TP2_exemples

```
K.<a>=GF(3^3)
```

```
a^4
```

```
a^2 + 2*a
```

```
# a quelle type de structure appartient a ?
a.parent()
```

```
Finite Field in a of size 3^3
```

```
#liste des elements de K
for s in K:
    print s
```

```
0
a
a^2
a + 2
a^2 + 2*a
2*a^2 + a + 2
a^2 + a + 1
a^2 + 2*a + 2
2*a^2 + 2
a + 1
a^2 + a
a^2 + a + 2
a^2 + 2
2
2*a
2*a^2
2*a + 1
2*a^2 + a
a^2 + 2*a + 1
2*a^2 + 2*a + 2
2*a^2 + a + 1
a^2 + 1
2*a + 2
2*a^2 + 2*a
2*a^2 + 2*a + 1
2*a^2 + 1
1
```

```
#polynome minimal dans F_3[X]
f=a.minpoly('X')
print f
```

```
X^3 + 2*X + 1
```

```
# A quelle structure appartient f ?
f.parent()
```

```
Univariate Polynomial Ring in X over Finite Field of size 3
```

```
#les coef de X,X^3 et X^5 dans f
f[1],f[3],f[5]
```

```
(2, 1, 0)
```

```
# la liste des coefficients de f
f.coefficients(sparse=False)
```

```
[1, 2, 0, 1]
```

```
#la liste des coefs avec des eventuels zeros additionnels
jusqu'au degre 10
f.padded_list(10)
```

```
[1, 2, 0, 1, 0, 0, 0, 0, 0, 0]
```

```
f.is_irreducible()
```

```
True
```

```
R.<X>=PolynomialRing(K,'X')
```

```
g=1+a*X^2+2*X^3
h=g*(X+1)
```

```
g.parent()
```

```
Univariate Polynomial Ring in X over Finite Field in a of siz
```

```
g[2]
```

```
a
```

```
#un quotient est une fraction rationnelle, meme si g divise h
quo=h/g
quo.parent()
```

```
Fraction Field of Univariate Polynomial Ring in X over Finite
in a of size 3^3
```

```
#une fraction rationnelle n'a pas de degre
quo.degree()
```

```
Traceback (click to the left of this block for traceback)
...
AttributeError: 'FractionFieldElement_1poly_field' object has
attribute 'degree'
```

```
# pour obtenir le quotient en tant que polynome, utiliser la
division euclidienne quo_rem (on obtient le quotient et le
reste en meme temps)
(quo2,rem2) =h.quo_rem(g)
print quo2.parent()
print quo2.degree()
```

```
Univariate Polynomial Ring in X over Finite Field in a of siz
1
```

```
## ici, l'espace des matrices auquel M appartient est
```

```
implicite.
M=Matrix([[1,a,a],[a,a^2,a^4],[1,2,3]])
M.parent()
```

```
Full MatrixSpace of 3 by 3 dense matrices over Finite Field i
size 3^3
```

```
# voici un espace de matrices explicite
MM=MatrixSpace(K,2,2)
```

```
#N matrice explicitement a coef dans K
N=MM([[1,2],[3,4]])
print N
print N.parent()
```

```
[1 2]
[0 1]
Full MatrixSpace of 2 by 2 dense matrices over Finite Field i
size 3^3
```

```
#N2 implicitement a coef entiers
N2=Matrix([[1,2],[3,4]])
print N2
print N2.parent()
```

```
[1 2]
[3 4]
Full MatrixSpace of 2 by 2 dense matrices over Integer Ring
```

```
print N.inverse()
print N2.inverse()
```

```
[1 1]
[0 1]
[  -2    1]
[ 3/2 -1/2]
```

```
M=MM([[1,2],[2,4]])
```

```
#noyau
M.right_kernel()
```

```
Vector space of degree 2 and dimension 1 over Finite Field in
size 3^3
Basis matrix:
[1 1]
```

```
#base du noyau
b=M.right_kernel().basis()
print b
```

```
[
(1, 1)
]
```

```
# b est une liste, iteration sur les vecteurs de b
for v in b:
```

```
    print v
```

```
(1, 1)
```

```
#travail avec des listes
l=[]
```

```
#on peut ajouter un element avec append
l.append(1)
l.append('coucou')
print l
```

```
[1, 'coucou']
```

```
#on peut additionner 2 listes pour les concatener
l=l+[5,6,7]
print l
```

```
[1, 'coucou', 5, 6, 7]
```

```
# les elements de la liste sont numerotes a partir 0
print l[0]
print l[1]
```

```
1
coucou
```

```
# la longueur de la liste est len(l)
print len(l)
```

```
5
```

```
#attention:
l.append([3,4,5])
print l
```

```
[1, 'coucou', 5, 6, 7, [3, 4, 5]]
```

```
#enlever le dernier element:
a=l.pop()
print a
print l
```

```
[3, 4, 5]
[1, 'coucou', 5, 6, 7]
```

```
a=l.pop()
print a
print l
```

```
7
[1, 'coucou', 5, 6]
```

```
#boucle sur les elements de la liste:
for a in l:
    print a
```

```
1
coucou
5
```