

TNSI	Langages et programmation	<b>Exercice - Générateur de mot de passes</b>
	Bases de la programmation	

Ce travail est à faire et à rendre sur pronote pour le **samedi 11 septembre** avant minuit.

On veut réaliser un générateur de mot de passes sécurisé. Pour cela on va procéder en plusieurs étapes.

- 1) Ecrire une fonction `chiffre()` qui renvoi un caractère (`str`) aléatoire entre "0" et "9".
- 2) Ecrire une fonction `majuscule()` qui renvoi un caractère (`str`) aléatoire entre "A" et "Z".
- 3) Ecrire une fonction `minuscule()` qui renvoi un caractère (`str`) aléatoire entre "a" et "z".
- 4) Ecrire une fonction `special()` qui renvoi un caractère (`str`) aléatoire parmi les caractères spéciaux de la table ascii standard.
- 5) Ecrire une fonction `mot_de_passe(n, c, M, m, s)` qui renvoi une chaîne de caractère contenant exactement `n` caractères et comprenant au moins `c` chiffres, `M` lettres majuscules, `m` lettres minuscules et `s` caractères spéciaux ascii. Si  $c + M + m + s < n$ , les caractères restants sont des minuscules. Les caractères imposés peuvent être placés n'importe où dans la chaîne.
- 6) Améliorer votre fonction `mot_de_passe` pour que tous les caractères soient mélangés.
- 7) Ecrire une fonction `contient_chiffres(c)` qui renvoi le nombre de chiffres contenus dans la chaîne `c`.
- 8) Faire de même pour les fonctions `contient_majuscules(c)`, `contient_minuscules(c)` et `contient_speciaux(c)`.
- 9) A partir des fonctions précédentes, écrire une fonction `tests()` permettant de tester aléatoirement la fonction `mot_de_passe` sur un grand nombre de cas.

Remarque : Les fonctions devront être documentées (docstrings), annotées et commentées.

TNSI	Langages et programmation	<b>Exercice - Générateur de mot de passes</b>
	Bases de la programmation	

Ce travail est à faire et à rendre sur pronote pour le **samedi 11 septembre** avant minuit.

On veut réaliser un générateur de mot de passes sécurisé. Pour cela on va procéder en plusieurs étapes.

- 1) Ecrire une fonction `chiffre()` qui renvoi un caractère (`str`) aléatoire entre "0" et "9".
- 2) Ecrire une fonction `majuscule()` qui renvoi un caractère (`str`) aléatoire entre "A" et "Z".
- 3) Ecrire une fonction `minuscule()` qui renvoi un caractère (`str`) aléatoire entre "a" et "z".
- 4) Ecrire une fonction `special()` qui renvoi un caractère (`str`) aléatoire parmi les caractères spéciaux de la table ascii standard.
- 5) Ecrire une fonction `mot_de_passe(n, c, M, m, s)` qui renvoi une chaîne de caractère contenant exactement `n` caractères et comprenant au moins `c` chiffres, `M` lettres majuscules, `m` lettres minuscules et `s` caractères spéciaux ascii. Si  $c + M + m + s < n$ , les caractères restants sont des minuscules. Les caractères imposés peuvent être placés n'importe où dans la chaîne.
- 6) Améliorer votre fonction `mot_de_passe` pour que tous les caractères soient mélangés.
- 7) Ecrire une fonction `contient_chiffres(c)` qui renvoi le nombre de chiffres contenus dans la chaîne `c`.
- 8) Faire de même pour les fonctions `contient_majuscules(c)`, `contient_minuscules(c)` et `contient_speciaux(c)`.
- 9) A partir des fonctions précédentes, écrire une fonction `tests()` permettant de tester aléatoirement la fonction `mot_de_passe` sur un grand nombre de cas.

Remarque : Les fonctions devront être documentées (docstrings), annotées et commentées.