

```

        imgUrl = imgTag['src']

        imgName = os.path.basename(imgUrl)

        # ne pas retélécharger l'image si elle existe déjà
        if not os.path.exists(imgName):
            urllib.urlretrieve("http://apod.nasa.gov/apod/"+imgUrl,imgName)

        # télécharge, redimensionne et enregistre l'image du gadget
        imgOriginal = Image.open(imgName)
        imgResized = imgOriginal.resize((200, 200), Image.NEAREST)
        imgResized.save(imgName)

        imgLoad = pygame.image.load(imgName)
        windowSurface.blit(imgLoad, (0,0))
        pygame.display.update()
        # si une exception se produit, ignorer le téléchargement pour cette fois
    except (IOError, TypeError, RuntimeError):
        print "Erreur de téléchargement, nouvel essai ultérieurement"

# pause de 8 heures car nous ne voulons pas inonder le serveur !
time.sleep(28800)

```

widget_rss.py :

```

import pygame, os, sys, feedparser, time
pygame.init()

# lit l'entrée standard à partir du processus parent et calcule la position
du gadget à l'écran
baseXPos = int(sys.stdin.readline()) - 300 - 10
os.environ['SDL_VIDEO_WINDOW_POS'] = str(baseXPos) + "," + str(430)

# création de la fenêtre Pygame et remplissage de l'arrière-plan avec des
blocs de couleur
screen = pygame.display.set_mode((300,150))
pygame.draw.rect(screen, (80,140,80), (0,0,300,50))
pygame.draw.rect(screen, (80,80,80), (0,50,300,50))
pygame.draw.rect(screen, (160,160,160), (0,100,300,50))

# définit la police pour afficher le texte RSS
font = pygame.font.SysFont('dejavuserif', 10, True)

while True:
    myFeed = feedparser.parse('http://www.raspberrypi.org/feed')

    # donne au titre de la fenêtre le nom du blog
    pygame.display.set_caption(myFeed['feed']['title']+" RSS")

    # récupère les articles depuis le flux RSS et affiche le texte
    for i in range(0, 3):
        layerText = pygame.Surface(screen.get_size())
        outputText = (myFeed['items'][i].title, myFeed['items'][i].updated,
                      myFeed['items'][i].link, myFeed['items'][i].description)
        # efface la surface à chaque boucle en la remplissant avec des
        pixels transparents
        layerText.set_colorkey((0,0,0))
        layerText.fill((0,0,0))

        j = 0
        for line in outputText:
            j = layerText.get_rect().y + j + 5
            text = font.render(line.rstrip('\n'), 0, (255,255,255))
            textpos = text.get_rect()
            textpos.x = layerText.get_rect().x + 5
            textpos.y = textpos.y+j
            layerText.blit(text, textpos)
            screen.blit(layerText, (0, 50*i))
            pygame.display.flip()
            j = j +5

    # pause pendant une heure, ne pas inonder le serveur !
    time.sleep(3600)

```

Suggestions d'améliorations

L'affichage des gadgets n'est rafraîchi que lorsqu'il faut vérifier si un nouveau contenu doit être téléchargé, ce qui cause l'apparition d'un blanc si une autre fenêtre est glissée par dessus. Essayez d'utiliser `datetime` pour déterminer quand télécharger un nouveau contenu tandis que la mise à jour de l'affichage à l'écran se fera séparément chaque seconde.