

Notions de programmation : langage Python

Instruction et commande PYTHON	Exemple commenté					
Affecter une valeur à une variable a de type float ou int a = valeur	<p>🟡 Commentaire Attention, la deuxième ligne ne veut pas dire que a et a + 1 sont égaux mais bien que a va prendre pour valeur la valeur actuelle de a augmentée de 1.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En langage naturel</th> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a ← 2 a ← a+1</td> <td>a = 2 a = a+1</td> </tr> </tbody> </table>	En langage naturel	En PYTHON	a ← 2 a ← a+1	a = 2 a = a+1
En langage naturel	En PYTHON					
a ← 2 a ← a+1	a = 2 a = a+1					
Affecter une chaîne de caractères à une variable a de type str a = "texte" ou a = 'texte'	<p>🟡 Commentaire Le mot bonjour est affecté à la variable mot.</p> <p>🟢 Remarque mot = bonjour renvoie un message d'erreur.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>mot = "bonjour" OU mot = 'bonjour'</td> </tr> </tbody> </table>	En PYTHON	mot = "bonjour" OU mot = 'bonjour'		
En PYTHON						
mot = "bonjour" OU mot = 'bonjour'						
Modifier une variable a de type int en une variable de type float par un calcul où le résultat du calcul n'est pas entier a = calcul	<p>🟡 Commentaire a est de type int à la première ligne mais est converti en type float à la deuxième ligne.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a = 2 a = a + 3.3</td> </tr> </tbody> </table>	En PYTHON	a = 2 a = a + 3.3		
En PYTHON						
a = 2 a = a + 3.3						
Modifier le type d'une variable a en le type flottant (ou un autre type) a = float(a)	<p>🟡 Commentaire a est de type int à la première ligne mais est converti en type float à la deuxième ligne.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a = 2 a = float(a)</td> </tr> </tbody> </table>	En PYTHON	a = 2 a = float(a)		
En PYTHON						
a = 2 a = float(a)						
Affecter une valeur de type str saisie par l'utilisateur à une variable a après l'affichage d'un texte a = input("texte")	<p>🟡 Commentaire Ce programme affiche Saisir un mot: attend la saisie d'une chaîne de caractères par l'utilisateur, qu'il affecte à a, puis affiche la chaîne de caractères saisie.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En langage naturel</th> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>Afficher "Saisir un mot :" a ← Valeur saisie Afficher a</td> <td>a = input("Saisir un mot :") print(a)</td> </tr> </tbody> </table>	En langage naturel	En PYTHON	Afficher "Saisir un mot :" a ← Valeur saisie Afficher a	a = input("Saisir un mot :") print(a)
En langage naturel	En PYTHON					
Afficher "Saisir un mot :" a ← Valeur saisie Afficher a	a = input("Saisir un mot :") print(a)					
Affecter une valeur de type float ou int saisie par l'utilisateur à une variable a après l'affichage d'un texte (on remplace float par int suivant le cas) a = input("texte") a = float(a) ou a = float(input("texte"))	<p>🟡 Commentaire Ce programme affiche a = ? attend la saisie d'une valeur par l'utilisateur, qu'il affecte à a, puis convertit a en flottant puis calcule et affiche le double de a.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En langage naturel</th> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>Afficher "a = ?" a ← Valeur saisie b ← 2*a Afficher b</td> <td>a = input("a = ?") a = float(a) b = 2*a print(b)</td> </tr> </tbody> </table>	En langage naturel	En PYTHON	Afficher "a = ?" a ← Valeur saisie b ← 2*a Afficher b	a = input("a = ?") a = float(a) b = 2*a print(b)
En langage naturel	En PYTHON					
Afficher "a = ?" a ← Valeur saisie b ← 2*a Afficher b	a = input("a = ?") a = float(a) b = 2*a print(b)					
Afficher un texte print("texte") ou print('texte')	<p>🟡 Commentaire Ce programme affiche 3.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a = 3 print(a)</td> </tr> </tbody> </table>	En PYTHON	a = 3 print(a)		
En PYTHON						
a = 3 print(a)						
Afficher la valeur d'une variable a print(a)	<p>🟡 Commentaire Ce programme affiche a = 3.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a = 3 print("a=", a)</td> </tr> </tbody> </table>	En PYTHON	a = 3 print("a=", a)		
En PYTHON						
a = 3 print("a=", a)						
Afficher un texte et la valeur d'une variable a print("texte", a)	<p>🟡 Commentaire Ce programme affiche a = 3.</p>	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #d9ead3;">En PYTHON</th> </tr> </thead> <tbody> <tr> <td>a = 3 print("a=", a)</td> </tr> </tbody> </table>	En PYTHON	a = 3 print("a=", a)		
En PYTHON						
a = 3 print("a=", a)						

- Dans toute la suite, certains blocs d'instructions sont précédés d'une ligne finissant par un double point et sont **indentés**, c'est-à-dire que toutes les instructions sont précédées d'un espace (de même taille). Dans ce cas, la fin de l'indentation marque la fin du bloc.
- Pour les tests de condition (après **if** ou **while**), les symboles =, ≠, ≤ et ≥ s'écrivent respectivement ==, !=, <= et >= en PYTHON.

Instruction et commande PYTHON	Exemple commenté					
<p>Écrire des instructions conditionnelles</p> <pre>if condition: instruction1 instruction2 etc else: instruction3 instruction4 etcbis</pre> <p>Si condition</p> <pre>instruction1 instruction2 etc</pre> <p>Sinon</p> <pre>instruction3 instruction4 etcbis</pre> <p>Fin si</p>	<table border="1" data-bbox="715 427 1225 680"> <thead> <tr> <th data-bbox="721 436 1219 472">En PYTHON</th> </tr> </thead> <tbody> <tr> <td data-bbox="721 481 1219 672"> <pre>x = float(input("x=? ")) if x >= 0: print("Le nombre est positif.") else: print("x<0") print("Son opposé est", -x) print("Bonne journée.")</pre> </td> </tr> </tbody> </table> <p>• Commentaire Si l'utilisateur choisit 5 pour valeur de x, le programme affichera Le nombre est positif puis Bonne journée. Si l'utilisateur choisit - 3, 2 pour valeur de x, le programme affichera x < 0 puis Son opposé est 3, 2 puis Bonne journée.</p> <p>• Remarque S'il n'y a pas de Sinon dans l'algorithme alors le bloc allant de else : à etcbis est retiré.</p>		En PYTHON	<pre>x = float(input("x=? ")) if x >= 0: print("Le nombre est positif.") else: print("x<0") print("Son opposé est", -x) print("Bonne journée.")</pre>		
En PYTHON						
<pre>x = float(input("x=? ")) if x >= 0: print("Le nombre est positif.") else: print("x<0") print("Son opposé est", -x) print("Bonne journée.")</pre>						
<p>Écrire une boucle bornée</p> <pre>for i in range(1,n+1): instruction1 instruction2 etc</pre> <p>Pour i allant de 1 à n</p> <pre>instruction1 instruction2 etc</pre> <p>Fin pour</p>	<table border="1" data-bbox="635 983 1300 1144"> <thead> <tr> <th data-bbox="635 992 1002 1028">En langage naturel</th> <th data-bbox="1008 992 1294 1028">En PYTHON</th> </tr> </thead> <tbody> <tr> <td data-bbox="635 1037 1002 1135"> <pre>Pour i allant de 1 à 100 a ← 2×i Afficher a Fin pour</pre> </td> <td data-bbox="1008 1037 1294 1135"> <pre>for i in range(1,101): a = 2*i print(a)</pre> </td> </tr> </tbody> </table> <p>• Commentaire Ce programme calcule et affiche tous les nombres pairs entre $2 \times 1 = 2$ et $2 \times 100 = 200$ (et non $2 \times 101 = 202$) car range (a, b) désigne les entiers entre a inclus et b exclu.</p>		En langage naturel	En PYTHON	<pre>Pour i allant de 1 à 100 a ← 2×i Afficher a Fin pour</pre>	<pre>for i in range(1,101): a = 2*i print(a)</pre>
En langage naturel	En PYTHON					
<pre>Pour i allant de 1 à 100 a ← 2×i Afficher a Fin pour</pre>	<pre>for i in range(1,101): a = 2*i print(a)</pre>					
<p>Écrire une boucle non bornée</p> <p>Tant que condition</p> <pre>instruction1 instruction2 etc</pre> <p>Fin tant que</p> <pre>while condition : instruction1 instruction2 etc</pre>	<table border="1" data-bbox="635 1305 1300 1489"> <thead> <tr> <th data-bbox="635 1314 1040 1350">En langage naturel</th> <th data-bbox="1046 1314 1294 1350">En PYTHON</th> </tr> </thead> <tbody> <tr> <td data-bbox="635 1359 1040 1480"> <pre>p ← 1 Tant que p ≤ 1 000 000 p ← 2 × p Fin tant que Afficher p</pre> </td> <td data-bbox="1046 1359 1294 1480"> <pre>p = 1 while p <= 1000000: p = 2*p print(p)</pre> </td> </tr> </tbody> </table> <p>• Commentaire Ce programme calcule les puissances de 2 qu'il affecte à la variable p (1 ; 2 ; 4 ; 8 ; etc.) jusqu'à ce que p > 1 000 000 puis affiche cette dernière valeur de p.</p>		En langage naturel	En PYTHON	<pre>p ← 1 Tant que p ≤ 1 000 000 p ← 2 × p Fin tant que Afficher p</pre>	<pre>p = 1 while p <= 1000000: p = 2*p print(p)</pre>
En langage naturel	En PYTHON					
<pre>p ← 1 Tant que p ≤ 1 000 000 p ← 2 × p Fin tant que Afficher p</pre>	<pre>p = 1 while p <= 1000000: p = 2*p print(p)</pre>					
<p>Écrire une fonction</p> <pre>fonctionnom(p1, p2, ...): ... return a</pre> <p>Retourner a</p> <p>programme principal</p> <pre>def nom(p1, p2, ...): ... return a</pre> <p>programme principal</p>	<table border="1" data-bbox="635 1619 1300 1803"> <thead> <tr> <th data-bbox="635 1628 1040 1664">En langage naturel</th> <th data-bbox="1046 1628 1294 1664">En PYTHON</th> </tr> </thead> <tbody> <tr> <td data-bbox="635 1673 1040 1794"> <pre>fonction difference (x, y) dif ← x-y retourner dif</pre> <p>Afficher difference (3, 5)</p> </td> <td data-bbox="1046 1673 1294 1794"> <pre>def difference(x, y): dif = x-y return dif</pre> <p>print(difference(3, 5))</p> </td> </tr> </tbody> </table> <p>• Commentaire Ce programme appelle la fonction difference avec 3 et 5 pour valeurs de ses variables x et y. Celle-ci calcule donc $3 - 5 = - 2$ (affectée à la variable dif) et retourne cette valeur qui est affichée puisque dans un print.</p>		En langage naturel	En PYTHON	<pre>fonction difference (x, y) dif ← x-y retourner dif</pre> <p>Afficher difference (3, 5)</p>	<pre>def difference(x, y): dif = x-y return dif</pre> <p>print(difference(3, 5))</p>
En langage naturel	En PYTHON					
<pre>fonction difference (x, y) dif ← x-y retourner dif</pre> <p>Afficher difference (3, 5)</p>	<pre>def difference(x, y): dif = x-y return dif</pre> <p>print(difference(3, 5))</p>					