽ điểm và gắn nhãn cho điểm, ta thường dùng 2 cú pháp sau

```
void dot(
    picture pic=currentpicture,
    pair z, // diem z hay cap (x,y)
    pen p=currentpen, // mau diem
    filltype filltype=Fill //to
        mau
    // Chi ve diem, mau diem va to
    diem
);
```

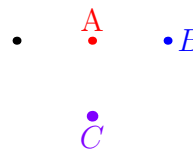
```
void dot(
    picture pic=currentpicture,
    Label L, // Dat trong dau " ", hoac tao kieu
        nhan Label
    pair z, // diem z
    align align=NoAlign, // Huong nhan
    string format=defaultformat, //Mau nhan=mau
        diem
    pen p=currentpen, filltype filltype=Fill );
```

Gắn nhãn cho một điểm ta dùng chuỗi ký tự **string**, Asymptote dùng kiểu dữ liệu như ngôn ngữ lập trình C++. Tức là phải để chuỗi trong dấu nháy kép " ", cũng có thể đặt chuỗi trong "\$ \$".

Vị trí đặt nhãn **align** =... gồm: **N**, **S**, **W**, **E**, **NE**, **NW**, **SE**, **SW** và **LeftSide**, **RightSide**, **Center** tương ứng Bắc, Nam, Tây, Đông, Đông Bắc, Tây Bắc, Đông Nam, Tây Nam, bên trái, bên phải, chính giữa.

Code 6: Vẽ điểm có nhãn – không nhãn

```
unitsize(1cm);
dot(0); // Ve diem O =(0,0) khong co nhan
dot("A", (1,0), N, red); // Diem (1,0) co nha la A, mau do
dot("$B$", (2,0), E, blue); // Diem (2,0) co nha la B, mau xanh
dot("$C$", (1,-1), S, 4bp+purple); //Diem (1,-1) co nha la C
```



1.2.2. Điều khiển nhãn

Để gắn nhãn tại 1 điểm, cũng như chỉ định các thuộc tính của nhãn tại một điểm đã chỉ định, Mando thường dùng lệnh sau.

```
void label(
    picture pic=currentpicture, //-- Anh hien hanh
    Label L, // -- 4 kieu Label liet ke o duoi
    pair position, //-- Diem gan nhan
    align align=NoAlign, //-- Huong nhan
    pen p=currentpen, // -- Mau nhan,
    filltype filltype=NoFill //-- To mau cho nhan
);
```

4 kiểu nhãn và các tùy chọn của nhãn

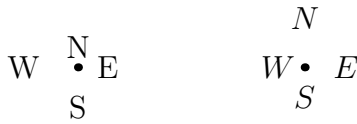
```
Label Label(
    string s="", // nhan dat trong dau " "
    pair position, // Vi tri dat nhan: BeginPoint=Relative(0); MidPoint=Relative(0.5);
    align align=NoAlign, // Huong nhan
    pen p=nullpen,
    embed embed=Rotate, // quay nhan
    filltype filltype=NoFill //-- To nhan
);
```

```
Label Label(string s="", align align=NoAlign, pen p=nullpen,
    embed embed=Rotate, filltype filltype=NoFill);
```

```
Label Label(Label L, pair position, align align=NoAlign,
    pen p=nullpen, embed embed=L.embed, filltype filltype=NoFill);
```

```
Label Label(Label L, align align=NoAlign, pen p=nullpen, embed
            embed=L.embed, filltype filltype=NoFill);
```

Code 7: Dùng lệnh dir để đặt vị trí nhãn



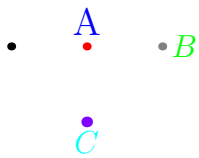
```
unitsize(3cm);
// ---
dot((0,0));
label("N", (0,0), N);
label("E", (0,0), 2E);
label("S", (0,0), 3S);
label("W", (0,0), 4W);
// ---
dot((1,0));
label("$N$", (1,0), 4dir(90));
label("$E$", (1,0), 3*dir(0));
label("$S$", (1,0), 2*dir(-90));
label("$W$", (1,0), dir(180));
```

Code 8: Các vị trí nhãn thường dùng

```
unitsize(3cm);
//-----
dot((0,0)); // Ve diem O=(0,0)
label("N", (0,0), N); // Nhan o phia Bac
label("E", (0,0), E); // Nhan o phia Dong
label("S", (0,0), S); // Nhan o phia Nam
label("W", (0,0), W); // Nhan o phia Tay
//-----
dot((1,0)); // Ve diem A=(1,0)
label("NE", (1,0), NE); // Nhan o phia Dong Bac
label("SE", (1,0), SE); // Nhan o phia Dong Nam
label("SW", (1,0), SW); // Nhan o phia Tay Nam
label("NW", (1,0), NW); // Nhan o phia Tay Bac
```



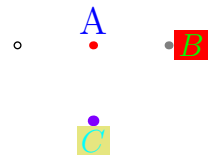
Code 9: Tô màu cho nhãn



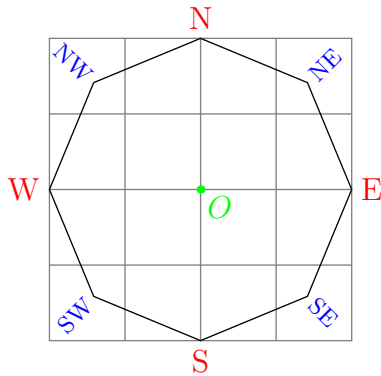
```
unitsize(1cm);
dot(0);
dot(Label("A", fontsize(14pt)+blue), (1,0), N, red); //size A la 14pt, mau
xanh
dot(Label("$B$", green), (2,0), dir(60), gray); // Diem B o vi tri 60 do
dot(Label("$C$", cyan), (1,-1), 2S, 4bp+purple); // do day diem 4bp
```

Code 10: Tô khung cho nhãn – tô điểm

```
unitsize(1cm);
dot(0, Fill(white)); // To mau white cho diem O=(0,0).
dot(Label("A", fontsize(14pt)+blue), (1,0), N, red);
dot(Label("$B$", green, Fill(red)), (2,0), E, gray);
dot(Label("$C$", cyan, Fill(gray+0.4*yellow)), (1,-0.8), S, 4bp+purple);
```



Code 11: Quay nhãn



```
import math; // de tao luoi
unitsize(0.8cm);
add(shift(-2,-2)*grid(4,4,gray)); // Tao luoi
pair O=(0,0); dot("$O$", (0,0), SE, green); //
real r=2; // khoang cach tu tam den dinh cua 8 giac deu
draw(rotate(22.5)*scale(r)*polygon(8)); // 8 giac deu
pen pa=fontsize(12pt)+1bp+red; // dinh nghia kieu but
pen pb=fontsize(9pt)+blue; //
//-----
label("N", O+r*N, N, pa); // O+r*N = (0,0)+2*(0,1)=(0,2)
label("E", O+r*E, E, pa); //
label("S", O+r*S, S, pa); // O+r*S = (0,0)+2*(0,-1)=(0,-2)
label("W", O+r*W, W, pa);
//----- Quay nhan --> rotate(angle)
label(rotate(45)*"NE", O+r*NE, NE, pb); //O+r*NE=(0,0)+2*(can 2/2,
label(rotate(-45)*"NW", O+r*NW, NW, pb); // can 2/2)
label(rotate(45)*"SW", O+r*SW, SW, pb);
label(rotate(-45)*"SE", O+r*SE, SE, pb);
```

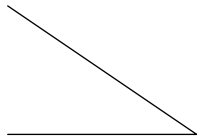
2. Đường dẫn – path

2.1. Path

Đường dẫn trong Asymptote là kiểu dữ liệu **path**: đoạn nối thẳng hoặc đường nối cong giữa hai điểm hoặc nhiều điểm lại với nhau. Đường dẫn có thể khép kín hoặc không khép kín. Tùy theo cách chỉ định mà Asy có các kiểu đường dẫn.

Path thẳng --

Để định nghĩa p là đoạn thẳng nối giữa hai điểm A và B , Mando dùng lệnh **path** $p = A--B$;

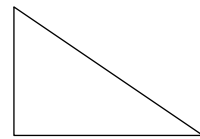


Code 12: Đường dẫn thẳng

```
unitsize(1cm);
pair A=(0,0), B=(2.5,0), C=(0,1.7);
path p = A--B--C;
draw(p); // Ve path p
// draw(p) = draw(A--B--C);
```

Code 13: Tam giác

```
unitsize(1cm);
pair A=(0,0), B=(2.5,0), C=(0,1.7);
path p = A--B--C--cycle; //
// -- cycle la noi C voi A lai de duoc duong khép kín
draw(p); // Ve path p (tam giac ABC)
```

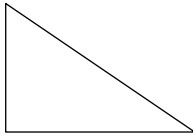


path $p = A--B--C--cycle$; là **path** khép kín nối lại từ 3 điểm A, B, C .

Path cong ..

Để định nghĩa p là đường cong nối giữa 3 điểm A, B và C , Mando dùng lệnh **path** $p = A..B..C$;

Code 14: Đường cong qua 3 điểm

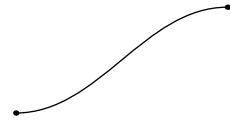


```
unitsize(1cm);
dotfactor = 3; // Tang do day cua diem
pair A =(0,0), B=(0.5,0.3), C=(2.5,0);
path p = A..B..C; // path cong qua 3 diem
dot(A--B--C); // Ve 3 diem A,B,C cung mot lenh draw
draw(p); // Ve path p
```

Để định nghĩa p là đường cong nối giữa điểm A và B, Mando dùng lệnh **path** p = A..B; với lệnh tùy chọn {...} sau A và trước B.

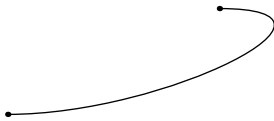
Code 15: Đường cong qua hai điểm 1

```
unitsize(1cm);
dotfactor = 3; // Tang do day cua diem
pair A =(0,0), B=(2,1);
/*-- Duong cong tu diem A di ra theo huong Dong vao diem
B theo huong doi cua Dong (la Tay)*/
path p = A{E}..{E}B; // path cong qua 2 diem
dot(A--B); // Ve 2 diem A,B cung mot lenh draw
draw(p); // Ve path p
```



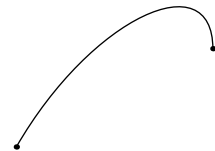
Code 16: Đường cong qua hai điểm 2

```
unitsize(1cm);
dotfactor = 3; // Tang do day cua diem
pair A =(0,0), B=(2,1);
/*-- Duong cong tu diem A di ra theo huong Dong vao diem
B theo huong doi cua Tay (la Dong)*/
path p = A{E}..{W}B; // path cong qua 2 diem
dot(A--B); // Ve 2 diem A,B cung mot lenh draw
draw(p); // Ve path p
```



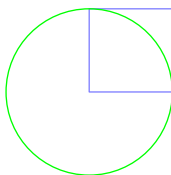
Code 17: Đường cong qua hai điểm 3

```
unitsize(1cm);
dotfactor = 4; // Tang do day cua diem
pair A =(0,0), B=(2,1);
/*-- Duong cong tu diem A di ra theo huong 60 do vao diem
B theo huong doi -90 la 90 do */
path p = A{dir(60)}..{dir(-90)}B;
dot(A--B); // Ve 3 diem A,B cung mot lenh draw
draw(p); // Ve path p
```



2.1.1. Path đơn vị

Trong asymptote có 2 **path** đơn vị định nghĩa sẵn là: **unitcircle** đường tròn đơn vị có tâm là gốc tọa độ (0,0), bán kính bằng 1 và **unitsquare** hình vuông cạnh 1 đơn vị có một đỉnh là (0,0) và có hai cạnh kề nằm trên tia Ox, Oy .



Code 18: Đường tròn, hình vuông đơn vị

```
unitsize(1.6cm);
path Hv = unitsquare;
path Ht = unitcircle;
//---- Ve Hv,Ht voi mau red,blue
draw(Hv,lightblue); // == draw(unitsquare);
draw(Ht,green); //== draw(unitcircle);
```

2.2. Path controls

2.2.1. Một điểm điều khiển

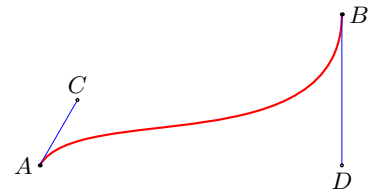
`path` cong đi qua hai điểm A và B với hai điểm C điểm điều khiển. Mando dùng lệnh sau `path p = A .. controls C .. B`; Ở đây, điểm C sẽ điều khiển `path` cong đi ra từ A vào B . Tương tự, CA và CB là hai tiếp tuyến của đường cong tại A và B .

2.2.2. Hai điểm điều khiển

`path` cong đi qua hai điểm A và B với hai điểm C, D là hai điểm điều khiển. Mando dùng lệnh sau `path p = A .. controls C and D .. B`; Ở đây, điểm C sẽ điều khiển `path` cong đi ra từ A và điểm D điều khiển `path` cong vào B . Mando tạm hiểu như thế này, CA là tiếp tuyến của đường cong tại A và DB là tiếp tuyến của đường cong tại B .

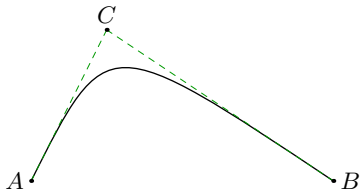
Code 19: `path` điều khiển bởi hai điểm

```
unitsize(1cm);
defaultpen(fontsize(9pt)); // size nhan
dotfactor = 3; // phong to diem
pair A = (0,0), B = (4,2);
real rc = 1, Gc = 60; //--> Lay diem C
real rd = 2, Gd = -90; //--> Lay diem D
/-- Lay diem C, D theo toa do cuc
pair C = A + rc * dir(Gc), D = B + rd * dir(Gd);
/-- path p
path p = A .. controls C and D .. B;
draw(p, red + 0.8pt);
/------ Noi AC, noi BD
draw(A--C, blue + 0.3pt);
draw(D--B, blue + 0.3pt);
/------ To nhan
dot("$A$", A, W); dot("$B$", B, E);
dot("$C$", C, N, Fill(white));
dot("$D$", D, S, Fill(white));
```



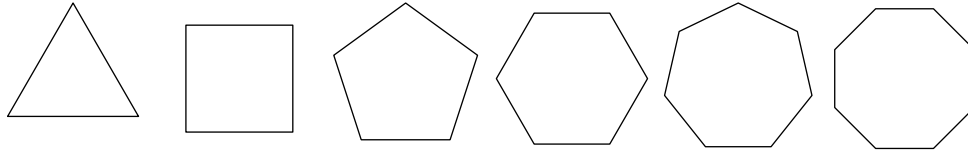
Code 20: `path` điều khiển bởi một điểm

```
unitsize(1cm);
defaultpen(fontsize(9pt)); // size nhan
dotfactor = 3; // phong to diem
pair A = (0,0), B = (4,0), C = (1,2);
/----- Tao path
path p = A .. controls C .. B; draw(p);
draw(A--C, 0.3pt + 0.6*green + dashed);
draw(C--B, 0.3pt + 0.6*green + dashed);
dot("$A$", A, W); dot("$B$", B, E); dot("$C$", C, N);
```



2.2.3. Đa giác đều n cạnh

`path` đa giác đều n cạnh có tâm là gốc tọa độ O và khoảng cách từ tâm đến một đỉnh bằng 1 có pháp `path Poly = polygon(n)`;

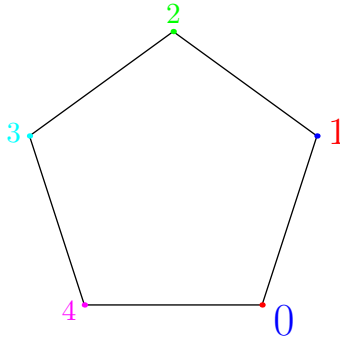


Code 21: Các đa giác đều

```
unitsize(1cm);
/-- Vẽ các giác gia đều 3,4,5,6,7,8,9 cạnh
for (int i =3; i<=8; ++i){
    draw(shift(2.2*i+3,0)*polygon(i)); // -- Vòng lặp for tính tiền đa giác đều n cạnh
}
```



`pair A=polygon(n),i);` Lấy đỉnh thứ i ($0 \leq i \leq n-1$) của đa giác đều n cạnh.

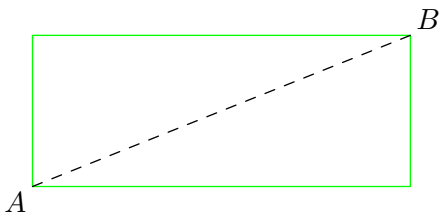


Code 22: Lấy các điểm trên ngũ giác đều

```
unitsize(2cm);
draw(polygon(5));
dot(Label("$0$",align=dir(-30),fontsize(16pt)+blue),point(polygon(5),0),red+2pt);
dot(Label("$1$",align=dir(10),fontsize(14pt)+red),point(polygon(5),1),blue+2pt);
dot(Label("$2$",align=dir(90),point(polygon(5),2),green+2pt);
dot(Label("$3$",align=dir(170),point(polygon(5),3),cyan+2pt);
dot(Label("$4$",align=dir(-160),point(polygon(5),4),magenta+2pt);
```

2.2.4. Hình chữ nhật

`path box(A,B);` Hình chữ nhật có đường chéo AB.



Code 23: Hình chữ nhật có 2 điểm trên đường chéo

```
unitsize(1cm);
pair A = (-2,-1), B=(3,1);
path h = box((A),(B));
draw(h,green);
draw(A--B,dashed);
label("$A$",A,SW); label("$B$",B,NE);
```

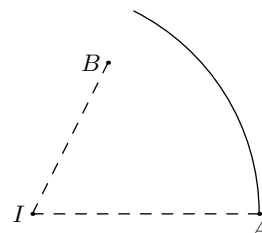
2.2.5. Đường tròn – cung tròn

Đường tròn, cung tròn trong asymptote cũng là kiểu dữ liệu `path` được định nghĩa theo lệnh sau:

- 1) `path ci = circle(pair I, real r);` Đường tròn ci có tâm là I và bán kính r .
- 2) `path ar = arc(pair I, real r, real Goc1, real Goc2);` Cung ar có tâm I , bán kính r , quét từ $Goc1$ độ đến $Goc2$ độ.
- 3) `path ar = arc(pair I, pair A, pair B);` Cung ar có tâm I , bán kính $r = AB$, quét từ điểm A đến điểm cuối nằm trên đường thẳng IB .

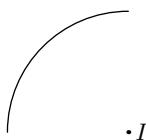
Code 24: Cung quét từ A đến B

```
unitsize(1cm); dotfactor =3;
pair I =(0,0), A =(3,0), B =(1,2);
//-----
path ar = arc(I,A,B); draw(ar);
dot("$I$",I,W); dot("$A$",A,S);
dot("$B$",B,W);
draw(I--A,dashed);
draw(I--B,dashed);
```



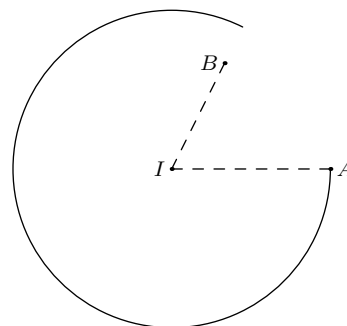
Code 25: Cung tròn bán kính r với số đo 90°

```
unitsize(1.2cm); dotfactor =3;
real r=1.6;
pair I =(0,0);
//----- Ve cung 90 do tu 90 -> 180
draw(arc(I,r,90,180));
dot("$I$",I,E);
```

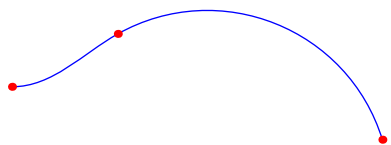


Code 26: Cung quét từ A đến B cùng chiều kim đồng hồ

```
unitsize(1cm); dotfactor = 3;
defaultpen(fontsize(8pt));
pair I =(0,0), A =(3,0), B =(1,2);
//-- CW cung chiều kim đồng hồ (Chieu am)
path ar = arc(I,A,B,CW); draw(ar);
dot("$I$",I,W); dot("$A$",A,S);
dot("$B$",B,W);
draw(I--A,dashed);
draw(I--B,dashed);
```



2.2.6. Một số ví dụ khác về path

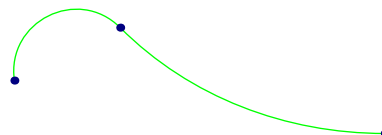


Code 27: Đường cong qua 3 điểm

```
unitsize(0.7cm);
path p=(-3,1){right}..(-1,2)..(4,0);
draw(p,blue); // ve p
dot(p,red); // Ve cac diem p di qua.
```




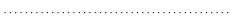



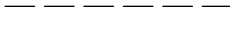

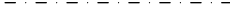


Code 28: Đường cong qua 3 điểm 2

```
unitsize(1cm);
path p=(-3,1)..{dir(-45)}(-1,2)..{right}(4,0);
draw(p,green); // Ve p
dot(p,0.5*blue); // Ve cac diem p di qua.
```



2.3. Kiểu đường dẫn

Hỗ trợ các kiểu sau: `solid` (mặc định), `dotted`, `dashed`, `longdashed`, `dashdotted`, `longdashdotted`. Các kiểu đường dẫn này thuộc kiểu dữ liệu `pen`. Ngoài ra ta còn có thể định nghĩa kiểu đường dẫn như sau `pen p = linetype(" ")` và các tùy chọn ẩn hiện.

	<code>solid+solid+blue+1bp</code>		<code>linetype(" ")</code>
	<code>dotted+.5blue+1bp</code>		<code>linetype("0 4")</code>
	<code>dashed+red+1bp</code>		<code>linetype("8 8")</code>
	<code>longdashed+.5red+1bp</code>		<code>linetype("24 8")</code>
	<code>dashdotted+green+1bp</code>		<code>linetype("8 8 0 8")</code>
	<code>longdashdotted+.5green+1bp</code>		<code>linetype("0 4 4 4")</code>




2.4. Kiểu nối và đầu mút của path

Kiểu đầu mút

Có 3 loại kiểu đầu mút của `path` thuộc kiểu dữ liệu `pen`.

- `pen squarecap = linecap(0)`; vuông
- `pen roundcap = linecap(1)`; bo tròn
- `pen extendcap = linecap(2)`; mở rộng

Code 29: Kiểu đầu mút

	<code>squarecap</code>	<code>unitsize(1cm);</code>
	<code>roundcap</code>	<code>path gA = (0,0)--(3,0);</code>
	<code>extendcap</code>	<code>draw(Label("squarecap",align=15E),gA,squarecap+10bp);</code>
		<code>path gB = (0,-0.6)--(3,-0.6);</code>
		<code>draw(Label("roundcap",align=15E),gB,roundcap+10bp);</code>
		<code>path gC = (0,-1.2)--(3,-1.2);</code>
		<code>draw(Label("extendcap",align=15E),gC,extendcap+10bp);</code>

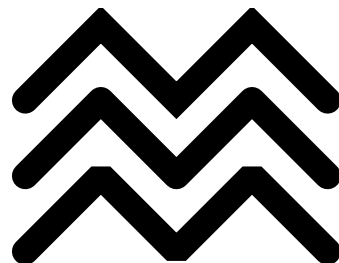
Kiểu nối

Có 3 kiểu nối `miterjoin`, `roundjoin`, `beveljoin` thuộc kiểu dữ liệu `pen`.

- `pen miterjoin = linejoin(0)`; Nhọn
- `pen roundjoin = linejoin(1)`; Bo tròn
- `pen beveljoin = linejoin(2)`; Cắt

Code 30: Kiểu nối

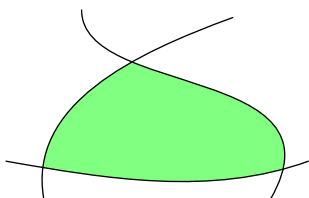
```
unitsize(1cm);
transform t=shift(0,-1);
path p=(0,0)--(1,1)--(2,0)--(3,1)--(4,0);
draw(t~1*p,10bp+miterjoin);
draw(t~2*p,10bp+roundjoin);
draw(t~3*p,10bp+beveljoin);
```



2.5. Tạo path kín từ các path

`path buildcycle(path[] p);` path kín là giao của các các path.

Code 31: Tạo đường dẫn kép kín

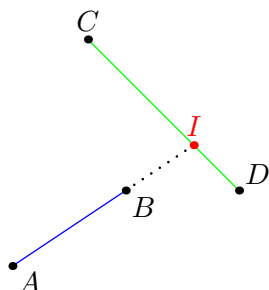


```
unitsize(1cm);
//-- Cac diem
pair A = (0,0), B=(4,0);
pair C = (3.5,-0.5), D =(1,2);
pair E = (3,1.9), F=(0.5,-0.5);
//-- path
path
    a = A{dir(-10)}..{dir(20)}B,
    b = C{dir(60)}..{dir(90)}D,
    c = E{dir(200)}..{dir(-80)}F,
    p = buildcycle(a,c,b);
//-----
fill(p,lightmagenta);
draw(a^^b^^c);
```

2.6. Giao điểm – điểm trên path

Giao điểm của 2 path

- `pair extension(pair A, pair B, pair C, pair D);` Giao điểm của đường thẳng AB và CD .
- `pair intersectionpoint(path p, path q);` Giao điểm của 2 path
- `pair[] intersectionpoints(path p, path q);` Mảng giao điểm của 2 path.
- `pair interp(A,B,k);` Điểm ảnh của B qua phép vị tự tâm A , tỷ số k .

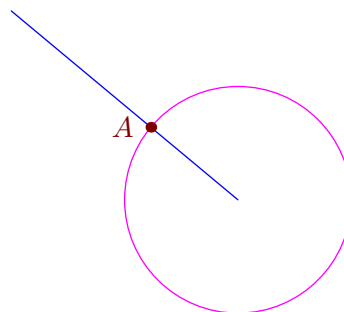


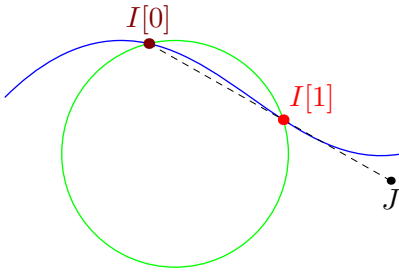
Code 32: Giao điểm của 2 đường thẳng

```
unitsize(1cm);
// ---- 4 diem
pair A=(0,1),B=(1.5,2),C=(1,4),D=(3,2);
// -----
draw(A--B,blue);
draw(C--D,green);
dot("$A$",A,SE); dot("$B$",B,SE);
dot("$C$",C,N); dot("$D$",D,NE);
// ---- giao diem
pair I=extension(A,B,C,D);
dot("$I$",I,N,red);
draw(B--I,1pt+dotted);
```

Code 33: Giao điểm của 2 path

```
unitsize(0.5cm);
//-- cac path
path p=(-5,5)--(1,0),
    c=circle((1,0),3);
pair A=intersectionpoint(p,c);
// -- ve diem, path
draw(p,blue); draw(c,magenta);
dot("$A$",A,2W,4bp+brown);
```



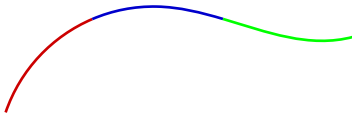


Code 34: Mảng giao điểm

```
unitsize(1cm);
defaultpen(.5bp); // Mac dinh do day cua path
// Dinh nghia path
path p=(-3,1){dir(45)}..(-1,2)..{dir(10)}(4,0),
      c=circle(0,2); // Duong tron Tam O, bk 2
pair [] I =intersectionpoints(p,c);
// Ve cac giao diem
draw(p,blue); draw(c,green);
//--- Diem vi tu
pair J=interp(I[0],I[1],1.8);
draw(I[0]--J,dashed+.3pt);
dot("$J$",J,S);
//--- Hai diem giao
dot("$I[0]$",I[0],N,4bp+brown);
dot("$I[1]$",I[1],NE,4bp+red);
```

2.7. Path con – subpath

- `real length(p)`; Độ dài của `path` `p`.
- `path subpath(path p, real a, real b)`; `path` của `path` `p` từ `a` đến `b`.



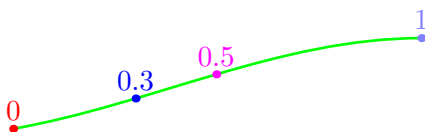
Code 35: 3 path con trên 1 path

```
unitsize(2cm);
//-- Path cha
path p=(0,0)..(1,0.7)..(2.3,.5){dir(15)};
//-- Cac path con
path
    pA = subpath(p,0,length(p)/3),
    pB = subpath(p,length(p)/3,2*length(p)/3),
    pC = subpath(p,2*length(p)/3,length(p));
//--- Ve cac path
draw(pA,1pt+.8red+linecap(0));
draw(pB,1pt+.8blue+linecap(0));
draw(pC,1pt+green+linecap(0));
```

2.8. Điểm trên path

Đối với `path` đi qua 2 điểm

`pair point(path p, real t)`; Lấy điểm cuối trên `path` `p` con sao cho tỷ số độ dài của `path` con và `path` cha bằng `t`, với `t` thuộc đoạn `[0;1]`.



Code 36: Điểm trên path 1

```
unitsize(1.2cm);
path p=(0,0){dir(10)}..{dir(0)}(4.5,1);
draw(p,green+1bp);
//--- Cac diem tren path
dot("$0$",point(p,0),N,red);
dot("$0.3$",point(p,0.3),N,blue);
dot("$0.5$",point(p,0.5),N,magenta);
dot("$1$",point(p,1),N,lightblue);
```

Đối với **path** đi qua 3 điểm trở lên

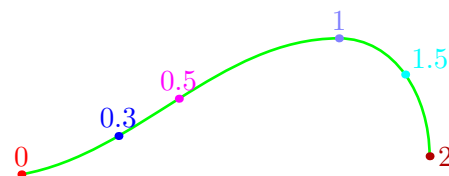
Giả sử ta có **path** p đi qua 3 điểm thứ tự A, B, C , khi đó **path** p được xem như là 2 **path** con gồm: pAB path nối từ A tới B và pBC path nối từ B tới C . Khi đó

pair **point**(**path** p , **real** t); sẽ lấy điểm trên p với t thuộc đoạn $[0; 2]$.

- Nếu t thuộc đoạn $[0; 1]$ thì điểm được lấy trên pAB
- Nếu t thuộc đoạn $[1; 2]$ thì điểm được lấy trên pBC

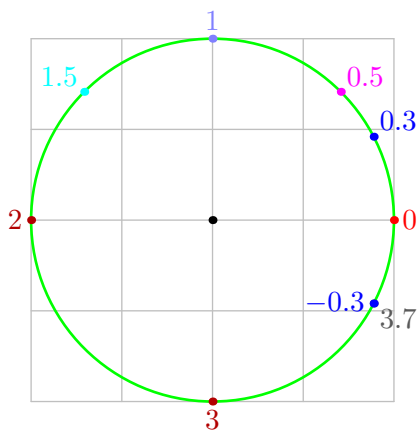
Code 37: Điểm trên **path** 2

```
unitsize(1.2cm);
pair A=(0,0), B=(3.5,1.5), C = (4.5,0.2);
path p=A{dir(10)}..{dir(0)}B{dir(0)}..{dir(-90)}C;
draw(p,green+1bp);
/-- Các điểm trên path
dot("$0$",point(p,0),N,red);
dot("$0.3$",point(p,0.3),N,blue);
dot("$0.5$",point(p,0.5),N,magenta);
dot("$1$",point(p,1),N,lightblue);
dot("$1.5$",point(p,1.5),NE,cyan);
dot("$2$",point(p,2),E,0.7*red);
```



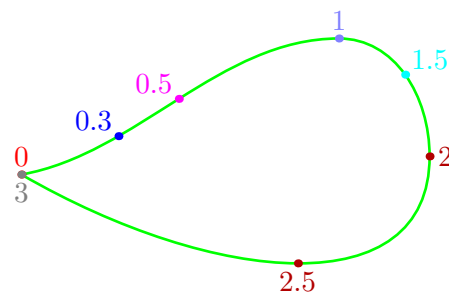
Code 38: Điểm trên **path** kín 1

```
import geometry; /-- Dung de tao luoi
unitsize(1.2cm);
add(shift(-2,-2)*grid(4,4,mediumgray)); /--Luoi
path p=circle(0,2); /-- duong tron path kín
draw(p,green+1bp); dot(0);
/-- Các điểm trên path
dot("$0$",point(p,0),E,red);
dot("$0.3$",point(p,0.3),NE,blue);
dot("$0.5$",point(p,0.5),N,magenta);
dot("$1$",point(p,1),N,lightblue);
dot("$1.5$",point(p,1.5),NW,cyan);
dot("$2$",point(p,2),W,0.7*red);
dot("$3$",point(p,3),S,0.7*red);
dot("$3.7$",point(p,3.7),SE,0.7*gray);
dot("$-0.3$",point(p,-0.3),W,blue);
```



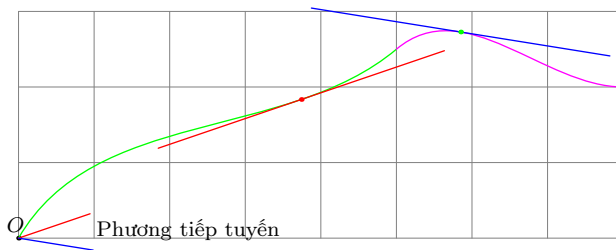
Code 39: Điểm trên **path** kín 2

```
unitsize(1.2cm);
pair A=(0,0), B=(3.5,1.5), C = (4.5,0.2);
path p=A{dir(10)}..{dir(0)}B{dir(0)}..{dir(-90)}C{dir(-90)}..{dir(150)}A..cycle;
draw(p,green+1bp);
/-- Các điểm trên path
dot("$0$",point(p,0),N,red);
dot("$0.3$",point(p,0.3),NW,blue);
dot("$0.5$",point(p,0.5),NW,magenta);
dot("$1$",point(p,1),N,lightblue);
dot("$1.5$",point(p,1.5),NE,cyan);
dot("$2$",point(p,2),E,0.7*red);
dot("$2.5$",point(p,2.5),S,0.7*red);
dot("$3$",point(p,3),S,gray);
```



2.9. Tiếp tuyến tại điểm thuộc `path`

`pair dir(path p, real t, bool normalize=true);` Phương tiếp tuyến của `path` `p` tại điểm `t` với $t \geq 0$ tùy thuộc vào `path` `p` được tạo bởi bao nhiêu `path` con.



Code 40: Tiếp tuyến đường cong

```
import math; //--- dùng để tạo lưới.
unitsize(1cm);
texpreamble("\usepackage[utf8]{vietnam}"); //-- Gọi tiếng viet
defaultpen(fontsize(8pt));
dotfactor=3.5;
//--- Tạo lưới
add(grid(8,3,gray+0.3pt));
//--- Các điểm để tạo path cong
pair A=(0,0), B=(5,2.5), C=(8,2);
dot("$O$", (0,0), dir(100));
path p=A{dir(60)}..{dir(40)}B..{dir(0)}C;
//--- Path con của path p
draw(subpath(p,0,1),green); draw(subpath(p,1,2),magenta);
//--- Các số thực để lấy điểm trên path p
real m=0.75, n=1.3;
//-- Tiếp tuyến 1
pair
    X=point(p,m), //-- Điểm X thuộc path p với m=0.75
    Y=dir(p,m), //-- Phương của tiếp tuyến tại X
    C=X+2Y, //-- Điểm C thuộc tiếp tuyến
    D=X-2Y; //-- Điểm D thuộc tiếp tuyến
draw(C--D,red); //-- vẽ tiếp tuyến
dot(X,red); //-- Tiếp điểm
draw((0,0)--Y,red); label(shift((0,-0.2cm))*"Phương tiếp tuyến",Y,E); //-- vẽ phương của tiếp
tuyến
//--- Tiếp tuyến 2
pair U=point(p,n), V=dir(p,n);
draw((U+2V)--(U-2V),blue); draw((0,0)--V,blue); dot(U,green);
```

3. Vẽ

Để vẽ một đối tượng `path` trong asymptote, Mando dùng lệnh

```
void draw(
    picture pic=currentpicture, // Ảnh hiện tại
    Label L="", // Nhan dat trong dau " " hoặc định nghĩa Label ở mục nhan.
    path g, // path g
    align align=NoAlign, // N, S, W, E, NE, NW, SE, SW, LeftSide, RightSide, Center, NoAlign
    pen p=currentpen, // Màu mặc định
    arrowbar arrow=None, // Mũi tên: Arrow, BeginArrow, MidArrow, EndArrow, Arrows
    arrowbar bar=None, // Thanh ngang: BeginBar, EndBar, Bars
```

```
margin margin=NoMargin, // Co path
Label legend="", // Chu thích path trong khung vẽ đồ thị
marker marker=nomarker // Danh dấu trên path
);
```

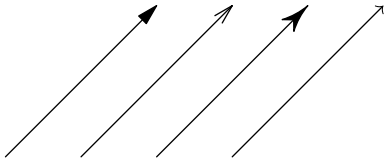


- align: gồm N, S, E, W, NE, NW, SE, SW, LeftSide, RightSide, Center.
- arrow: gồm Arrow, BeginArrow, MidArrow, EndArrow, Arrows
- bar: gồm BeginBar, EndBar, Bars.

3.1. Mũi tên – Arrow – Bar

Có 4 kiểu mũi tên: Arrow, Arrow(SimpleHead), Arrow(HookHead), Arrow(TeXHead).

Code 41: 4 kiểu mũi tên

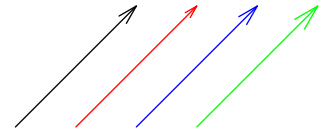


```
unitsize(1cm);
draw((0,0)--(2,2),Arrow); // Vẽ path (0,0)--(2,2) với mũi
                             tên Arrow mặc định
draw((1,0)--(3,2),Arrow(SimpleHead)); // kiểu SimpleHead
draw((2,0)--(4,2),Arrow(HookHead)); // kiểu HookHead
draw((3,0)--(5,2),Arrow(TeXHead)); // Kiểu TeXHead
```

Các tùy chọn cho mũi tên và vị trí đặt mũi tên trên path

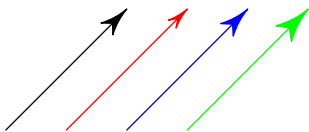
Code 42: Độ dài kiểu mũi tên SimpleHead

```
unitsize(1cm); // SimpleHead với độ dài mũi tên tùy chọn
draw((0,-3)--(2,-1),Arrow(SimpleHead));
draw((1,-3)--(3,-1),red,Arrow(SimpleHead,5bp));
draw((2,-3)--(4,-1),blue,Arrow(SimpleHead,8bp));
draw((3,-3)--(5,-1),green,Arrow(SimpleHead,10bp));
```



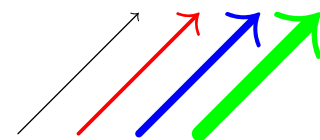
Code 43: Độ dày của kiểu mũi tên HookHead

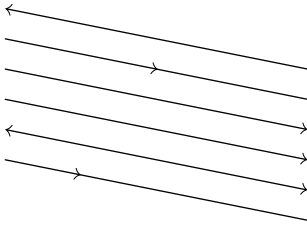
```
unitsize(0.8cm); // HookHead với độ dài tùy chọn
draw((0,0)--(2,2),Arrow(HookHead));
draw((1,0)--(3,2),red,Arrow(HookHead,5bp));
draw((2,0)--(4,2),blue,Arrow(HookHead,8bp));
draw((3,0)--(5,2),green,Arrow(HookHead,10bp));
```



Code 44: Mặt định kiểu TeXHead, thay đổi độ dày mũi tên

```
unitsize(1cm); // độ dày của mũi tên: size = a mm
DefaultHead=TeXHead; // Mặt định kiểu TeXHead
draw((0,-3)--(2,-1),Arrow());
draw((1,-3)--(3,-1),.5mm+red,Arrow(size=1mm));
draw((2,-3)--(4,-1),1mm+blue,Arrow(size=1.5mm));
draw((3,-3)--(5,-1),2mm+green,Arrow(size=2mm));
```



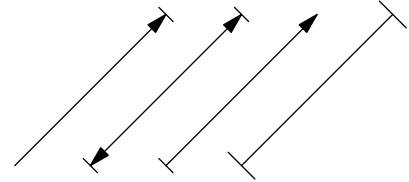


Code 45: Vị trí đặt mũi tên

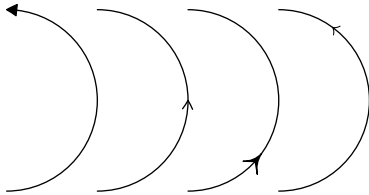
```
unitsize(1cm);
DefaultHead=TeXHead; // mat dinh kieu TeXHead
draw((0,1)--(5,0),BeginArrow()); // Dat mũi tên o đầu
draw((0,.5)--(5,-.5),MidArrow()); // Dat mũi tên o giữa
draw((0,0)--(5,-1),EndArrow()); // Dat mũi tên o cuối
draw((0,-.5)--(5,-1.5),Arrow()); // Mũi tên o cuối
draw((0,-1)--(5,-2),Arrows()); // Hai mũi tên o 2 đầu mút
draw((0,-1.5)--(5,-2.5),Arrow(Relative(.25))); // vị
trí 0.25
```

Code 46: Tùy chọn Bar cho path

```
unitsize(1cm);
draw((0,0)--(2,2),Arrow,Bar); // bar o cuối và mũi tên
draw((1,0)--(3,2),Arrows,Bars); // bar o hai đầu mút
draw((2,0)--(4,2),Arrow,BeginBar); // bar o đầu
draw((3,0)--(5,2),Bars(5mm)); // bar o hai đầu, dài 5mm
```



Code 47: Mũi tên theo path



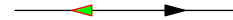
```
unitsize(.8cm);
draw((0,0)..(2,2)..(0,4),ArcArrow);
draw((2,0)..(4,2)..(2,4),ArcArrow(SimpleHead,Relative(0.5)));
draw((4,0)..(6,2)..(4,4),ArcArrow(HookHead,Relative(0.3)));
draw((6,0)..(8,2)..(6,4),ArcArrow(TeXHead,Relative(0.8)));
```

Cộng thêm mũi tên vào path



Code 48: Cộng thêm mũi tên và vẽ lại path

```
unitsize(1cm);
//---
path p=(0,0)--(3,0);
draw(p,Arrow(Relative(0.3)));
//--- thêm mũi tên ở vị trí 0.7
add(arrow(p,Relative(0.7)));
```



Code 49: Cộng thêm mũi tên và không vẽ lại path

```
unitsize(1cm);
path p = (0,0)--(3,0);
draw(p,Arrow(Relative(0.75)));
//-- Cộng thêm mũi tên nhưng không vẽ lại path p
add(arrow(reverse(p),invisible,FillDraw(green,red),Relative(0.75)));
```

3.2. Nhãn

3.2.1. Nhãn cho điểm

```
void label(
    picture pic=currentpicture, //-- Anh hiện hành
    Label L, // -- 4 kiểu Label liệt kê ở dưới
    pair position, //--- Điểm gán nhãn
    align align=NoAlign, //--- Hướng nhãn
    pen p=currentpen, // -- Màu nhãn,
    filltype filltype=NoFill //-- Tô màu cho nhãn
);
```

```
Label Label(
```

```

string s="", // nhan dat trong dau " "
pair position, // Vi tri dat nhan: BeginPoint=Relative(0); MidPoint=Relative(0.5);
align align=NoAlign, // Huong nhan
pen p=nullpen,
embed embed=Rotate, // quay nhan
filltype filltype=NoFill //-- To nhan
);

```

```

Label Label(string s="", align align=NoAlign, pen p=nullpen,
embed embed=Rotate, filltype filltype=NoFill);

```

```

Label Label(Label L, pair position, align align=NoAlign,
pen p=nullpen, embed embed=L.embed, filltype filltype=NoFill);

```

```

Label Label(Label L, align align=NoAlign, pen p=nullpen, embed
embed=L.embed, filltype filltype=NoFill);

```

A

pair position gồm: `BeginPoint=Relative(0)`; `MidPoint=Relative(0.5)`; `EndPoint=Relative(1)`. Tổng quát `Relative(real a)`; a thuộc đoạn $[0; 1]$.

3.2.2. Nhãn cho path

Để gắn nhãn cho `path`, ta dùng cú pháp sau

```

void label(
picture pic=currentpicture,
Label L, //-- kieu Label 1 trong 4 kieu Label liet ke o tren
path g, // -- path can gan nhan
align align=NoAlign, //-- Huong nhan
pen p=currentpen, //-- mau
filltype filltype=NoFill //-- To
);

```

Hoặc gắn nhãn kèm trong lệnh `draw`

```

void draw(
picture pic=currentpicture, // Anh hien tai
Label L="", // Nhan dat trong dau " " hoac dinh nghia Label o muc nhan.
path g, // path g
align align=NoAlign, // N, S, W, E, NE, NW, SE, SW, LeftSide, RightSide, Center, NoAlign
pen p=currentpen, // Mau mac dinh
arrowbar arrow=None, // Mui ten: Arrow, BeginArrow, MidArrow, EndArrow, Arrows
arrowbar bar=None, // Thanh ngang: BeginBar, EndBar, Bars
margin margin=NoMargin, // Co path
Label legend="", // Chu thích path trong khung ve do thi
marker marker=nomarker // Danh dau tren path
);

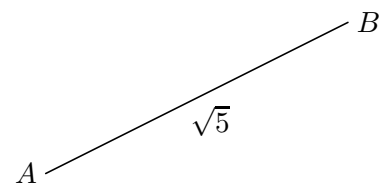
```

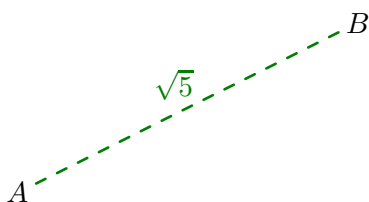
Code 50: Nhãn mặc định cho đoạn thẳng

```

unitsize(2cm);
pair A=(0,0), B=(2,1);
path seg = A--B;
label("$A$",A,W);
label("$B$",B,E);
draw("$\sqrt{5}$", // nhan
seg,
linewidth(0.6pt));

```



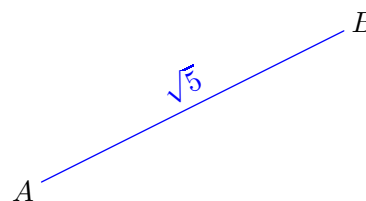


Code 51: *align=LeftSide*

```
unitsize(2cm);
pair A=(0,0), B=(2,1);
path seg=A--B;
label("$A$",A,dir(A-B));
label("$B$",B,dir(B-A));
draw(Label("$\sqrt{5}$",
    align=LeftSide),
    seg,
    bp+deepgreen+linetype("4 5") );
```

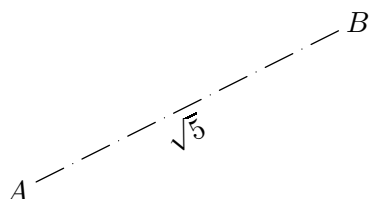
Code 52: *Quay nhãn Rotate(dir(seg))*

```
unitsize(2cm);
pair A=(0,0), B=(2,1);
path seg=A--B;
label("$A$",A,dir(A-B));
label("$B$",B,dir(B-A));
draw(Label("$\sqrt{5}$",
    Rotate(dir(seg)), // Quay nhãn theo phương AB
    align=LeftSide), // Đặt nhãn bên trái
    seg,
    blue );
```



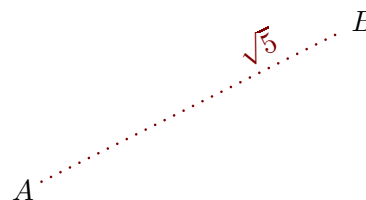
Code 53: *Quay nhãn, align=S*

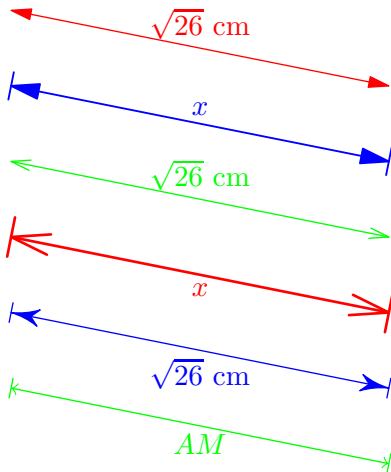
```
unitsize(2cm);
pair A=(0,0), B=(2,1);
path seg=A--B;
label("$A$",A,dir(A-B));
label("$B$",B,dir(B-A));
draw(Label("$\sqrt{5}$",
    Rotate(dir(seg)), // Quay nhãn theo phương AB
    align=S), // Đặt nhãn phía Nam
    seg,
    longdashdotted ); // Kiểu path
```



Code 54: *Đặt nhãn position=Relative(.75)*

```
unitsize(2cm);
pair A=(0,0), B=(2,1);
path seg=A--B;
label("$A$",A,dir(A-B));
label("$B$",B,dir(B-A));
draw(Label("$\sqrt{5}$",
    Rotate(dir(seg)), //-Có thể dùng dir(real g):
    quay nha góc g đo
    align=LeftSide,
    position=Relative(.75)), // Vị trí đặt nhãn
    seg,
    bp + brown + dotted ); // Kiểu path
```



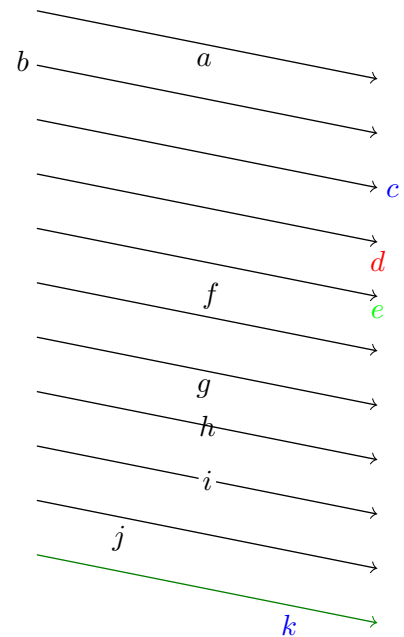


Code 55: Nhãn của đoạn thẳng có 2 đầu mút

```
unitsize(1cm);
//----
draw("$\sqrt{26}$ cm", (0,5)--(5,4), N, red, Arrows);
draw("$x$", (0,4)--(5,3), N, .7bp+blue, Arrows, Bars);
draw("$\sqrt{26}$ cm", (0,3)--(5,2), N, green, Arrows(
    SimpleHead));
draw("$x$", (0,2)--(5,1), S, 1bp+red, Arrows(SimpleHead),
    Bars);
draw("$\sqrt{26}$ cm", (0,1)--(5,0), S, blue, Arrows(
    HookHead), Bars);
draw("$AM$", (0,0)--(5,-1), S, green, Arrows(TeXHead), Bars
    );
```

Code 56: Một số kiểu hay dùng

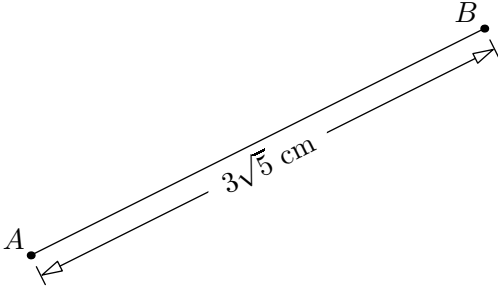
```
unitsize(1cm);
path seg=(0,4)--(5,3); // path đoạn thẳng
transform t=shift(0,-0.8); // Phep vi tu
DefaultHead=TeXHead;
draw("$a$", seg, Arrow()); // mat dinh o phia duoi
draw(Label("$b$", BeginPoint), t*seg, Arrow());
draw(Label("$c$", EndPoint, blue), t^2*seg, Arrow());
draw(Label("$d$", EndPoint, red), t^3*seg, S, Arrow());
draw(Label("$e$", EndPoint, align=S, green), t^4*seg, Arrow());
draw(Label("$f$", align=LeftSide), t^5*seg, Arrow());
draw(Label("$g$", align=RightSide), t^6*seg, Arrow());
draw(Label("$h$", align=Center), t^7*seg, Arrow());
draw(Label("$i$", align=Center, UnFill), t^8*seg, Arrow());
draw(Label("$j$", Relative(.25)), t^9*seg, Arrow());
draw(Label("$k$", Relative(.75), blue), t^10*seg, .5green,
    Arrow());
```



Khoảng cách

Kí hiệu khoảng cách cho path thẳng, đoạn thẳng. Cần thêm gói geometry.

```
void distance(picture pic=currentpicture,
    Label L="", // Nhan de trong " "
    pair A, // Diem dau
    pair B, // Diem cuoi
    bool rotated=true, // Quay nhan theo phuong cua đoạn thẳng
    real offset=3mm, // Khoảng cách từ điểm
    pen p=currentpen, // Màu
    pen joinpen=invisible, //
    arrowbar arrow=Arrows(NoFill) // To màu cho mũi tên
);
```

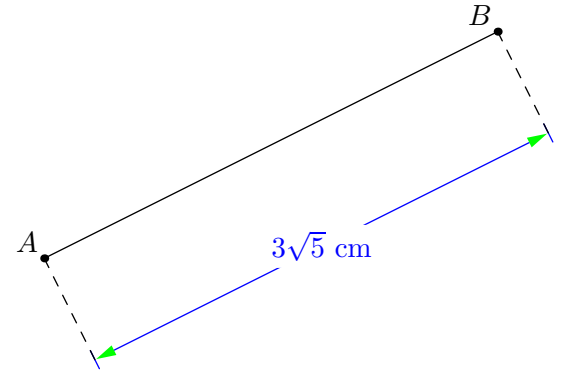


Hình 1: Ký hiệu khoảng cách

Code 57: Ký hiệu khoảng cách

```
import geometry; // -----
unitsize(1cm);
pair A=(-1,2), B=(5,5);
draw(A--B);
dot("$A$",A,NW); dot("$B$",B,NW);
distance("$3\sqrt{5}$ cm",A,B); //---
```

```
import geometry;
unitsize(1cm);
pair A=(-1,2), B=(5,5);
draw(A--B); // Vẽ đoạn thẳng AB
dot("$A$",A,NW); dot("$B$",B,NW);
distance("$3\sqrt{5}$ cm", //
A,B, // point A, point B
rotated=false, //
offset=15mm, //
p=blue, //
joinpen=dashed, //
Arrows(Fill(green)) //
);
shipout(bbox(5mm,white)); // Mờ khung hình
```



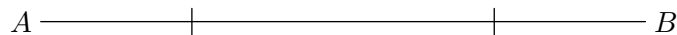
Hình 2: Ký hiệu khoảng cách

3.3. Ký hiệu cho path

Để đánh dấu cho `path` ta thêm `import markers`; Có 4 kiểu thường dùng.

3.3.1. Kiểu `StickIntervalMarker`

```
marker StickIntervalMarker(
    int i=2, int n=1, // số thanh chắn mặc định 2, thanh chắn đơn (1)
    real size=0, real space=0, // độ dài, khoảng cách thanh chắn
    real angle=0, // Góc quay thanh chắn
    pair offset=0, // Độ dịch chuyển các thanh chắn
    bool rotated=true, // Mặc định quay
    pen p=currentpen, // màu sắc thanh chắn
    frame uniform=newframe, // thêm chấm
    bool above=true // -- đặt thanh chắn trước hoặc sau path
);
```



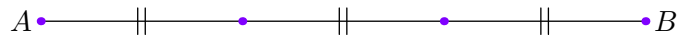
Code 58: Mặc định

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(4,0);
draw(
```

```

A--B, // path AB
StickIntervalMarker( )); //-- Mac dinh
//-----
label("$A$",A,W);label("$B$",B,E);

```

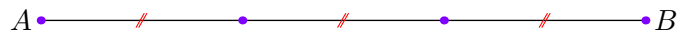


Code 59: Bổ sung chấm điểm

```

import markers;
unitsize(2cm);
pair A=(0,0), B=(4,0);
draw(
  A--B, // path AB
  StickIntervalMarker(
    3, //-- 3 thanh chan tren path
    2, //-- thanh chan doi
    dotframe(purple) // 3 thanh chan - 4 diem tren path, mau purple
  )); //-- Mac dinh
//-----
label("$A$",A,W);label("$B$",B,E);

```



Code 60: Điều khiển thanh chắn

```

import markers;
unitsize(2cm);
pair A=(0,0), B=(4,0);
draw(
  A--B, // path AB
  StickIntervalMarker(
    3, //-- 3 thanh chan
    2, //-- thanh chan doi,
    size = 6, // Do dai thanh chan
    space = 1, // khoang cach
    angle = -30, // Goc nghien
    red+0.4pt, // Mau va do day thanh chan
    dotframe(purple) // 3 thanh chan - 4 diem
  ) );
//-----
label("$A$",A,W);label("$B$",B,E);

```

3.3.2. Kiểu CrossIntervalMarker

```

marker CrossIntervalMarker(
  int i=2, int n=3, // So sao, so canh
  real size=0, //-- Ban kinh canh
  real space=0, //
  real angle=0, //-- Goc quay cua canh
  pair offset=0, //-- Do lech
  bool rotated=true, // Mac dinh qua
  pen p=currentpen, // Mau sac, do day
  frame uniform=newframe, // dotframe(color) hoac frame duoc tao la cac marker trong modul
  markers.
  bool above=true //--- dat sao truoc hoac sau path
);

```



Code 61: 2 dấu sao, 3 điểm

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(
    A--B,
    CrossIntervalMarker(
        2, // 2 đầu sao
        3, // sao 3 cánh
        bp+red, // do đây 1bp, màu đỏ
        dotframe(purple)) // 2 đầu sao 3 chấm
    );
//-----
label("$A$",A,W);label("$B$",B,E);
```



Code 62: 2 dấu sao 6 cánh, 3 điểm

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(
    A--B,
    CrossIntervalMarker(
        2, // 2 đầu sao
        6, // sao 6 cánh
        size = 6,
        angle = 45,
        bp+green, // do đây 1bp, màu đỏ
        dotframe(purple)) // 2 đầu sao 3 chấm
    );
//-----
label("$A$",A,W);label("$B$",B,E);
```

3.3.3. Kiểu CircleBarIntervalMarker

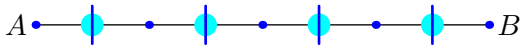
```
marker CircleBarIntervalMarker(
    int i=2, int n=1, //-- 2 stick, 1 thanh chắn
    real barsize=0, real radius=0, //-- do dài của thanh chắn, bán kính của đường tròn
    real angle=0, pair offset=0, //-- Góc quay của thanh chắn, do dịch chuan các stick
    bool rotated=true, //-- Quay thanh chắn
    pen p=currentpen, filltype filltype=NoFill, // màu, cho thanh chắn, to màu cho đường
    tron
    bool circleabove=false,
    frame uniform=newframe, //-- frame là các marker được tạo
    bool above=true
);
```

2020



```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      CircleBarIntervalMarker(3,2,
        barsize=5mm,radius=1.5mm,
        angle=-45,darkgreen,
        filltype=NoFill,dotframe(purple))
);
label("$A$",A,W);label("$B$",B,E);
```

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      CircleBarIntervalMarker(4,1,
        barsize=5mm,radius=1.5mm,bp+magenta,
        filltype=Fill(green),
        circleabove=true,dotframe(purple),
        above=false));
label("$A$",A,W);label("$B$",B,E);
```

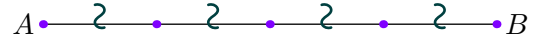
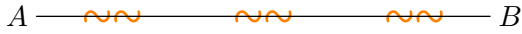


```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      CircleBarIntervalMarker(4,1,
        barsize=5mm,radius=1.5mm,
        bp+blue,filltype=Fill(cyan),
        dotframe(blue)) );
label("$A$",A,W);label("$B$",B,E);
```

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      CircleBarIntervalMarker(3,0,
        radius=1.2mm,bp+fuchsia,
        filltype=NoFill,dotframe(cyan)) );
//---
label("$A$",A,W);label("$B$",B,E);
```

3.3.4. Kiểu TildeIntervalMarker

```
marker TildeIntervalMarker(
  int i=2, int n=1, //--- 2 stick, 1 dau nga
  real size=0, real space=0, //--
  real angle=0, pair offset=0, //-- Goc quay, do dich chuyen
  bool rotated=true, //-- quay dau nga
  pen p=currentpen, frame uniform=newframe,
  bool above=true
);
```

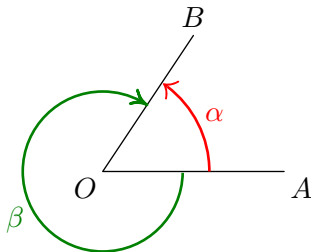
```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      TildeIntervalMarker(3,2,size=1mm,
                           space=4mm,bp+orange,above=false) );
//----
label("$A$",A,W);label("$B$",B,E);
```

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
draw(A--B,
      TildeIntervalMarker(4,1,size=1mm,
                           angle=-90,offset=(0,3), bp+darkcyan,
                           dotframe(purple),above=true) );
label("$A$",A,W);label("$B$",B,E);
```

3.4. Vẽ góc

```
void markangle(
    picture pic=currentpicture, //-- Anh hien hanh
    Label L="", //------- ten cua goc dat trong dau " "
    int n=1, //------- So cung
    real radius=0, //-- -----
    real space=0, //-------
    pair A, pair O, pair B, //-- Danh dau goc AOB theo chieu duong (nguoc chieu kim dong ho)
    arrowbar arrow=None, //-- Cung mui ten
    pen p=currentpen, //-- Mau sac
    filltype filltype=NoFill, //-- To mau cho goc
    margin margin=NoMargin,
    marker marker=nomarker //-- Gan thanh chan len cung cua goc
);
```

Code 63: Đánh dấu góc bằng cung



```
import markers;
unitsize(1.2cm);
pair O=0, A=(2,0), B=(1,1.5); draw(A--O--B);
markangle(
    "$\alpha$", //-- ten cho goc la alpha
    radius=40, //-- bkinh dat nhan 40 mm tinh tu O
    A,O,B, //-- Goc AOB
    ArcArrow(TeXHead,2), //-- Cung danh dau goc co mui ten
    bp+red //-- Mau do cho cung va text
);
markangle(
    "$\beta$", //-- ten goc la beta
    radius=30, //-- ban kinh cho goc
    B,O,A, //-- Goc BOA
    BeginArcArrow(HookHead,1mm), //-- Kieu mui ten cung
    bp+deepgreen //-- Mau
);
label("$O$",O,SW); label("$A$",A,SE); label("$B$",B,N);
```



- `pair unit(pair z);` Vectơ đơn vị có hướng vectơ z .
- `pair dir(path p, path q);` Vectơ đơn vị có hướng là tổng của 2 hướng của 2 `path` p và q .
- `pair interp(A,B,k);` Điểm ảnh của B qua phép vị tự tâm A , tỷ số k .
- `pair X = A + C-B;` X là ảnh của A qua phép tịnh tiến \overrightarrow{BC} .
- `TrueMargin(real a, real b);` Cho `path` trong tùy chọn `margin` của lệnh `draw`.



Code 65: Điểm trên phân giác, đặt điểm

```
import geometry; //- dùng để tạo lưới tọa độ
add(shift((-1,-2))*grid(5,4,lightgray));
unitsize(2cm); //-- Đơn vị tọa độ
dotfactor =3; dot("$O$",O,N);
----
pair A=(0,-0.5), B=(3,0), C=(1,2); draw(A--B--C--cycle);
dot("$A$",A,W); dot("$B$",B,S); dot("$C$",C,N);
---- duong tron don vi tam B, duong tron don vi tam A
draw(circle(B,1),dashed); draw(circle(A,1),dashed+blue);
---- D, E tren phan giac trong cua goc B, F ngoai
pair
    D = B + dir(B--A,B--C), ----
    E = B + 2*dir(B--A,B--C), ----
    F = B + (-0.8)*dir(B--A,B--C); ----
---- Duong phan giac
draw(E--interp(E,F,1.2));
---- Ve cac diem D,E,F
dot("$D$",D,dir(-160),red);
dot("$E$",E,W,blue);
dot("$F$",F,E,magenta);
---- Diem X, Y tren tren OA
pair
    X = A + unit(A), // -- AX = 1
    Y = A +(-2)*unit(A); // AY = 2
dot("$X$",X,S); dot("$Y$",Y,N);
---- Diem T tren CB sao cho CT = 0.8
pair T = C + 0.8*unit(B-C); dot("$T$",T,E,blue);
```

3.4.1. Cách tạo các marker để đánh dấu góc

Để tạo các marker là các frame dùng để đánh dấu góc Mado dùng cú pháp sau

```
marker marker(
    uniform, //-- dotframe(color) hoặc 1 trong các frame liệt kê bên dưới
    markinterval(
        1, //-- Số stick trên path, mặc định 1
        frame,
        bool true //-- Quay các stick
    );
```

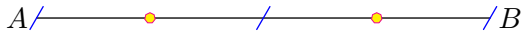
và 1 trong 4 tùy chọn frame liệt kê ở dưới.

```
frame stickframe(int n=1, real size=0, pair space=0, real angle=0, pair offset=0,
    pen p=currentpen );
```

```
frame circlebarframe(int n=1, real barsize=0, real radius=0,real angle=0, pair offset=0,
    pen p=currentpen, filltype filltype=NoFill, bool above=false );
```

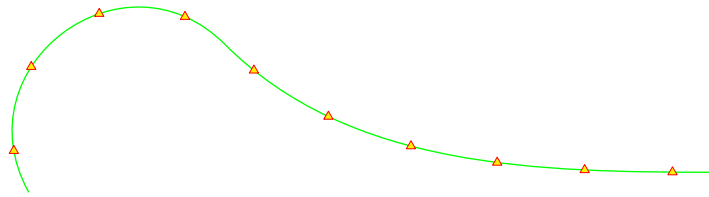
```
frame crossframe(int n=3, real size=0, pair space=0,real angle=0, pair offset=0, pen p=
    currentpen );
```

```
frame tildeframe(int n=1, real size=0, pair space=0, real angle=0, pair offset=0, pen p=
    currentpen);
```



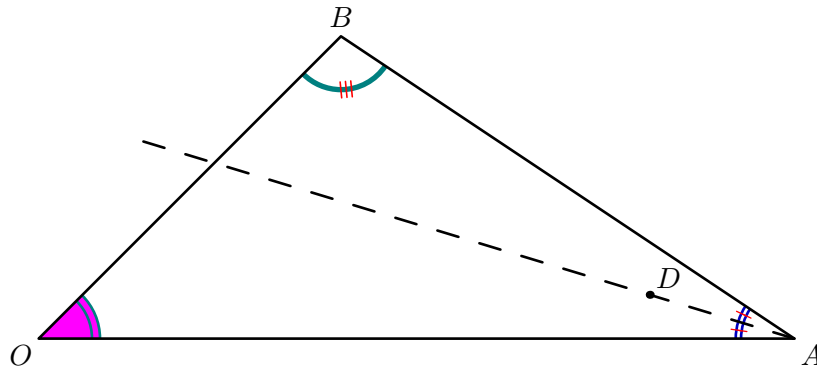
```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
//-- Tao frame that giac deu to mau vang,vien
magenta
frame f;
path p = scale(2)*polygon(7);
filldraw(f,p,Yellow,0.2pt+Magenta);
//-- Tao marker
marker mk = marker(
    stickframe(1,angle=-30,p=blue),
    markinterval(2,f)
);
draw(A--B,mk);
label("$A$",A,W);label("$B$",B,E);
```

```
import markers;
unitsize(2cm);
pair A=(0,0), B=(3,0);
//-- Tao frame that giac deu to mau vang,vien
magenta
frame f;
path p = scale(2)*polygon(7);
filldraw(f,p,Yellow,0.2pt+Magenta);
//-- Tao marker
marker mk = marker(
    //--stickframe(1,angle=-30,p=blue),
    markinterval(3,f)
);
draw(A--B,mk);
label("$A$",A,W);label("$B$",B,E);
```



Code 66: Đường cong có đánh dấu

```
import markers;
unitsize(1.3cm);
path p=(-3,0.5)..{dir(-45)}(-1,2){dir(-45)}..{right}(4,0.7);
//-- Tao frame that giac deu to mau vang,vien magenta
frame f;
path fp=scale(2)*polygon(3);
filldraw(f,fp,Yellow,0.2pt+red);
//-- Tao marker
marker mk = marker( markinterval(10,f) );
draw(p,green,mk); // Ve p
```



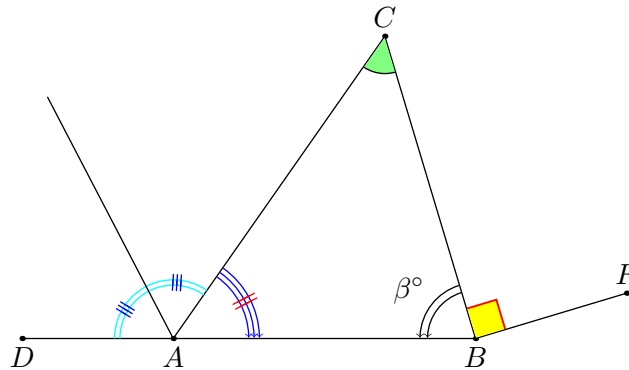
Code 67: Vẽ, đánh dấu góc

```
import markers; unitsize(2cm);
pair
```

```

O=(0,0), A=(5,0), B=(2,2),
D=A + dir(A--O,A--B); // Diem D nam tren tia p giac
//----- Ve goc O, 2 cung, to mau
markangle(n=2, radius=20, space=3, A, O, B, 1bp+deepcyan, filltype=Fill(magenta) );
//----- Tao marker cho goc B, stick 3
frame fB= stickframe(n=3, size=6, space=2,0.6*bp+red);
marker mkB = marker(markinterval(fB,true));
//-- Danh dau goc B, 1 stick
markangle(n=1, radius=20, O,B,A, p=2bp+deepcyan, mkB);
//----- Tao marker cho goc A, 2 stick, stick 1
frame fA = stickframe(n=1, size=4,red);
marker mkA = marker(markinterval(2, fA,true));
//-- Danh dau goc A
markangle(n=2, radius=20, space=2, B,A,O, bp+heavyblue, mkA);
//-----
draw(A--interp(A,D,4.5), bp+dashed); draw(A--O--B--cycle, linewidth(bp));
label("$O$", O, SW); dot("$D$", D, NE); label("$A$", A, SE); label("$B$", B, N);

```

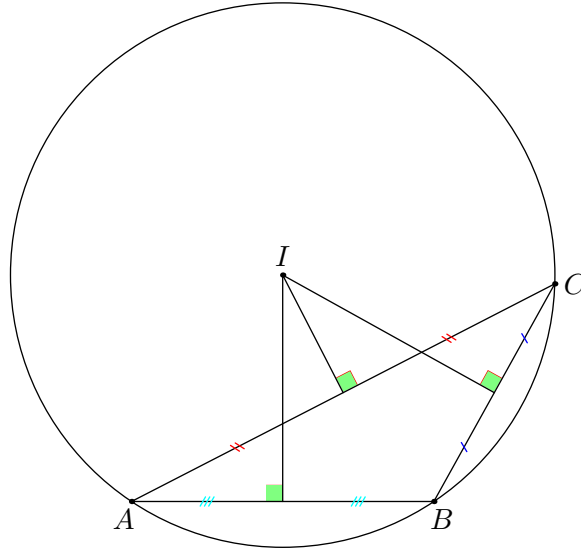


Code 68: Vẽ góc vuông

```

import markers; import geometry;
unitsize(4cm); dotfactor=4; //----- Dieu khien do day cua diem
//----- Cac diem
pair A=O, B=(1,0), C=(0.7,1), D=(-0.5,0), Fp = (B+C)/2, //-- Trung diem cua BC
F=rotate(-90,B)*Fp; //-- Quay Fp quanh B, goc -90 do, == F=rotate(-90)*(C-B)/2+B;
//-----
draw(A--B); draw(A--C); //defaultpen(linewidth(0.4pt));
//-----
draw(B--C); draw(A--D); draw(B--F);
dot("$A$", A, S); dot("$B$", B, S); dot("$C$", C, N);
dot(Label("$D$", D, S)); dot(Label("$F$", F, N));
//--- Ve goc ACB
real r = linewidth()/2; //--- Do day cua path chia 2
markangle(radius=14, A, C, B, Fill(lightgreen), TrueMargin(r, r));
//--- Ve Goc CBA
markangle("$\beta^\circ$", 2, radius=18, C, B, A, ArcArrow(TeXHead, size=0.7pt), TrueMargin(r, r));
//----- Tao marker cho goc A
frame fA = stickframe(n=2, size=5, space=2, red);
marker mkA = marker(markinterval(fA, true));
//--- Ve, danh dau goc A trong
markangle(3, space=2, B, A, C, p=blue, BeginArcArrow(TeXHead, size=0.7pt), TrueMargin(r, r), mkA);
//----- Danh dau goc vuong B, ham nay trong modulo geometry
perpendicular(
    B, //-- Goc vuong tai B
    NE, //-- O vi tri Dong Bac

```

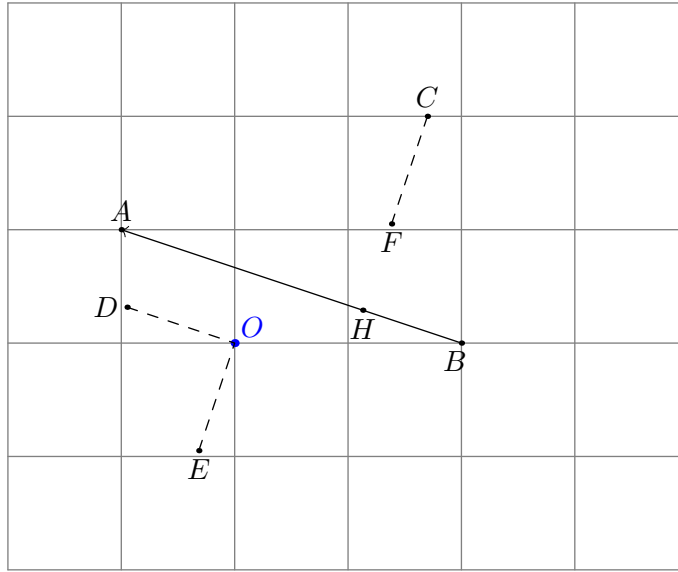



Code 70: Đường tròn ngoại tiếp tam giác

```
import markers; import geometry; //-- Dung chuc nang goc vuong
unitsize(1.6cm);
dotfactor =4; //------- Dieu khien do day cua diem
//----- Cac diem
pair
    A=(-1,0), B=(1.5,0), C=(2.5,1.8),
    mA=(B+C)/2, mB=(C+A)/2, mC=(A+B)/2, // trung diem
    pA=rotate(90,mA)*C, pB=rotate(90,mB)*A, pC=rotate(90,mC)*B; // diem tren trung truc
// Tam I, abs(pair z) la modul cua z, abs(real r) tri tuyet do cua r, abs(A-I) la do dai doan IA
pair I = extension(pA,mA,pB,mB); draw(circle(I,abs(A-I)));
//---- Danh dau goc vuong
real r = linewidth()/2;
perpendicular(mA,NE,C-mA,size=6,red,TrueMargin(r,r),Fill(lightgreen));
perpendicular(mB,NE,C-mB,size=6,red,TrueMargin(r,r),Fill(lightgreen));
perpendicular(mC,SE,A-mC,size=6,red,TrueMargin(r,r),Fill(lightgreen));
draw(I--mA^^I--mB^^I--mC); //draw(A--B--C--cycle);
draw(B--C,StickIntervalMarker(2,1,size=4,angle=-30,blue));
draw(C--A,StickIntervalMarker(2,2,size=4,space=2,angle=30,red));
draw(A--B,StickIntervalMarker(2,3,size=4,space=1.6,angle=-30,cyan));
dot("$A$",A,SSW); dot("$B$",B,SSE); dot("$C$",C,E); dot("$I$",I,N);
```

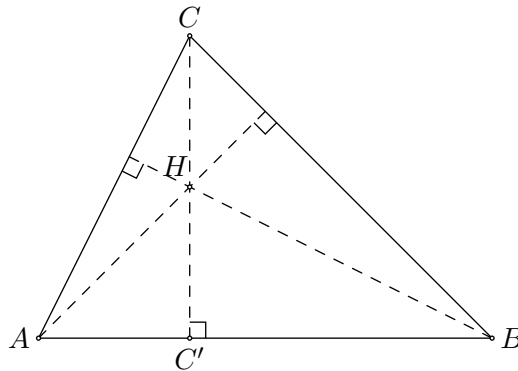


- `real length(pair z)`; Modul của z
- `real abs(pair z)`; Nếu z là 1 số thì nó là trị tuyệt đối, còn nếu z là `pair` thì nó là môđun của z .
- `real linewidth()`; độ dày mặc định của `path`.
- `pair dir(real degrees)`; là vectơ đơn vị có hướng là degrees độ.
- `real angle(pair z, bool warn=true)`; là số đo góc(radian) của z với tia Ox , nằm trong đoạn $[-\pi; \pi]$ hoặc là số 0 nếu `warn=false` và $z=(0,0)$.
- `real degrees(pair z, bool warn=true)`; là số đo góc(độ) của z với tia Ox , nằm trong đoạn $[0; 360)$ hoặc là số 0 nếu `warn=false` và $z=(0,0)$.



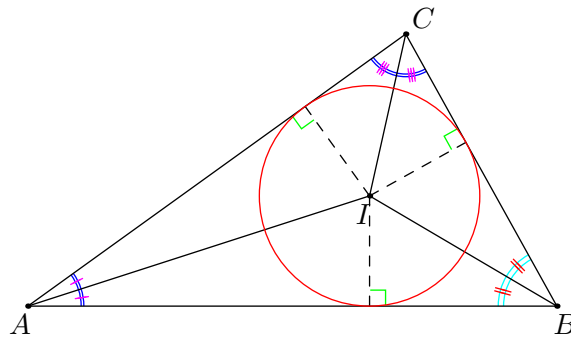
Code 71: Minh họa hình chiếu vuông góc của điểm trên đường thẳng

```
import math; //-- dùng tạo lưới
unitsize(1.5cm);
dotfactor =4; // Điều khiển độ dày điểm
//----- Tạo lưới
add(shift(-2,-2)*grid(6,5,gray));
pair O=(0,0), A=(-1,1), B=(2,0), C=(1.7,2);
//-----
dot("$O$",O,NE,blue+3bp); //-- Góc tọa độ O
//-----
real deg=degrees(A-B,false); //-- Số đo góc giữa tia BA (vectơ BA) và tia Ox.
pair
    D = dir(deg), //-- Điểm D là vectơ đơn vị theo hướng của vectơ BA, (tia OD, tia Ox) = (tia BA, tia Ox)
    E = dir(90+deg), //-- Điểm E là vectơ đơn vị vuông góc với BA
    F = C+dir(90+deg), //-- Tính điểm F theo vectơ OE
    H = extension(C,F,A,B); //-- Hình chiếu vuông góc của C trên AB
//-----
draw(B--A,Arrow(TeXHead)); draw(D--(0,0)--E,dashed);
draw(C--F,dashed); dot("$H$",H,S);
dot("$D$",D,W); dot("$E$",E,S); dot("$F$",F,S);
dot("$A$",A,N); dot("$B$",B,E); dot("$C$",C,N);
```



Code 72: Trục tâm của tam giác

```
import geometry;
unitsize(2cm); dotfactor =4;
filltype to=Fill(white); //--- To diem
pair
    A = (0,0), B=(3,0), C=(1,2);
draw(A--B--C--cycle);
//--- 3 chän duong cao ha tu 3 dinh, truc tam H
pair
    hC = extension(C,C+dir(90+degrees(A-B,false)),A,B),
    hB = extension(B,B+dir(90+degrees(A-C,false)),A,C),
    hA = extension(A,A+dir(90+degrees(C-B,false)),C,B),
    H = extension(A,hA,B,hB);
//----- Danh dau goc vuong
perpendicular(size=6,hC,NE,B-hC); perpendicular(size=6,hA,NW,C-hA);
perpendicular(size=6,hB,NE,A-hB);
//----- 3 Duong cao
draw(A--hA--B--hB--C--hC,dashed);
//-----
dot("$A$",A,W,to); dot("$B$",B,E,to); dot("$C$",C,N,to);
dot("$C'$",hC,S,to); dot("$H$",H,1.5*dir(120),to);
```



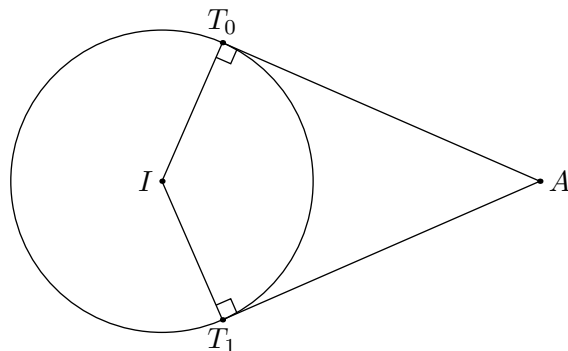
Code 73: Đường tròn nội tiếp tam giác

```
import markers; import math; import geometry; //-- Dung chuc nang goc vuong
unitsize(2cm);
dotfactor =4; //----- Dieu khien do day cua diem
//----- Cac diem
pair A=(-1,0), B=(2.5,0), C=(1.5,1.8);
//--- 3 diem tren 3 phan giac
pair pA= A+dir(A--C,A--B), pB= B+dir(B--A,B--C), pC=C+dir(C--A,C--B);
//-- Tam, ban kinh duong tron noi tiep
pair
    I = extension(A,pA,B,pB), // Tam noi tiep
    iC = extension(I,I+dir(90+degrees(A-B,false)),A,B), //-- 3 tiep diem tren 3 canh
    iB = extension(I,I+dir(90+degrees(A-C,false)),A,C),
    iA = extension(I,I+dir(90+degrees(C-B,false)),C,B);
real r=abs(I-iA); //--- Bkinh duong tron noi tiep r
//----- Danh dau goc A
frame fBo = stickframe(n=2,size=4,space=1.2,red);
marker mkBo = marker(markinterval(2,fBo,true));
markangle(2,radius=20,space=2,C,B,A,cyan,mkBo);
//----- Danh dau goc B
frame fAo = stickframe(n=1,size=4,magenta);
marker mkAo = marker(markinterval(2,fAo,true));
markangle(2,radius=20,space=1,B,A,C,blue,mkAo);
```

```

//--- Danh dau goc C
frame fCo = stickframe(n=3,size=4,space=1,magenta);
marker mkCo = marker(markinterval(2,fCo,true));
markangle(2,radius=15,space=1,A,C,B,blue,mkCo);
//----- Tam giac
draw(A--B--C--cycle); draw(A--I^^B--I^^C--I);
//--- Noi tam voi cac tiep diem
draw(I--iA,dashed);draw(I--iB,dashed); draw(I--iC,dashed);
//---- Danh dau goc vuong
real k = linewidth()/2; //-- do day cau path mac dinh tren 2
perpendicular(size=6,iC,NE,B-iC,green,TrueMargin(k,k));
perpendicular(size=6,iA,NE,C-iA,green,TrueMargin(k,k));
perpendicular(size=6,iB,NE,A-iB,green,TrueMargin(k,k));
//--- Duong tron noi tiep
draw(circle(I,r),red);
//--- Cac diem
dot("$A$",A,SSW); dot("$B$",B,SSE); dot("$C$",C,NE); dot("$I$",I,SSW);

```



Code 74: Tiếp tuyến của đường tròn tại điểm không thuộc được tròn

```

import geometry;
unitsize(1cm);
dotfactor =4;
pair I=(0,0), A=(5,0); //-- Tam I va diem A
real r = 2; //-- Ban kinh duong tron C
path ci = circle(I,r); draw(ci); // Duong tron C, tam I, bank kinh r
pair M=midpoint(I--A); //-- Trung diem IA == M=A/2+B/2 == M=interp(A,B,0.5)
//----- 2 tiep diem
pair [] T=intersectionpoints(ci,circle(M,abs(M-I)));
dot("$T_0$",T[0],N); dot("$T_1$",T[1],S);
//--- Noi cac tiep tuyen
draw(A--T[0]--I--T[1]--A);
//-----
perpendicular(size=6,T[0],NE,I-T[0]);
perpendicular(size=6,T[1],SE,I-T[1]);
//-----
dot("$I$",I,W); dot("$A$",A,E); //-- dot("$M$",M,E);

```

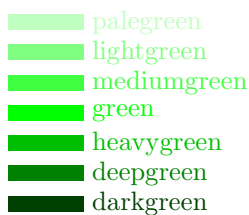


4. Màu – tô màu

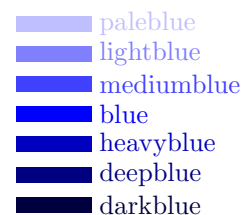
4.1. Màu sắc



Hình 3: Màu đỏ



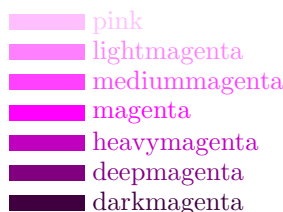
Hình 4: Màu Xanh lá cây



Hình 5: Màu xanh biển



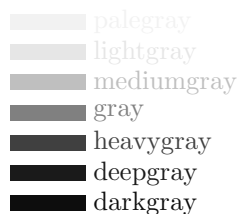
Hình 6: Màu cyan



Hình 7: Màu hồng



Hình 8: Màu vàng



Hình 9: Màu xám



Hình 10: Màu không có tiền tố 1



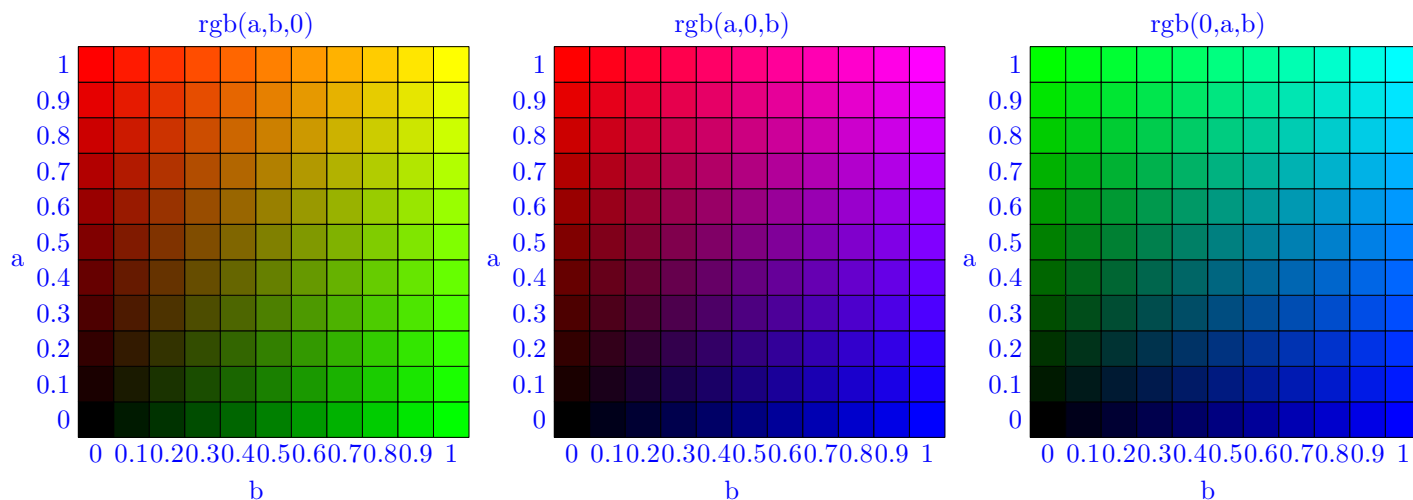
Hình 11: Màu không có tiền tố 2

Định nghĩa màu dựa vào các hệ màu hỗ trợ

Một số định nghĩa màu sắc được hỗ trợ trong Asymptote.

- `pen gray(real g)`; số `g` nằm trong đoạn $[0; 1]$
- `pen rgb(real r, real g, real b)`; các số `r`, `g`, `b` thuộc đoạn $[0; 1]$.
- `pen RGB(int r, int g, int b)`; với các số nguyên `r`, `g`, `b` thuộc đoạn $[0; 255]$.
- `pen cmyk(real c, real m, real y, real k)`; với các số thực `c`, `m`, `y`, `k` thuộc đoạn $[0; 1]$
- `pen hsv(int h, real s, real v)`; với số nguyên `h` thuộc đoạn $[0; 360]$ và `s`, `v` thuộc đoạn $[0; 1]$
- `pen invisible`; màu vô hình.

Một số mẫu màu minh học theo mã `rgb`



4.2. Tô màu

Để tô màu cho một `path` kín trong Asymptote, Mando dùng lệnh `fill` hoặc `filldraw` với cú pháp như sau

```
void fill(
    picture pic=currentpicture, // hình hiện tại
    path g, // path kín
    pen p=currentpen // màu cần tô
);
```

```
void filldraw(
    picture pic=currentpicture, // hình hiện tại
    path g, // path kín
    pen fillpen=currentpen, // Màu tô
    pen drawpen=currentpen // Màu của viền (path)
);
```

4.2.1. Đơn màu

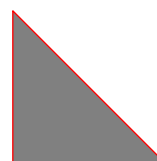
Code 75: Tô màu không viền



```
unitsize(1cm);
// -- path kín nói 3 điểm, cycle ở cuối
path p = (0,0)--(0,2)--(2,0)--cycle;
//-- Tô phần trong của path
fill(p,gray);
```

Code 76: Tô màu có viền

```
unitsize(1cm);
// -- path kín nói 3 điểm, cycle ở cuối
path p = (0,0)--(0,2)--(2,0)--cycle;
//-- tô màu gray, vẽ viền màu đỏ
filldraw(p,fillpen=gray,drawpen=red); // red+1pt sẽ tăng độ dày
của viền
```

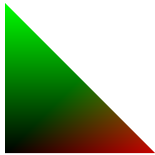


4.2.2. Nhiều màu

Pha trộn nhiều màu.

```
void latticeshade(  
    picture pic=currentpicture,  
    path g, // path kin  
    bool stroke=false, // mat dinh false, neu true thi chi ve vien  
    pen fillrule=currentpen, // luat to mau theo pen[][]  
    pen[][] p // mang mau 2 chieu  
);
```

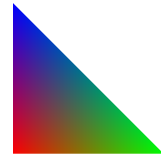
Code 77: Pha mảng 2 màu



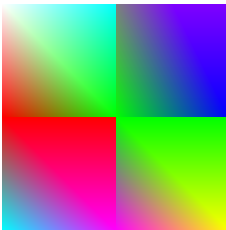
```
unitsize(1cm);  
/-- Mang mau 2 chieu  
pen[][] p = {{rgb(green)}, {rgb(.8red)}};  
/-- path kin  
path pA = (0,0)--(0,2)--(2,0)--cycle;  
/-- To mau  
latticeshade(pA, p);
```

Code 78: Pha mảng 4 màu

```
unitsize(1cm);  
/-- Mang mau 2 chieu, 4 mau  
pen[][] p={{rgb(blue),rgb(black)},  
            {rgb(red),rgb(green)}};  
/-- path kin  
path pA = (0,0)--(0,2)--(2,0)--cycle;  
/-- To mau  
latticeshade(pA, p);
```



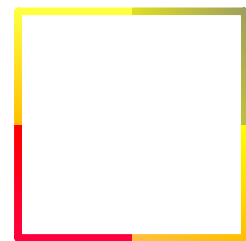
Code 79: Pha nhiều màu



```
unitsize(3cm);  
/-- Mang mau 2 chieu, 4 mau  
pen[][] p={{rgb(white),rgb(grey),rgb(black)},  
            {red,green,blue},  
            {cyan,magenta,yellow}};  
/-- path kin  
path pA = unitsquare;  
/-- To mau  
latticeshade(pA, p);
```

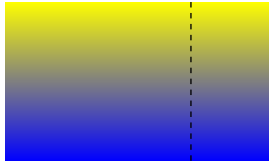
Code 80: Chỉ vẽ viền theo kiểu màu latticeshade

```
unitsize(3cm);  
/-- Mang mau 2 chieu, 4 mau  
pen[][] p={{rgb(white),rgb(grey),rgb(black)},  
            {red,yellow,blue},  
            {cyan,magenta,red}};  
/-- path kin  
path pA = unitsquare;  
/-- To mau  
latticeshade(pA, stroke=true, linewidth(3bp), p);
```



4.2.3. Màu kiểu tịnh tiến

```
void axialshade(
    picture pic=currentpicture,
    path g, // path kín
    bool stroke=false, //
    pen pena, pair a, bool extenda=true,
    pen penb, pair b, bool extendb=true
);
```



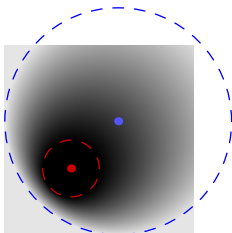
Hình 12: Tô màu theo phương là đường thẳng

```
unitsize(1cm);
pen p1=blue, p2=yellow;
path g = box((0,0),(5,3));
pair pA = (3.5,0), pB=(3.5,3);
axialshade(g,p1,pA,p2,pB);
draw(pA--pB,linetype("4 4"));
```

4.2.4. Màu kiểu trung tâm

```
void radialshade(
    picture pic=currentpicture,
    path g, // path kín cần tô màu
    bool stroke=false, // kho to viền
    pen penA, pair A, real rA, // Màu, tam, bán kính tô màu trung tâm
    bool extenda=true,
    pen penB, pair B, real rB, // Màu, tam, bán kính tô màu xung quanh
    bool extendb=true);
```

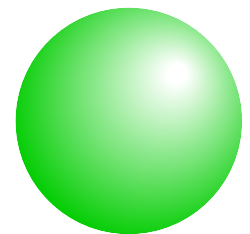
Code 81: Minh họa tô màu kiểu radialshade



```
unitsize(2.5cm);
//----- Diem A, B
pair A=(0.35,0.35), B=(0.6,0.6);
//----- Tô màu hình vuông đơn vị
radialshade(
    unitsquare, // Tô màu hình vuông
    black,A,0.15, // Màu đen tam A, bk 0.15
    lightgrey,B,.6); // Màu lightgrey tam B, bk 0.6
//----- Hien tam A, B
dot(A,.8red); dot(B,blue+grey);
//----- 2 Đường tròn
draw(circle(A,0.15),dashed+.8red);
draw(circle(B,0.6),dashed+blue);
```

Code 82: Tạo giả cầu

```
unitsize(1.3cm);
pair O=0; // Tâm đường tròn
real R = 1, rA=0.1, rB=1.2;
//-----
radialshade(
    circle(O,R),
    white,0.6*dir(45), rA, // tam A =(0.6*cos(45),0.6*sin(45))
    0.8*green,0.2*dir(45),rB ); // tam B =(0.6*cos(45),0.6*sin(45))
```

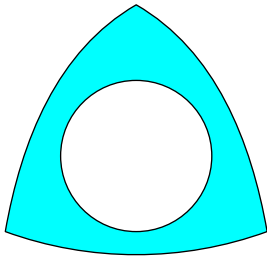


4.2.5. Tô màu miền giao nhau - evenodd

Giả sử ta có nhiều miền kín giao nhau(các path kín giao nhau) như vậy sẽ có các miền giao là giao của

số lẻ hoặc số chẵn các path. Ta cần tô màu cho miền giao là giao của số lẻ path thì sử dụng lệnh `evenodd`.

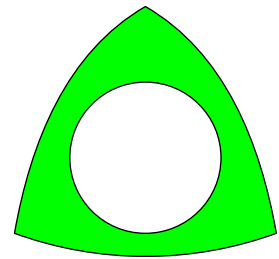
Code 83: Tô màu cho miền giao



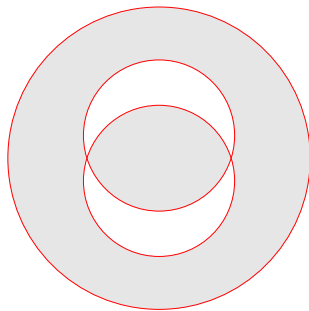
```
unitsize(2cm);
//--- Duong tron tam O, bk 0.5
path c = circle(0,0.5);
pair A = dir(-30), B = dir(90), C = dir(210);
//---- path kin tao boi 3 diem A,B,C
path
    p = A{dir(100)}
      ..{dir(150)}B{dir(210)}
      ..{dir(-100)}C{dir(-20)}
      ..{dir(20)}A..cycle;
filldraw(p^^c, evenodd+cyan, black);
```

Code 84: Tô màu cho miền giao

```
unitsize(2cm);
//--- Duong tron tam O, bk 0.5
path c = circle(0,0.5);
pair A = dir(-30), B = dir(90), C = dir(210);
//---- path kin tao boi 3 diem A,B,C
path
    p = A{dir(100)}
      ..{dir(150)}B{dir(210)}
      ..{dir(-100)}C{dir(-20)}
      ..{dir(20)}A..cycle;
filldraw(p^^reverse(c), green, black);
```



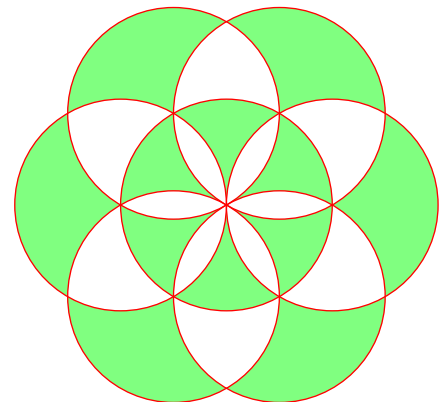
Code 85: Tô màu giao của 3 path

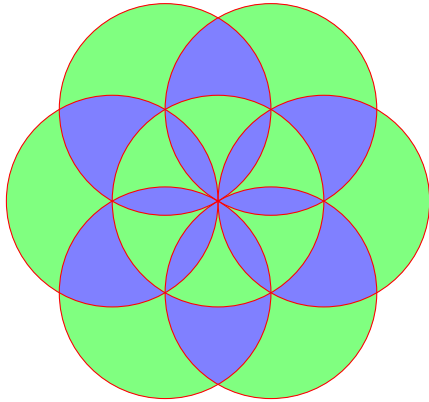


```
unitsize(1cm);
//--- Duong tron o ngoai
path cI = circle(0,2);
//--- Duong tron nho, o trong phia tren
path cT = circle((0,0.3),1);
//--- Duong tron nho, o trong phia duoi
path cD = circle((0,-0.3),1);
filldraw(
    cI^^cT^^cD, // 3 duong tron
    evenodd+gray, // chuc nang evenodd
    red+linewidth(0.2pt) );
```

Code 86: Tô màu giao của 7 path

```
unitsize(2cm);
//--- Duong tron o giua
path cI = circle(0,1); // draw(cI);
//--- 6 duong tron xung quanh
path c[];
for (int i=0; i<=5; ++i) {
    c[i] = circle(dir(i*60),1); }
//----- To mau kieu evenodd
filldraw(
    cI^^c[0]^c[1]^c[2]^c[3]^c[4]^c[5],
    evenodd+lightgreen, red);
```

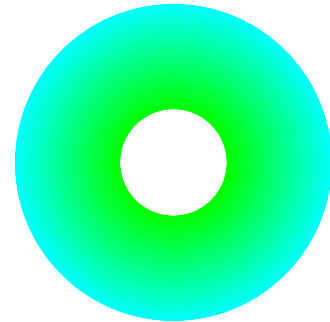




Code 87: Tô màu giao của 7 path 2

```
unitsize(2cm);
path cI = circle(0,1); // Duong tron giua
//--- 6 duong tron xung quanh
path c[];
for (int i =0; i<=5; ++i) {
  c[i] = circle(dir(i*60),1);}
//-- To mau blue cho mien giao chan
fill(cI^^c[0]^c[1]^c[2]^c[3]^c[4]^c[5],
      zerowinding+lightblue);
//-- To mau green cho mien giao le
filldraw(cI^^c[0]^c[1]^c[2]^c[3]^c[4]^c[5],
          evenodd+lightgreen,red);
```

```
unitsize(1cm);
real ri=1, ro=3;
path
  in=circle(0,ri), // duong tron tam 0, bk 1
  out=circle(0,ro); // Duong tron tam 0, bk 3
radialshade(
  in^^out, // Hình vành khuyên
  green+evenodd, // luật chan le
  (0,0),ri,cyan,(0,0),ro); // d/tron lon xanh, d/
  tron nho do
```



4.2.6. Tô màu cho text

Dùng hàm `axialshade` và hàm `min`, `max`.

- `pair min(path p)`; điểm bên trái thấp nhất của hình chữ nhật chứa `path p`.
- `pair max(path p)`; điểm bên phải cao nhất của hình chữ nhật chứa `path p`.
- `path[] texpath(Label L)`; Mảng `path` là các ký tự (string) trong `Label L`.

$$\int_{-\infty}^{+\infty} e^{-\alpha x^2} dx = \sqrt{\frac{\pi}{\alpha}}$$

```
unitsize(1cm);
path[] p=texpath("$\displaystyle\int_{-\infty}^{+\infty}e^{-\alpha x^2}\,dx= \sqrt{\frac{\pi}{\alpha}}$");
pair m=min(p), M=max(p);
//-- m.x: hoành độ của điểm m, M.y tung độ của điểm M
axialshade(p,yellow,m,red,(m.x,M.y));
draw(p,blue); // mau của vien text
```



Code 88: Kết hợp latticeshade cho viền và text

```
unitsize(1cm);
//--- Text
path[] p=tepath("\large \bf 2020");
transform k = scale(0.3);
//--- Mau co text
pen [] cola ={{white,gray,yellow},{red,green,blue},{cyan, magenta,yellow}};
latticeshade(k*p,cola);
//---- Mau cho vien
pen [] colb={{red,blue,cyan},{cyan,white,green}};
latticeshade(k*p,stroke=true,linewidth(2bp),colb);
```

5. Phép biến hình

6. Pen – font chữ

6.1. Pen

Kiểu dữ liệu **pen** bao gồm: màu sắc, kiểu **path**(solid, dashed, dotted,...), độ dày **path**(linewidth()), font chữ

Trong kiểu dữ liệu **pen**: Ta có thể dùng dấu + các yếu tố trên với nhau.

```
defaultpen(font + fontsize(11pt) + linewidth(3pt)+red);
```

Mặc định **size** chữ 11pt, độ dày của **path** 3pt, màu của **path** đỏ.

10 15 20 25 30 35 40 45

10 15 20 25 30 35 40 45

Code 89: Không có modulo fontsize

```
unitsize(1cm);
defaultpen(fontsize(10pt));
label("$10$", (0,0),blue);
label("$15$", (1,0),fontsize(15pt)+red);
label("$20$", (2,0),fontsize(20pt)+green);
label("$25$", (3,0),fontsize(25pt)+blue);
label("$30$", (4,0),fontsize(30pt)+red);
label("$35$", (5,0),fontsize(35pt)+green);
label("$40$", (6,0),fontsize(40pt)+blue);
label("$45$", (7,0),fontsize(45pt)+red);
```

Code 90: Thêm modulo fontsize

```
import fontsize; unitsize(1cm);
defaultpen(fontsize(10pt));
label("$10$", (0,0),blue);
label("$15$", (1,0),fontsize(15pt)+red);
label("$20$", (2,0),fontsize(20pt)+green);
label("$25$", (3,0),fontsize(25pt)+blue);
label("$30$", (4,0),fontsize(30pt)+red);
label("$35$", (5,0),fontsize(35pt)+green);
label("$40$", (6,0),fontsize(40pt)+blue);
label("$45$", (7,0),fontsize(45pt)+red);
```

		fontsize(16pt)+orange			
		fontsize(14pt)+blue			
		fontsize(12pt)+red			
		fontsize(10pt)+green			
		fontsize(8pt)+purple			

```
import math;
unitsize(1cm);
add(grid(6,6,lightgray)); // Tao luoi toa do
label("fontsize(8pt)+purple", (3,1),fontsize(8pt)+purple);
label("fontsize(10pt)+green", (3,2),fontsize(10pt)+green);
label("fontsize(12pt)+red", (3,3),fontsize(12pt)+red);
label("fontsize(14pt)+blue", (3,4),fontsize(14pt)+blue);
label("fontsize(16pt)+orange", (3,5),fontsize(16pt)+orange);
```

6.2. Font chữ

Các font chữ hỗ trợ: [AvantGarde](#), [Bookman](#), [Courier](#), [Helvetica](#), [NewCenturySchoolBook](#), [Palatino](#), [TimesRoman](#), [ZapfChancery](#), [Symbol](#). Các font chữ này không hỗ trợ tiếng Việt.

Font [AvantGarde](#)

AvantGarde
 ABCDEFGHIJK
 abcdefghijk
 0123456789

Code 91: Thường

```
unitsize(1cm,0.5cm);
pen p=AvantGarde(series="m", shape="n");
string s[] ={
  "AvantGarde", "ABCDEFGHIJK",
  "abcdefghijk", "0123456789" };
for (int i=0; i<s.length; ++i){
  if(i==0) label(s[i],(0,-i),p + red);
  else label(s[i],(0,-i),p); }
```

AvantGarde
 ABCDEFGHIJK
 abcdefghijk
 0123456789

AvantGarde
 ABCDEFGHIJK
 abcdefghijk
 0123456789

Code 92: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=AvantGarde(series="m", shape="it");
string s[] ={
  "AvantGarde", "ABCDEFGHIJK",
  "abcdefghijk", "0123456789" };
for (int i=0; i<s.length; ++i){
  if(i==0) label(s[i],(0,-i),p + red);
  else label(s[i],(0,-i),p); }
```

AvantGarde
 ABCDEFGHIJK
 abcdefghijk
 0123456789

Code 93: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=AvantGarde(series="b", shape="n");
string s[] ={
    "AvantGarde","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 94: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=AvantGarde(series="b", shape="it");
string s[] ={
    "AvantGarde","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Font Bookman

Bookman
ABCDEFGHIJK
abcdefghijkl
0123456789

Bookman
ABCDEFGHIJK
abcdefghijkl
0123456789

Code 95: Thường

```
unitsize(1cm,0.5cm);
pen p=Bookman(series="m", shape="n");
string s[] ={
    "Bookman","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 96: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Bookman(series="m", shape="it");
string s[] ={
    "Bookman","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Bookman
ABCDEFGHIJK
abcdefghijkl
0123456789

Bookman
ABCDEFGHIJK
abcdefghijkl
0123456789

Code 97: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=Bookman(series="b", shape="n");
string s[] ={
    "Bookman","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 98: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Bookman(series="b", shape="it");
string s[] ={
    "Bookman","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Font Courier

Courier
ABCDEFGHIJK
abcdefghijkl
0123456789

Courier
ABCDEFGHIJK
abcdefghijkl
0123456789

Code 99: Thường

```
unitsize(1cm,0.5cm);
pen p=Courier(series="m", shape="n");
string s[] ={
    "Courier","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Courier
ABCDEFGHJK
abcdefghijk
0123456789

Code 100: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Courier(series="m", shape="it");
string s[] ={
    "Courier","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Courier
ABCDEFGHJK
abcdefghijk
0123456789

Code 101: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=Courier(series="b", shape="n");
string s[] ={
    "Bookman","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Helvetica
ABCDEFGHJK
abcdefghijk
0123456789

Code 102: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Courier(series="b", shape="it");
string s[] ={
    "Courier","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Helvetica
ABCDEFGHJK
abcdefghijk
0123456789

Font Helvetica

Code 103: Thường

```
unitsize(1cm,0.5cm);
pen p=Helvetica(series="m", shape="n");
string s[] ={
    "Helvetica","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Helvetica
ABCDEFGHJK
abcdefghijk
0123456789

Code 104: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Helvetica(series="m", shape="it");
string s[] ={
    "Helvetica","ABCDEFGHJK",
    "abcdefghijk","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Helvetica
ABCDEFGHJK
abcdefghijk
0123456789

Code 105: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=Helvetica(series="b", shape="n");
string s[] ={
    "Helvetica","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 106: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=Helvetica(series="b", shape="it");
string s[] ={
    "Helvetica","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Font NewCenturySchoolBook

NewCenturySchoolBook

ABCDEF GHIJK
abcdefghijkl
0123456789

NewCenturySchoolBook

ABCDEF GHIJK
abcdefghijkl
0123456789

Code 107: Thường

```
unitsize(1cm,0.5cm);
pen p=NewCenturySchoolBook(series="m", shape="n");
string s[] ={
    "NewCenturySchoolBook","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 108: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=NewCenturySchoolBook(series="m", shape="it");
string s[] ={
    "NewCenturySchoolBook","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

NewCenturySchoolBook

ABCDEF GHIJK
abcdefghijkl
0123456789

NewCenturySchoolBook

ABCDEF GHIJK
abcdefghijkl
0123456789

Code 109: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=NewCenturySchoolBook(series="b", shape="n");
string s[] ={
    "NewCenturySchoolBook","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Code 110: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=NewCenturySchoolBook(series="b", shape="it");
string s[] ={
    "NewCenturySchoolBook","ABCDEFGHIJK",
    "abcdefghijkl","0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Font Palatino

Palatino
ABCDEFGHJK
abcdefghijk
0123456789

Code 111: Thường

```
unitsize(1cm,0.5cm);  
pen p=Palatino(series="m", shape="n");  
string s[] ={"Palatino", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

Palatino
ABCDEFGHJK
abcdefghijk
0123456789

Code 112: Thường, nghiêng

```
unitsize(1cm,0.5cm);  
pen p=Palatino(series="m", shape="it");  
string s[] ={"Palatino", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

Palatino
ABCDEFGHJK
abcdefghijk
0123456789

Code 113: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);  
pen p=Palatino(series="b", shape="n");  
string s[] ={"Palatino", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

Palatino
ABCDEFGHJK
abcdefghijk
0123456789

Code 114: Dậm, nghiêng

```
unitsize(1cm,0.5cm);  
pen p=Palatino(series="b", shape="it");  
string s[] ={"Palatino", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

Font TimesRoman

TimesRoman
ABCDEFGHJK
abcdefghijk
0123456789

Code 115: Thường

```
unitsize(1cm,0.5cm);  
pen p=TimesRoman(series="m", shape="n");  
string s[] ={"TimesRoman", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

TimesRoman
ABCDEFGHJK
abcdefghijk
0123456789

Code 116: Thường, nghiêng

```
unitsize(1cm,0.5cm);  
pen p=TimesRoman(series="m", shape="it");  
string s[] ={"TimesRoman", "ABCDEFGHIJK",  
             "abcdefghijk", "0123456789" };  
for (int i=0; i<s.length; ++i){  
    if(i==0) label(s[i],(0,-i),p + red);  
    else label(s[i],(0,-i),p); }
```

TimesRoman
ABCDEFGHJK
abcdefghijkl
0123456789

Code 117: Dậm, thẳng đứng

```
unitsize(1cm,0.5cm);
pen p=TimesRoman(series="b", shape="n");
string s[] ={
    "TimesRoman", "ABCDEFGHJK",
    "abcdefghijkl", "0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

TimesRoman
ABCDEFGHJK
abcdefghijkl
0123456789

Code 118: Dậm, nghiêng

```
unitsize(1cm,0.5cm);
pen p=TimesRoman(series="b", shape="it");
string s[] ={
    "TimesRoman", "ABCDEFGHJK",
    "abcdefghijkl", "0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Font ZapfChancery

ZapfChancery
ABCDEFGHJK
abcdefghijkl
0123456789

Code 119: Thường

```
unitsize(1cm,0.5cm);
pen p=ZapfChancery(series="m", shape="n");
string s[] ={
    "ZapfChancery", "ABCDEFGHJK",
    "abcdefghijkl", "0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

ZapfChancery
ABCDEFGHJK
abcdefghijkl
0123456789

Code 120: Thường, nghiêng

```
unitsize(1cm,0.5cm);
pen p=ZapfChancery(series="m", shape="it");
string s[] ={
    "ZapfChancery", "ABCDEFGHJK",
    "abcdefghijkl", "0123456789" };
for (int i=0; i<s.length; ++i){
    if(i==0) label(s[i],(0,-i),p + red);
    else label(s[i],(0,-i),p); }
```

Asymptote: The Vector Graphics Language

Code 121: Tiêu đề tài liệu

```
unitsize(0.6cm);
//-----
defaultpen(AvantGarde(series="b", shape="n")); //-- Mac dinh font chu
path[] p=texpath("\large \bf \centering Asymptote: \\ The Vector Graphics Language");
//--- Mau co text
pen [][] cola={{white,gray,yellow},{red,green,blue},{cyan, magenta,yellow}};
latticeshade(scale(0.1)*p,cola);
//---- Mau cho vien
pen [][] colb={{red,blue,lightgreen},{cyan,white,green}};
latticeshade(scale(0.1)*p,stroke=true,linewidth(0.8bp),colb);
```


7. Nền – patterns

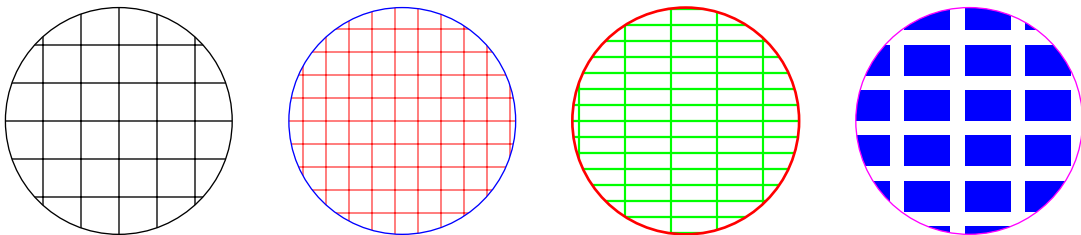
modul `patterns` bổ sung thêm một số kiểu nền để tô miền kín. Để sử dụng modul này, Mando thêm dòng lệnh `import patterns;` ở đầu tập tin vẽ hình.

Cú pháp chung để tạo 1 kiểu nền và để tô nền Mando sử dụng lệnh `fill` hoặc `filldraw` và sử dụng giống như lệnh tô miền kín.

```
void add(  
    string name, //-- Dat ten cho kieu nen  
    picture pic, //-- Mot trong cac kieu nen ma asy ho tro  
    pair lb=0, pair rt=0 //-- dich chuyen vi tri pattern  
);
```

7.1. Kiểu `tile` – lưới

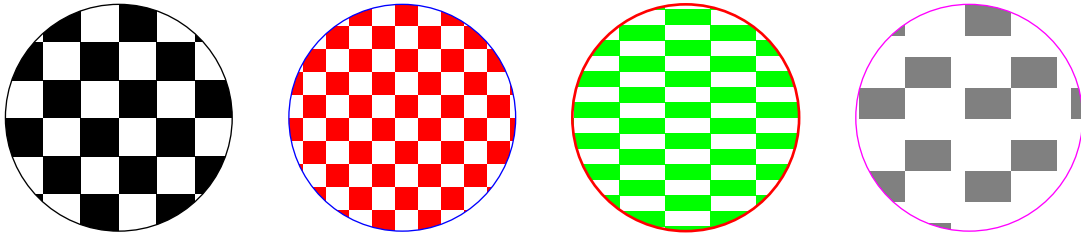
```
picture tile(  
    real Hx=5mm, //-- o luoi canh 5mm  
    real Hy=0, // -- Hy=0 thi o luoi la hình vuông, Hy khác 0 thi o luoi la hình chu nha  
    pen p=currentpen, //-- Mau luoi (Mac dinh black)  
    filltype filltype=NoFill //-- To mau, mac dinh la khong to  
);
```



```
import patterns;  
unitsize(1.5cm);  
/-- Duong tron - path kín  
path cA = circle(0,1), cB=shift(2.5,0)*cA, cC=shift(5,0)*cA, cD=shift(7.5,0)*cA;  
/-- Tao pattern, kieu tile mac dinh  
add("TiA",tile());  
filldraw(cA,pattern("TiA"));  
/-- tile co tuy chon B  
add("TiB",tile(3mm,red+0.2pt)); //--rut gon cua tile(Hx=3mm,Hy=0,p=red+0.2pt)  
filldraw(cB,pattern("TiB"),blue);  
/-- tile co tuy chon C  
add("TiC",tile(6mm,2mm,green+0.8pt)); //-- rut gon cua tile(Hx=6mm,Hy=2mm,p=green+0.8pt)  
filldraw(cC,pattern("TiC"),red+1bp);  
/-- tile co tuy chon  
add("TiD",tile(6mm,4mm,blue,Fill),(1mm,1mm),(1mm,1mm));/--  
filldraw(cD,pattern("TiD"),magenta);
```

7.2. Kiểu `checker` – ô caro

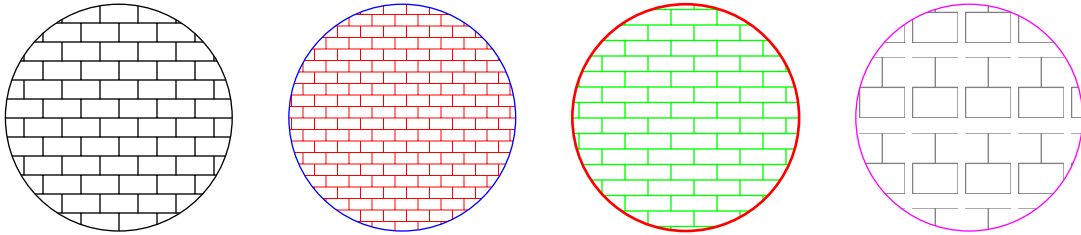
```
picture checker(  
    real Hx=5mm, //-- Canh o vuông 5mm  
    real Hy=0, //-- Hy=0 thi o la hình vuông  
    pen p=currentpen //-- mau  
);
```



```
import patterns;
unitsize(1.5cm);
//--- Duong tron - path kin
path cA = circle(0,1), cB=shift(2.5,0)*cA, cC=shift(5,0)*cA, cD=shift(7.5,0)*cA;
//--- Tao pattern, kieu tile mac dinh
add("ChA",checker());
filldraw(cA,pattern("ChA"));
//--- tile co tuy chon B
add("ChB",checker(3mm,red+0.2pt));
filldraw(cB,pattern("ChB"),blue);
//-- tile co tuy chon C
add("ChC",checker(6mm,2mm,green+0.5pt));
filldraw(cC,pattern("ChC"),red+1bp);
//---
add("ChD",checker(6mm,4mm,gray),(1mm,2mm),(1mm,1mm));!--
filldraw(cD,pattern("ChD"),magenta);
```

7.3. Kiểu brick – lát gạch

```
picture brick(
    real Hx=5mm,
    real Hy=0,
    pen p=currentpen
);
```

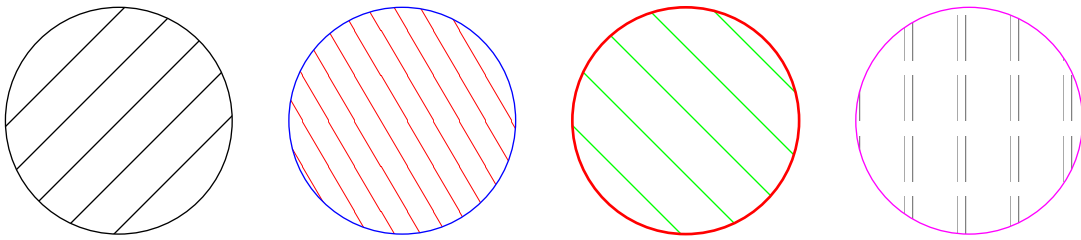


```
import patterns;
unitsize(1.5cm);
//--- Duong tron - path kin
path cA = circle(0,1), cB=shift(2.5,0)*cA, cC=shift(5,0)*cA, cD=shift(7.5,0)*cA;
//--- Tao pattern, kieu tile mac dinh
add("BrA",brick());
filldraw(cA,pattern("BrA"));
//--- tile co tuy chon B
add("BrB",brick(3mm,red+0.2pt));
filldraw(cB,pattern("BrB"),blue);
//-- tile co tuy chon C
add("BrC",brick(6mm,2mm,green+0.5pt));
filldraw(cC,pattern("BrC"),red+1bp);
//---
```

```
add("BrD", brick(6mm, 4mm, gray), (1mm, 2mm)); //--
filldraw(cD, pattern("BrD"), magenta);
```

7.4. Kiểu hatch – gạch sọc

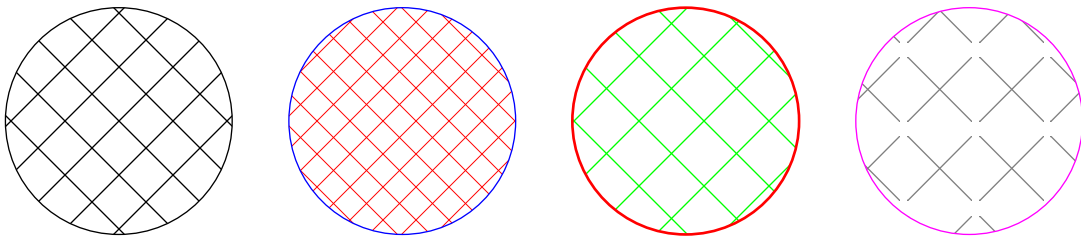
```
picture hatch(
    real H=5mm, //-- Khoảng cách giữa các đoạn thẳng song song
    pair dir=NE, //-- Phương của các đường thẳng. Mặc định hướng 45 độ, có thể dùng dir(angle)
    hoac 1 vecto để chỉ phương
    pen p=currentpen //-- Màu
);
```



```
import patterns;
unitsize(1.5cm);
/---- Đường tròn - path kín
path cA = circle(0,1), cB=shift(2.5,0)*cA, cC=shift(5,0)*cA, cD=shift(7.5,0)*cA;
/---- Tạo pattern, kiểu tile mặc định
add("HaA", hatch());
filldraw(cA, pattern("HaA"));
/---- tile có tùy chọn B
add("HaB", hatch(3mm, dir(-60), red+0.2pt)); //-- hatch(Hx=3mm, dir=dir(-60), p=red+0.2pt)
filldraw(cB, pattern("HaB"), blue);
/---- tile có tùy chọn C
add("HaC", hatch(6mm, NW, green+0.5pt)); //-- hatch(Hx=6mm, dir=NW, p=green+0.5pt)
filldraw(cC, pattern("HaC"), red+1bp);
/----
add("HaD", hatch(6mm, N, gray), (1mm, 2mm)); //--
filldraw(cD, pattern("HaD"), magenta);
```

7.5. Kiểu crosshatch – 2 đường chéo

```
crosshatch(real H=5mm, pen p=currentpen)
```



```
import patterns;
unitsize(1.5cm);
/---- Đường tròn - path kín
```

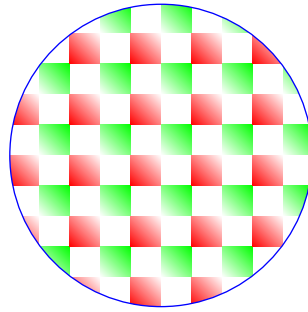
```

path cA = circle(0,1), cB=shift(2.5,0)*cA, cC=shift(5,0)*cA, cD=shift(7.5,0)*cA;
//--- Tao pattern, kieu tile mac dinh
add("CrA",crosshatch());
filldraw(cA,pattern("CrA"));
//--- tile co tuy chon B
add("CrB",crosshatch(3mm,red+0.2pt)); //--- hatch(Hx=3mm,p=red+0.2pt)
filldraw(cB,pattern("CrB"),blue);
//--- tile co tuy chon C
add("CrC",crosshatch(6mm,green+0.5pt)); //--- hatch(Hx=6mm,p=green+0.5pt)
filldraw(cC,pattern("CrC"),red+1bp);
//---
add("CrD",crosshatch(6mm,gray),(1mm,2mm)); //---
filldraw(cD,pattern("CrD"),magenta);

```

Tạo một pattern đẹp

Bây giờ Mando sẽ vẽ hình tròn đơn vị với mỗi ô vuông cạnh bằng $\frac{1}{5}$ đơn vị với kiểu tô màu `axialshade`.



```

import patterns;
unitsize(2cm);
real d=2/5*cm; //--- Do dai don vi chi 5
//--- Ve 2 hình vuông Xanh, Do
path SqX=scale(d)*unitsquare, SqD=shift(d,d)*SqX;
//--- Do dai canh hình vuông = do dai canh chia 5
picture pic; //--- Tao 2 picture voi gom 2 hình vuông //--- Dung ham axialshade
axialshade(pic,SqX,white,(0,0),green,(d,d));
axialshade(pic,SqD,white,(2d,2d),red,(d,d));
//--- Tao pattern
add("BackGround",pic);
filldraw(unitcircle,pattern("BackGround"),blue); //--- To hình tròn đơn vị

```