

Activité 1 :

1. ASCII : American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information)
2. ASCII utilise 7 bits. En pratique 1 octet est utilisé ; le bit de poids fort (bit de gauche) sert de somme de contrôle.
3. 4C 61 20 4E 53 49 20 65 73 74 20 31 20 73 75 70 65 72 20 64 69 73 63 69 70 6C 69 6E 65 21
4. L'ASCII ne permet pas de représenter les caractères accentués, les idéogrammes...

Activité 2 :

1. $2^8 = 256$ caractères en ISO 8859.
2. Il y a 16 tables ISO 8859.
3. La table ISO 8859-1 (Latin-1) est utilisée pour le français. On peut également se servir de sa révision ISO 8859-15 qui ajoute notamment le signe €.
4. C1 20 71 75 65 6C 6C 65 20 E2 67 65 20 61 70 70 72 65 6E 64 72 65 20 6C 61 20 4E 53 49 3F

Activité 3 :

1. Avec la norme ISO-10646 on a $2^{32} = 4294967296$ caractères possibles.
2. 32 bits représentent 4 octets.
3. point de code de é : 00E9. On représente les points de code avec le préfixe U+ : U+00E9. En binaire $E9 = 11101001$

é → 233 → E9

Activité 4 :

1. Il faut 2 octets pour encoder la lettre é.
2. On utilise deux octets : 110xxxxx 10xxxxxx. On place la représentation binaire de la lettre é et on complète avec des zéros : 11000011 10101001
3. — **ord** : Renvoie le nombre entier représentant le code Unicode du caractère représenté par la chaîne donnée.
— **hex** : Convertit un entier en chaîne hexadécimale préfixée de 0x.
4. Retrouver le point de code

```
1 def utf8(car: str)->str:  
2     return hex(ord(car))
```