

Master 2 Humanités Numériques – Introduction à Python

Interrogation – Durée : 30 minutes

(10 points pour le QCM, 10 points pour les exercices, +2 points bonus Zen of Python)

Durée : 30 min

Nom : _____

Date : _____

Partie 1 – QCM (10 points, 10 minutes)

Question 1 – Chaînes et listes

```
phrase = "Le numérique transforme les humanités"  
mots = phrase.split()  
print(len(mots))
```

Que va afficher le programme ?

- a) 5
- b) 4
- c) 6
- d) ["Le", "numérique", "transforme", "les", "humanités"]

Question 2 – Listes

```
auteurs = ["Matheson", "Beaumont", "King"]  
auteurs.append("Serling")  
auteurs.remove("Beaumont")  
print(auteurs)
```

Résultat attendu :

- a) ["Matheson", "Beaumont", "King", "Serling"]
- b) ["Matheson", "King", "Serling"]
- c) ["Matheson", "Serling", "King"]
- d) ["Serling", "Matheson", "King"]

Question 3 – Conditions et logique

```
mot = "Python"  
if "y" in mot and len(mot) > 5:  
    print("Mot valide")  
else:  
    print("Mot invalide")
```

- a) Mot valide
- b) Mot invalide
- c) Erreur
- d) Rien

Question 4 – Dictionnaires

```
texte = {"titre": "1984", "auteur": "George Orwell", "année": 1949}  
print(texte["auteur"])
```

- a) "Orwell" b) "George Orwell" c) "auteur" : "George Orwell" d) Erreur

Question 5 – Fonctions : print vs return

```
def affiche_mot(mot):  
    print(mot.upper())  
  
resultat = affiche_mot("humanités")  
print(resultat)
```

- a) HUMANITÉS b) None c) humanités d) Erreur

Question 6 – Classes et héritage

```
class Document:  
    def __init__(self, titre):  
        self.titre = titre  
  
class Article(Document):  
    def __init__(self, titre, auteur):  
        super().__init__(titre)  
        self.auteur = auteur  
  
art = Article("Humanités numériques", "Claire Martin")  
print(art.titre, "-", art.auteur)
```

- a) "Humanités numériques - Claire Martin"
b) "Claire Martin - Humanités numériques"
c) Erreur car la classe parente n'a pas d'auteur
d) Rien n'est affiché

Question 7 – Modules : random

```
import random  
mots = ["texte", "donnée", "analyse", "code"]  
mot = random.choice(mots)  
print(mot)
```

- a) Affiche toujours "texte"
b) Affiche un mot choisi aléatoirement
c) Affiche la liste complète
d) Erreur car random n'est pas défini

Question 8 – Traitement de texte : chaînes, boucles, conditions

```
texte = "Les humanités numériques transforment les Humanités, les données et la culture."  
compteur = 0  
for mot in texte.split():  
    if mot.lower().strip(",.") == "humanités":  
        compteur += 1  
print(compteur)
```

- a) 1 b) 2 c) 3 d) Erreur car strip() ne s'applique pas à une chaîne

Total QCM : 10 points

Partie 2 – Exercices (10 points, 20 minutes)

Exercice 1 – Gérer des données textuelles (3 points)

On dispose d'une liste de titres d'articles issus d'un corpus de recherche :

```
articles = [  
    "Les humanités numériques aujourd'hui",  
    "Analyse de données textuelles",  
    "Python pour les historien·ne·s",  
    "Introduction à la fouille de textes"  
]
```

1. Affiche le nombre total de titres dans la liste. (1 pt)
2. Affiche le premier et le dernier titre de la liste. (1 pt)
3. Trie la liste par ordre alphabétique et affiche-la. (1 pt)

Exercice 2 – Écrire la documentation d'une fonction (2 points)

Ajouter à la fonction suivante une **docstring**, des **commentaires**, et une **annotation de types** claire.

```
def extrait_mots(texte, n):  
  
    mots = texte.split()  
    extrait = mots[:n]  
    print(" ".join(extrait))
```

Exercice 3 – Les œuvres d'art et l'héritage en Python (5 points)

On souhaite modéliser une collection d'œuvres d'art pour un projet en Humanités numériques.

1. Crée une classe **Oeuvre** comportant un titre, une date et un lieu de création. Elle doit posséder une méthode **description()** affichant :

La Joconde - créé en 1503 à Florence

2. Crée une classe **ArtRenaissance** héritant de **Oeuvre**, ajoutant un attribut **courant** et une méthode **description_renaissance()** qui appelle la méthode parente avec **super()** puis affiche :

David - créé en 1504 à Florence.

Cette œuvre s'inscrit dans le courant Renaissance italienne.

3. Crée une classe **Sculpture** héritant de **ArtRenaissance**, ajoutant les attributs **artiste**, **pierre** et **technique**, et une méthode **details()** affichant :

Sculpture de Michel-Ange - marbre (taille directe)

Sortie attendue :

David - créé en 1504 à Florence. Cette œuvre s'inscrit dans le courant Renaissance italienne.

Sculpture de Michel-Ange - marbre (taille directe)