

#### ./a-aire-disque.py

```
import math

def principale():
    # initialisation de l'état du système
    rayon = 1

    while rayon < 5:
        # affichage de l'état du système
        aire_disque = math.pi * rayon**2
        print("L'aire vaut", aire_disque, "pour r =", rayon)

        # mise à jour de l'état du système
        rayon = rayon + 0.03

principale()
```

#### ./b-diag.py

```
import math

def principale():
    # initialisation de l'état du système
    largeur = 1
    longueur = 1

    while longueur < 5:
        # affichage de l'état du système
        diag = (largeur**2 + longueur**2)**0.5
        print("La diagonale est", diag)

        # mise à jour de l'état du système
        largeur = largeur + 0.03
        longueur = longueur + 0.09

principale()
```

#### ./c-disque-aff.py

```
import math
from qtido import *

def principale():
    f = creer(800, 600)

    # initialisation de l'état du système
    rayon = 1

    while not est_fermee(f):
        # affichage de l'état du système
        couleur(f, 0, 1, 0)
        disque(f, 400, 300, rayon)

        # gestion des événements etc
        attendre_pendant(f, 10) # 10 ms
        # mise à jour de l'état du système
        rayon = rayon + 0.03

principale()
```

#### ./d-disque-aff-unite.py

```
import math
from qtido import *
```

```
def principale():
    f = creer(800, 600)

    # initialisation de l'état du système
    rayon = 1 # cm

    px_par_cm = 38 # px / cm

    while not est_fermee(f):
        # affichage de l'état du système
        effacer(f)
        couleur(f, 0, 1, 0)
        disque(f, 400, 300, rayon * px_par_cm)

        # gestion des événements etc
        attendre_pendant(f, 10) # 10 ms
        # mise à jour de l'état du système
        rayon = rayon + 0.03

    principale()
```

./e-mru.py

```
from qtido import *

def principale():
    f = creer(800, 600)

    # initialiser l'état
    x = 0 # m

    dt = 0.010 # s
    v = 5 # m/s
    r = 1 # m
    px_par_m = 10 # px / m

    while not est_fermee(f):
        # afficher l'état
        effacer(f)
        disque(f,
                400 + x * px_par_m,
                300,
                r * px_par_m)

        # événement et mise à jour
        attendre_pendant(f, dt*1000) # ms

        x = x + v * dt

    principale()
```

./f-mrua.py

```
from qtido import *

def principale():
    f = creer(800, 600)

    # initialiser l'état
    y = 0 # m
    v = 10 # m/s

    dt = 0.010 # s
    r = 1 # m
    px_par_m = 10 # px / m
    g = 9.81 # m / s2

    while not est_fermee(f):
```

```
# afficher l'état
effacer(f)
disque(f,
        400,
        300 - y * px_par_m,
        r * px_par_m)

# événement et mise à jour
attendre_pendant(f, dt*1000) # ms

y = y + v * dt
v = v + (-g) * dt
```

```
principale()
```