



## DOCUMENTATION PYXEL

### PALETTE DE COULEURS

0	#000000 0.0.0	1	#2B335F 43.51.95	2	#7E2072 126.32.114	3	#19959C 25.149.156
4	#8B4B52 139.72.82	5	#395C9B 57.92.152	6	#A9C1FF 169.193.255	7	#EEEEEE 238.238.238
8	#D41B6C 212.24.108	9	#D3B441 211.132.65	10	#E9C35B 233.195.91	11	#70C6A9 112.198.169
12	#7696DE 118.150.222	13	#A3A3A3 163.163.163	14	#FF979B 255.151.152	15	#EDC7B0 237.199.176



### CRÉER UNE APPLICATION PYXEL

Après avoir importé le module Pyxel dans votre script Python, spécifiez d'abord la taille de la fenêtre avec la fonction `init`, puis lancez l'application Pyxel avec la fonction `run`.

```
import pyxel

pyxel.init(128, 128, title="Nuit du Code")

def update():
    if pyxel.btnp(pyxel.KEY_Q):
        pyxel.quit()

def draw():
    pyxel.cls(0)
    pyxel.rect(10, 10, 20, 20, 11)

pyxel.run(update, draw)
```

Les arguments de la fonction `run` sont la fonction `update` pour mettre à jour chaque frame et la fonction `draw` pour dessiner sur l'écran quand c'est nécessaire.

Dans une vraie application, il est recommandé de mettre le code Pyxel dans une classe comme ci-dessous :

```
import pyxel

class App:
    def __init__(self):
        pyxel.init(128, 128, title="Nuit du Code")
        self.x = 0
        pyxel.run(self.update, self.draw)

    def update(self):
        self.x = (self.x + 1) % pyxel.width

    def draw(self):
        pyxel.cls(0)
        pyxel.rect(self.x, 0, 8, 8, 9)

App()
```

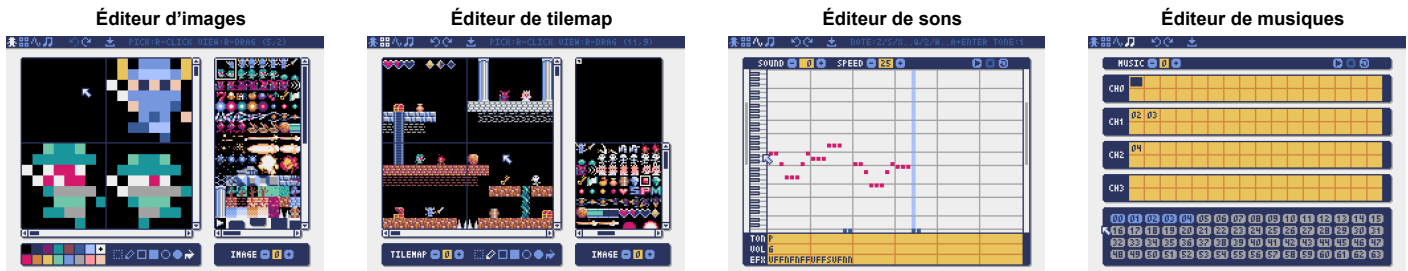
## LANCER UNE APPLICATION PYXEL

Si vous utilisez une console Python, le script créé se lance avec la commande suivante : `pyxel run FICHIER.PY`

## ÉDITER UNE RESSOURCE

Si vous utilisez une console Python, l'éditeur Pyxel se lance avec la commande suivante : `pyxel edit FICHIER.PYXRES`

Un fichier ressource possède l'extension `.pyxres`. La ressource peut être chargée avec la fonction `load` (voir plus bas).



## DOCUMENTATION DE L'API

### Système

- **width, height**  
La largeur et la hauteur de l'écran
- **frame\_count**  
Le nombre de frames passées
- **init(width, height, [title], [fps], [quit\_key], [display\_scale], [capture\_scale], [capture\_sec])**  
Initialise l'application Pyxel avec un écran de taille (width, height). Il est possible de passer comme options : le titre de la fenêtre avec **title**, le nombre d'images par seconde avec **fps**, la touche pour quitter l'application avec **quit\_key**, l'échelle de l'affichage avec **display\_scale**, l'échelle des captures d'écran avec **capture\_scale**, et le temps maximum d'enregistrement vidéo avec **capture\_sec**.  
Par exemple : `pyxel.init(160, 120, title="My Pyxel App", fps=60, quit_key=pyxel.KEY_NONE, capture_scale=3, capture_sec=0)`
- **run(update, draw)**  
Lance l'application Pyxel et appelle la fonction **update** et la fonction **draw**.
- **show()**  
Affiche l'écran jusqu'à ce que la touche **Esc** soit appuyée.
- **flip()**  
Rafraîchit l'écran d'une image. L'application se termine lorsque la touche **Esc** est pressée. Cette fonction ne fonctionne pas dans la version web.
- **quit()**  
Quitte l'application Pyxel.

### Ressources

- **load(filename, [excl\_images], [excl\_tilemaps], [excl\_sounds], [excl\_musics])**  
Charge la ressource (`.pyxres`). Si une option est **True**, la ressource ne sera pas chargée. Si un fichier de palette (`.pyxpal`) du même nom existe au même endroit que le fichier de ressource, la couleur d'affichage de la palette sera également modifiée. Le fichier de palette est une entrée hexadécimale des couleurs d'affichage (par exemple `1100FF`), séparée par des nouvelles lignes. Le fichier de palette peut également être utilisé pour changer les couleurs affichées dans l'éditeur Pyxel.

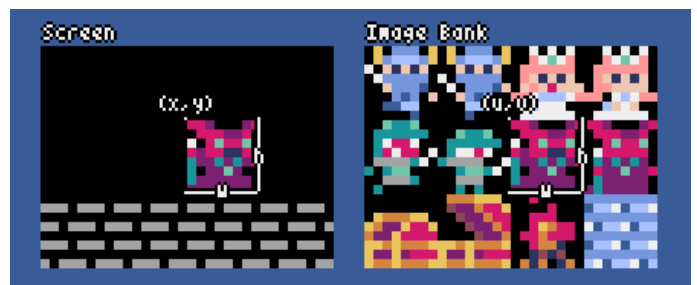
### Entrées

- **mouse\_x, mouse\_y**  
La position actuelle du curseur de la souris
- **mouse\_wheel**  
La valeur actuelle de la molette de la souris
- **btn(key)**  
Renvoie **True** si la touche **key** est appuyée, sinon renvoie **False**. (voir liste des touches plus bas).
- **btnp(key, [hold], [repeat])**  
Renvoie **True** si la touche **key** est appuyée à cette frame, sinon renvoie **False**. Quand **hold** et **repeat** sont spécifiés, **True** sera renvoyé à l'intervalle de frame **repeat** quand la touche **key** est appuyée pendant plus de **hold** frames.
- **btnr(key)**  
Renvoie **True** si la touche **key** est relâchée à cette frame, sinon renvoie **False**

- **mouse(visible)**  
Si **visible** est **True**, affiche le curseur de la souris. Si **False**, le curseur est caché. Même si le curseur n'est pas affiché, sa position est actualisée.

## Graphiques

- **colors**  
Liste les couleurs de la palette. Les couleurs sont spécifiées avec une valeur 24-bit. Vous pouvez utiliser `colors.from_list` et `colors.to_list` pour directement donner et recevoir une liste Python.  
Par exemple `old_colors = pygame.colors.to_list(); pygame.colors.from_list([0x111111, 0x222222, 0x333333]); pygame.colors[15] = 0x112233`
- **images**  
Liste des banques d'images (0-2). (Voir la classe Image)  
Par exemple `pygame.images[0].load(0, 0, "title.png")`
- **tilemaps**  
Liste des cartes de tuiles (0-7). (voir la classe Tilemap)
- **clip(x, y, w, h)**  
Défini la zone de dessin (x, y) avec une largeur w et une hauteur h. Réinitialiser la zone de dessin au plein écran avec `clip()`
- **camera(x, y)**  
Change the upper left corner coordinates of the screen to (x, y). Reset the upper left corner coordinates to (0, 0) with `camera()`.
- **pal(col1, col2)**  
Remplace la couleur col1 avec col2 au dessin. `pal()` pour réinitialiser la palette de couleurs.
- **dither(alpha)**  
Applique le tramage (pseudo-transparence) au dessin. Fixe alpha dans l'intervalle 0.0-1.0, où 0.0 est transparent et 1.0 est opaque.
- **cls(col)**  
Efface l'écran avec la couleur col
- **pget(x, y)**  
Renvoie la couleur au pixel (x, y).
- **pset(x, y, col)**  
Dessine un pixel de couleur col à (x, y).
- **line(x1, y1, x2, y2, col)**  
Dessine une ligne de couleur col de (x1, y1) à (x2, y2).
- **rect(x, y, w, h, col)**  
Dessine un rectangle de largeur w, de hauteur h et de couleur col à partir de (x, y).
- **rectb(x, y, w, h, col)**  
Dessine les contours d'un rectangle de largeur w, de hauteur h et de couleur col à partir de (x, y).
- **circ(x, y, r, col)**  
Dessine un cercle de rayon r et de couleur col à (x, y).
- **circb(x, y, r, col)**  
Dessine le contour d'un cercle de rayon r et de couleur col à (x, y).
- **elli(x, y, w, h, col)**  
Dessinez une ellipse de largeur w, de hauteur h et de couleur col à partir de (x, y).
- **ellib(x, y, w, h, col)**  
Dessinez le contour d'une ellipse de largeur w, de hauteur h et de couleur col à partir de (x, y).
- **tri(x1, y1, x2, y2, x3, y3, col)**  
Dessine un triangle avec les sommets (x1, y1), (x2, y2), (x3, y3) et de couleur col
- **trib(x1, y1, x2, y2, x3, y3, col)**  
Dessine les contours d'un triangle avec les sommets (x1, y1), (x2, y2), (x3, y3) et de couleur col
- **fill(x, y, col)**  
Remplit la zone connectée avec la même couleur que (x, y) avec la couleur col.
- **blt(x, y, img, u, v, w, h, [colkey], [rotate], [scale] )**  
Copie la région de taille (w, h) de (u, v) de la banque d'image img(0-2) à (x, y). Si une valeur négative est assignée à w et/ou h, la région sera retournée horizontalement et/ou verticalement. Si colkey est spécifiée, elle sera traitée comme une couleur transparente. Si rotate (en degrés), scale (1.0 = 100%) ou les deux sont spécifiés, les transformations correspondantes seront appliquées.



- **bltm(x, y, tm, u, v, w, h, [colkey], [rotate], [scale] )**

Copie la région de taille (w, h) de (u, v) de la tilemap **tm**(0-7) à (x, y). Si une valeur négative est mise pour **w** et/ou **h**, la région sera retournée horizontalement et/ou verticalement. Si **colkey** est spécifiée, elle sera traitée comme une couleur transparente. Si **rotate** (en degrés), **scale** (1.0 = 100%) ou les deux sont spécifiés, les transformations correspondantes seront appliquées. La taille d'une tuile est de 8x8 pixels et est stockée dans une carte de tuiles sous forme de tuple (**image\_tx**, **image\_ty**).

- **text(x, y, s, col)**

Dessine une chaîne de caractères **s** de couleur **col** à (x, y).



## Audio

- **sounds**

Liste des sons (0-63). (voir la classe Sound)

par exemple : `pyxel.sounds[0].speed = 60`

- **musics**

Liste des musiques (0-7). (voir la classe Music)

- **play(ch, snd, [tick], [loop])**

Joue le son **snd**(0-63) sur le canal **ch**(0-3). Si **snd** est une liste, les sons seront joués dans l'ordre. La position de début de lecture peut être spécifiée par **tick**(1 tick = 1/120 secondes). Si **loop** est à **True**, le son est joué en boucle.

- **playm(msc, [tick], [loop])**

Joue la musique **msc**(0-7). La position de début de lecture peut être spécifiée par **tick**(1 tick = 1/120 secondes). Si **loop** est mis à **True**, la musique est jouée en boucle.

- **stop([ch])**

Arrête le son du canal spécifié **ch**(0-3). **stop()** arrête tous les canaux.

- **play\_pos(ch)**

Récupère la position du son du canal **ch**(0-3) comme un tuple (**sound no**, **note no**). Renvoie **None** quand le son est arrêté.

## Mathématiques

- **ceil(x)**

Renvoie le plus petit nombre entier supérieur ou égal à **x**.

- **floor(x)**

Renvoie le plus grand nombre entier inférieur ou égal à **x**.

- **sgn(x)**

Renvoie 1 lorsque **x** est positif, 0 lorsqu'il est nul, et -1 lorsqu'il est négatif.

- **sqr(x)**

Renvoie la racine carrée de **x**.

- **sin(deg)**

Renvoie le sinus de **deg** degrés.

- **cos(deg)**

Renvoie le cosinus de **deg** degrés.

- **atan2(y, x)**

Retourne l'arctangente de **y/x** en degrés.

- **rseed(seed)**

Définit la graine du générateur de nombres aléatoires.

- **rndi(a, b)**

Renvoie un nombre entier aléatoire supérieur ou égal à **a** et inférieur ou égal à **b**.

- **rndf(a, b)**

Renvoie une décimale aléatoire supérieure ou égale à **a** et inférieure ou égale à **b**.

- **nseed(seed)**

Définit la graine du bruit de Perlin.

- **noise(x, [y], [z])**

Renvoie la valeur du bruit de Perlin pour les coordonnées spécifiées.

## Classe Image

- **width, height**  
La largeur et la hauteur d'une image
- **get(x, y)**  
Renvoie les données de l'image à (x, y).
- **set(x, y, data)**  
Définit l'image à (x, y) à l'aide d'une liste de chaînes.  
Exemple : `pyxel.images[0].set(10, 10, ["0123", "4567", "89ab", "cdef"])`
- **load(x, y, filename)**  
Charge l'image (PNG/GIF/JPEG) à (x, y).

## Classe Tilemap

- **width, height**  
La largeur et la hauteur de la tilemap
- **imgsrc**  
La banque d'image (0-2) référencée par la tilemap
- **set(x, y, data)**  
Met la tilemap à (x, y) suivant une liste de chaînes.  
Par exemple `pyxel.tilemap(0).set(0, 0, ["0000 0100 a0b0", "0001 0101 a1b1"])`
- **load(x, y, filename, layer)**  
Chargez la couche dans l'ordre de dessin **couche**(0-) à partir du fichier TMX (Tiled Map File) à (x, y).
- **pget(x, y)**  
Renvoie la tile à (x, y). Une tile est un tuple (tile\_x, tile\_y).
- **pset(x, y, tile)**  
Dessine une **tile** à (x, y). Une tile est un tuple (tile\_x, tile\_y).

## Classe Sound

- **notes**  
Liste des notes (0-127). Plus le nombre est haut, plus la note est haute, et à 33 ça devient 'A2' (440Hz). Le reste est à -1.
- **tones**  
Liste les tons (0:Triangle / 1:Square / 2:Pulse / 3:Noise)
- **volumes**  
Liste les volumes (0-7)
- **effects**  
Liste les effets (0:None / 1:Slide / 2:Vibrato / 3:FadeOut)
- **speed**  
Vitesse de lecture. 1 est le plus rapide, et plus le nombre est grand, plus la vitesse est lente. à 120, la longueur d'une note est de 1 second.
- **set(notes, tones, volumes, effects, speed)**  
Met les valeurs de notes, tones, volumes et effects avec une chaîne. Si les tons, volumes et effects sont plus courts que les notes, ils sont répétés depuis le début.
- **set\_notes(notes)**  
Met les notes avec une chaîne de 'CDEFGAB'+#+'01234' ou 'R'. Insensible à la casse et les espaces sont ignorés.  
Par exemple `pyxel.sounds[0].set_notes("G2B-2D3R RF3F3F3")`
- **set\_tones(tones)**  
Met les tons avec une chaîne de 'TSPN'. Insensible à la casse et les espaces sont ignorés.  
Par exemple `pyxel.sounds[0].set_tones("TTSS PPPN")`
- **set\_volumes(volumes)**  
Met les volumes avec une chaîne de '01234567'. Insensible à la casse et les espaces sont ignorés.  
Par exemple `pyxel.sounds[0].set_volumes("7777 7531")`
- **set\_effects(effects)**  
Met les effets avec une chaîne de 'NSVF'. Insensible à la casse et les espaces sont ignorés.  
Par exemple `pyxel.sounds[0].set_effects("NFNF NVVS")`

## Classe Music

- **seqs**  
Liste bidimensionnelle de sons (0-63) avec le nombre de canaux
- **set(seq0, seq1, seq2, ...)**  
Régler les listes de sons (0-63) des canaux. Si une liste vide est passée, ce canal n'est pas utilisé.  
Par exemple `pyxel.musics[0].set([0, 1], [], [3])`

## Touches et souris

KEY_BACKSPACE	KEY_C	KEY_PAGEDOWN	KEY_COPY	KEY_KP_VERTICALBAR	KEY_AC_REFRESH
KEY_TAB	KEY_D	KEY_RIGHT	KEY_PASTE	KEY_KP_DBLVERTICALBAR	KEY_AC_BOOKMARKS
KEY_RETURN	KEY_E	KEY_LEFT	KEY_FIND	KEY_KP_COLON	KEY_BRIGHTNESSDOWN
KEY_ESCAPE	KEY_F	KEY_DOWN	KEY_MUTE	KEY_KP_HASH	KEY_BRIGHTNESSUP
KEY_SPACE	KEY_G	KEY_UP	KEY_VOLUMEUP	KEY_KP_SPACE	KEY_DISPLAYSWITCH
KEY_EXCLAM	KEY_H	KEY_NUMLOCKCLEAR	KEY_VOLUMEDOWN	KEY_KP_AT	KEY_KBDILLUMTOGGLE
KEY_QUOTEDBL	KEY_I	KEY_KP_DIVIDE	KEY_KP_COMMA	KEY_KP_EXCLAM	KEY_KBDILLUMDOWN
KEY_HASH	KEY_J	KEY_KP_MULTIPLY	KEY_KP_EQUALSAS400	KEY_KP_MEMSTORE	KEY_KBDILLUMUP
KEY_DOLLAR	KEY_K	KEY_KP_MINUS	KEY_ALTERASE	KEY_KP_MEMRECALL	KEY_EJECT
KEY_PERCENT	KEY_L	KEY_KP_PLUS	KEY_SYSREQ	KEY_KP_MEMCLEAR	KEY_SLEEP
KEY_AMPERSAND	KEY_M	KEY_KP_ENTER	KEY_CANCEL	KEY_KP_MEMADD	KEY_NONE
KEY_QUOTE	KEY_N	KEY_KP_1	KEY_CLEAR	KEY_KP_MEMSUBTRACT	KEY_SHIFT
KEY_LEFTPAREN	KEY_O	KEY_KP_2	KEY_PRIOR	KEY_KP_MEMMULTIPLY	KEY_CTRL
KEY_RIGHTPAREN	KEY_P	KEY_KP_3	KEY_RETURN2	KEY_KP_MEMDIVIDE	KEY_ALT
KEY_ASTERISK	KEY_Q	KEY_KP_4	KEY_SEPARATOR	KEY_KP_PLUSMINUS	KEY_GUI
KEY_PLUS	KEY_R	KEY_KP_5	KEY_OUT	KEY_KP_CLEAR	MOUSE_POS_X
KEY_COMMA	KEY_S	KEY_KP_6	KEY_OPER	KEY_KP_CLEARENTRY	MOUSE_POS_Y
KEY_MINUS	KEY_T	KEY_KP_7	KEY_CLEARAGAIN	KEY_KP_BINARY	MOUSE_WHEEL_X
KEY_PERIOD	KEY_U	KEY_KP_8	KEY_CRSEL	KEY_KP_OCTAL	MOUSE_WHEEL_Y
KEY_SLASH	KEY_V	KEY_KP_9	KEY_EXSEL	KEY_KP_DECIMAL	MOUSE_BUTTON_LEFT
KEY_0	KEY_W	KEY_KP_0	KEY_KP_00	KEY_KP_HEXADECIMAL	MOUSE_BUTTON_MIDDLE
KEY_1	KEY_X	KEY_KP_PERIOD	KEY_KP_000	KEY_LCTRL	MOUSE_BUTTON_RIGHT
KEY_2	KEY_Y	KEY_APPLICATION	KEY_THOUSANDSSEPARATOR	KEY_LSHIFT	MOUSE_BUTTON_X1
KEY_3	KEY_Z	KEY_POWER	KEY_DECIMALSEPARATOR	KEY_LALT	MOUSE_BUTTON_X2
KEY_4	KEY_DELETE	KEY_KP_EQUALS	KEY_CURRENCYUNIT	KEY_LGUI	MOUSE_BUTTON_UNKNOWN
KEY_5	KEY_CAPSLOCK	KEY_F13	KEY_CURRENCYSUBUNIT	KEY_RCTRL	
KEY_6	KEY_F1	KEY_F14	KEY_KP_LEFTPAREN	KEY_RSHIFT	
KEY_7	KEY_F2	KEY_F15	KEY_KP_RIGHTPAREN	KEY_RALT	
KEY_8	KEY_F3	KEY_F16	KEY_KP_LEFTBRACE	KEY_RGUI	
KEY_9	KEY_F4	KEY_F17	KEY_KP_RIGHTBRACE	KEY_MODE	
KEY_COLON	KEY_F5	KEY_F18	KEY_KP_TAB	KEY_AUDIONEXT	
KEY_SEMICOLON	KEY_F6	KEY_F19	KEY_KP_BACKSPACE	KEY_AUDIOPREV	
KEY_LESS	KEY_F7	KEY_F20	KEY_KP_A	KEY_AUDIOSTOP	
KEY_EQUALS	KEY_F8	KEY_F21	KEY_KP_B	KEY_AUDIOPLAY	
KEY_GREATER	KEY_F9	KEY_F22	KEY_KP_C	KEY_AUDIOMUTE	
KEY_QUESTION	KEY_F10	KEY_F23	KEY_KP_D	KEY_MEDIASELECT	
KEY_AT	KEY_F11	KEY_F24	KEY_KP_E	KEY_WWW	
KEY_LEFTBRACKET	KEY_F12	KEY_EXECUTE	KEY_KP_F	KEY_MAIL	
KEY_BACKSLASH	KEY_PRINTSCREEN	KEY_HELP	KEY_KP_XOR	KEY_CALCULATOR	
KEY_RIGHTBRACKET	KEY_SCROLLLOCK	KEY_MENU	KEY_KP_POWER	KEY_COMPUTER	
KEY_CARET	KEY_PAUSE	KEY_SELECT	KEY_KP_PERCENT	KEY_AC_SEARCH	
KEY_UNDERSCORE	KEY_INSERT	KEY_STOP	KEY_KP_LESS	KEY_AC_HOME	
KEY_BACKQUOTE	KEY_HOME	KEY_AGAIN	KEY_KP_GREATER	KEY_AC_BACK	
KEY_A	KEY_PAGEUP	KEY_UNDO	KEY_KP_AMPERSAND	KEY_AC_FORWARD	
KEY_B	KEY_END	KEY_CUT	KEY_KP_DBLAMPERSAND	KEY_AC_STOP	