

# FICHE PYTHON 16 : INTERFACE GRAPHIQUE 5 - EVENEMENTS (4 pages)

## Vocabulaire

<b>Événement Event</b>	Nom donné à quelque chose de détectable : déplacement de la souris, appui de touche.
<b>Gestionnaire d'événement Event Handler</b>	<p>C'est le gestionnaire qui va donner l'ordre à la fonction désignée de s'activer si l'événement survient. Ces fonctions nécessitent un paramètre obligatoire qu'on nomme souvent <b>event</b> : c'est un objet qui contient la nature de l'événement lors de l'appel de la fonction.</p> <pre>def gest_recule(event):     recule_x()</pre> <p>Ici, la fonction <code>gest_recule</code> est celle qu'on lance à la suite d'un événement.</p>
<b>Méthode bind()</b>	<p>On crée le lien entre la fonction à déclencher et l'événement à l'aide de la méthode <b>bind()</b>. Cette méthode est une méthode des objets-fenêtre : on surveille les événements dans une fenêtre.</p> <pre>fen_princ.bind('&lt;Left&gt;', gest_recule)</pre> <p>On doit donner le nom de la fenêtre (<b>fen_princ</b> ici). Ensuite, on précise la nature de l'événement dans un string ('&lt;Left&gt;'). Enfin, on donne le nom de la fonction à lier à l'événement, <b>gest_recule</b> dans l'exemple.</p>

## Événement : Appui sur une touche et description du contenu de event

<b>Code</b>	<pre>def gest_affichage(event):     print('type : \t', event.type)     print('keysym : \t', event.keysym)     print('keycode : \t', event.keycode)     print('keysym_num : \t', event.keysym_num)     print(event.x, " ", event.y)  fen_princ.bind('&lt;Any-KeyPress&gt;', gest_affichage)</pre>
<b>Description des attributs de event</b>	<p><code>event.type</code> contient un nombre qui code le type d'événement déclencheur (2 pour l'appui sur une touche par exemple).</p> <p><code>event.keycode</code> correspond au code unicode du caractère principal de la touche. La touche « A » donne 65 qu'on affiche A ou a puisqu'il s'agit de la même touche physique.</p> <p><code>event.keysym</code> correspond au caractère qui va s'afficher dans une zone de texte en appuyant sur cette touche. Ainsi A et a donnent des résultats différents.</p> <p><code>event.keysym_num</code> correspond au code unicode du caractère : 65 pour A, 97 pour a.</p> <p><code>event.x</code> et <code>event.y</code> correspondent aux coordonnées de la souris lors de l'événement.</p>

## Description des touches hors clavier numérique

Nom de la touche	keysym (caractère obtenu)	keycode	keysym_num
A + Caps Lock activé	<b>A</b>	65	65
B + Caps Lock activé	<b>B</b>	66	66
...	...	...	...
Y + Caps Lock activé	<b>Y</b>	89	89
Z + Caps Lock activé	<b>Z</b>	90	90
Nom de la touche	keysym(caractère obtenu)	keycode	keysym_num
A	<b>a</b>	65	97
B	<b>b</b>	66	98
...	...	...	...
Y	<b>y</b>	89	121
Z	<b>z</b>	90	122

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Tabulation	<b>Tab</b>	9	65289
Verrouillage majuscule	<b>Caps_Lock</b>	20	65509
Touche shift de gauche(maj)	<b>Shift_L</b>	16	65505
Touche contrôle de gauche	<b>Control_L</b>	17	65507
Touche Windows de gauche	<b>Win_L</b>	91	65371
Touche Alt de gauche	<b>Alt_L</b>	18	65513
Space	<b>space</b>	32	32
Touche Alt Gr	<b>Alt_R</b>	18	65514
Touche Windows de droite	<b>Win_R</b>	92	65372
Touche Applications	<b>App</b>	93	65373
Touche contrôle de droite	<b>Control_R</b>	17	65508
Touche shift de droite(maj)	<b>Shift_R</b>	16	65506
Touche Entrée	<b>Return</b>	13	65293
Retour	<b>BackSpace</b>	8	65288

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Echap.	<b>Escape</b>	27	65307
F1	<b>F1</b>	112	65470
F2	<b>F2</b>	113	65471
F3	<b>F3</b>	114	65472
F4	<b>F4</b>	115	65473
F5	<b>F5</b>	116	65474
F6	<b>F6</b>	117	65475
F7	<b>F7</b>	118	65476
F8	<b>F8</b>	119	65477
F9	<b>F9</b>	120	65478
F10	<b>F10</b>	121	65479
F11	<b>F11</b>	122	65480
F12	<b>F12</b>	123	65481
Arrêt défilement	<b>Scroll_Lock</b>	145	65300
Pause	<b>Pause</b>	19	65299

Nom de la touche	keysym (caractère obtenu)	keycode	keysym_num
0 + Caps Lock activé	<b>0</b>	48	48
1 + Caps Lock activé	<b>1</b>	49	49
2 + Caps Lock activé	<b>2</b>	50	50
3 + Caps Lock activé	<b>3</b>	51	51
4 + Caps Lock activé	<b>4</b>	52	52
5 + Caps Lock activé	<b>5</b>	53	53
6 + Caps Lock activé	<b>6</b>	54	54
7 + Caps Lock activé	<b>7</b>	55	55
8 + Caps Lock activé	<b>8</b>	56	56
9 + Caps Lock activé	<b>9</b>	57	57
Nom de la touche	keysym (caractère obtenu)	keycode	keysym_num
0	<b>grave à</b>	48	224
1	<b>ampersand &amp;</b>	49	38
2	<b>eacute é</b>	50	233
3	<b>quotedbl "</b>	51	34
4	<b>quoteright '</b>	52	39
5	<b>parenleft (</b>	53	53
6	<b>minus -</b>	54	45
7	<b>egrave è</b>	55	232
8	<b>underscore _</b>	56	95
9	<b>ccedilla ç</b>	57	231

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Flèche Gauche	<b>Left</b>	37	65361
Flèche Haut	<b>Up</b>	38	65362
Flèche Droite	<b>Right</b>	39	65363
Flèche Bas	<b>Down</b>	40	65364
Page précédente	<b>Prior</b>	33	65365
Page suivante	<b>Next</b>	34	65366
Insérer	<b>Insert</b>	45	65379
Supprimer	<b>Delete</b>	46	65535
Début de ligne	<b>Home</b>	36	65360
Fin de ligne	<b>End</b>	35	65367

Exemple :

si vous voulez lancer l'appel de `gest_avance()` avec la touche **&**, il faut noter

```
fen_princ.bind('<Key-ampersand>',
gest_avance)
```

ou

```
fen_princ.bind('<ampersand>', gest_avance)
```

## Description des touches du clavier numérique

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Verrouillage Numérique	Num_Lock	144	65407

  

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Division	slash	111	47
Multiplication	asterisk	106	42
Soustraction	minus	109	45
Addition	plus	107	43
Entrée	Return	13	65293
Virgule/Point	period	110	46
0	0	96	48
1	1	97	49
2	2	98	50
3	3	99	51
4	4	100	52
5	5	101	53
6	6	102	54
7	7	103	55
8	8	104	56
9	9	105	57

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Division	slash	111	47
Multiplication	asterisk	106	42
Soustraction	minus	109	45
Addition	plus	107	43
Entrée	Return	13	65293

  

Nom de la touche	keysym (nom interne)	keycode	keysym_num
Point/Virgule	Delete	46	65535
0	Insert	45	65379
1	End	35	65367
2	Down	40	65364
3	Next	34	65366
4	Left	37	65361
5	Clear	12	65291
6	Right	39	65363
7	Home	36	65360
8	Up	38	65362
9	Prior	33	65365

## Description des événements

Type (event.type)	Nom à utiliser dans bind()	Description
2	KeyPress	L'utilisateur vient d'appuyer sur la touche. On peut également utiliser uniquement Key. Exemple : <code>'&lt;KeyPress-Left&gt;'</code> ou <code>'&lt;Key-Left&gt;'</code> ou encore <code>'&lt;Left&gt;'</code>
3	KeyRelease	L'utilisateur vient de relacher une touche. Exemple : <code>'&lt;KeyRelease-Left&gt;'</code> .
4	Button	L'utilisateur vient d'appuyer sur l'un des boutons de la souris. Exemple : <code>'&lt;Button-1&gt;'</code> ou <code>'&lt;Button-2&gt;'</code> ou <code>'&lt;Button-3&gt;'</code> . Si event est le nom donné au paramètre "événement" du gestionnaire, on peut obtenir les coordonnées à l'aide de <code>event.x</code> ou <code>event.y</code> . Si on utilise <code>monCanvas.bind('&lt;Button-1&gt;', gest_avance)</code> , on activera <code>gest_avance()</code> si on appuie sur le bouton 1 de la souris alors qu'on est au dessus du Canvas. Si vous voulez que cela marche sur toute la fenêtre, il faut utiliser <code>fen_princ.bind('&lt;Button-1&gt;', gest_avance)</code> .
5	ButtonRelease	L'utilisateur vient de relacher l'un des boutons de la souris. Exemple : <code>'&lt;ButtonRelease-1&gt;'</code> . A privilégier par rapport au type 4 : l'utilisateur peut sortir du widget pour désactiver son action en cas d'erreur, alors que l'exécution est automatique en cas de type 4 (Button).
6	Motion	L'utilisateur a bougé la souris : <code>'&lt;Motion&gt;'</code> . Si on utilise <code>monCanvas.bind('&lt;Motion&gt;', gest_avance)</code> , on lance l'appel de <code>gest_avance()</code> lorsque la souris bouge au dessus du widget <code>monCanvas</code> .
7	Enter	L'utilisateur vient de faire rentrer la souris dans le Widget : <code>'&lt;Enter&gt;'</code> . Même remarque qu'au dessus, la seule différence étant qu'on ne relance pas l'appel si on bouge ensuite dans le widget. L'activation ne se fait que lors de la rentrée. <code>monCanvas.bind('&lt;Enter&gt;', gest_avance)</code> va lancer <code>gest_avance()</code> lorsqu'on fait rentrer le pointeur de la souris dans <code>monCanvas</code> .
8	Leave	L'utilisateur vient de faire sortir la souris d'un Widget : <code>'&lt;Leave&gt;'</code> . Même remarque qu'au dessus, la seule différence étant qu'on ne relance pas l'appel si on bouge ensuite dans le widget. L'activation ne se fait que lors de la sortie. <code>monCanvas.bind('&lt;Leave&gt;', gest_recule)</code> va lancer <code>gest_recule()</code> lorsqu'on fait sortir le pointeur de la souris de <code>monCanvas</code> .

## Modificateur d'événement

On peut utiliser **Alt** ou **Control** ou **Shift** si on veut dire que la touche Alt (ou Control ou Shift) doit être enfoncée lors de l'événement.

`monCanvas.bind('<Alt-Button-1>', gest_monte)` veut dire qu'on active `gest_monte` si on appuie ET qu'on maintient ALT enfoncé dans le même temps.

On peut utiliser **Any** pour généraliser l'événement.

`fen_princ.bind('<Any-KeyPress>', gest_monte)` veut dire qu'on active `gest_monte` si on appuie sur n'importe quelle touche.

On peut utiliser **Lock** pour imposer que le verrouillage majuscule soit actif.

`fen_princ.bind('<Lock-KeyRelease-A>', gest_monte)` veut dire qu'on active `gest_monte` si on relache A alors que le verrouillage majuscule est actif.