
Groupe A : Nuages de points

Programmes à construire

```
c = 3.2
```

```
def liste_de_points (n, x) :  
    ListePoints = []  
    for k in range(n) :  
        ListePoints.append((k,x))  
        x = c*x*(1-x)  
    return ListePoints
```

```
def nuage_de_points (n, x) :  
    Graphique = points(liste_de_points (n, x), ymin=0, ymax=1, color='blue')  
    show(Graphique)
```

```
def nuage_de_points_double (n, x, y) :  
    Graphique = points(liste_de_points (n, x), ymin=0, ymax=1, color='blue')  
    Graphique += points(liste_de_points (n, y), ymin=0, ymax=1, color='red')  
    show(Graphique)
```

```
nuage_de_points_double(50, 0.3, 0.5)
```

Groupe B : Valeurs précises de la suite

Programmes à construire

```
RealNumber = RealField(500)
```

```
c = 2
```

```
def suite_de_valeurs (n ,x) :  
    for k in range(n):  
        print(x)  
        x = c*x*(1-x)
```

```
def suite_de_valeurs_texte (n ,x) :  
    fichier_texte = open('ValeursSuite.txt', 'w')  
    for k in range(n):  
        fichier_texte.write(x)  
        x = c*x*(1-x)  
    fichier_texte.close()
```

```
suite_de_valeurs_texte(500, 0.1)
```

Groupe C : Histogramme

Programmes à construire

```
c = 3.2
```

```
def liste_de_valeurs (n, x) :  
    ListeValeurs = []  
    for k in range(n):  
        ListeValeurs.append(x)  
        x = c*x*(1-x)  
    return ListeValeurs
```

```
def suite_histogramme (n, x) :  
    Graphique = histogram(liste_de_valeurs(n,x), bins=1000, density=true, range=[0,1])  
    show(Graphique)
```

```
suite_histogramme(10000, 0.2)
```

Groupe D : Toile d'araignée

Programmes à construire

```
c = 3.2

def liste_de_points (n ,x) :
    ListePoints = [(x,0)]
    for k in range(n):
        y = x
        x = c*x*(1-x)
        ListePoints.append((y, y))
        ListePoints.append((y, x))
    return ListePoints

def fonction_identite (x) :
    return x

def fonction_quadratique (x) :
    return c*x*(1-x)

def toile_araignee (n ,x) :
    Graphique = plot(fonction_identite, color='black')
    Graphique += plot(fonction_quadratique, color='blue')
    Graphique += line(liste_de_points (n ,x), xmin=0, xmax=1, ymin=0, ymax=1,
        aspect_ratio=1, color='red')
    show(Graphique)

toile_araignee(20, 0.1)
```

Groupe E : Algorithme de seuil

Programmes à construire

```
RealNumber = RealField(500)
```

```
c = 2.7
```

```
def algorithme_de_seuil (eps, x) :  
    for k in range(10000) :  
        y = c*x*(1-x)  
        if abs(x-y) < eps :  
            print('Nombre d\'iterations : ' + str(k))  
            print('Valeur atteinte : ' + str(y))  
            return  
        x = y  
    print('Le seuil de precision n\'a pas ete atteint')
```

```
algorithme_de_seuil(10-50, 0.1)
```
