

# Présentation

Informatique et Science du Numérique



# Présentation

Informatique et Science du Numérique



## Les 4 fondamentaux

1. **Données** (représentation de l'information)
2. **Algorithmique** (vite et bien)
3. **Langages** (vite et juste)
4. **Machine** (architecture)

# Présentation

## Informatique et Science du Numérique



### Le projet

But : imaginer des solutions qui répondent à l'expression d'un besoin. Son déroulement est basé sur une démarche projet :



1. définir le besoin ou le problème à résoudre



2. formaliser le besoin et concevoir des solutions



3. répartir les rôles et les tâches



4. réaliser tout ou partie de la solution choisie

5. évaluer les performances ou l'adéquation de la solution

6. évaluer le projet (retour d'expérience)

# Présentation

Informatique et Science du Numérique



## Proposition de sujets

- Stéganographie
- Transfert de fichiers par protocole kermit
- Jeu vidéo 2D
- Chiffrement (Caesar, Vigenere, RSA)
- Site web dynamique
- Compression de fichiers par codage de Huffman
- Robotique (challenge ITER)
- Analyse de l'information génétique
- Chat par liaison RS232
- ...

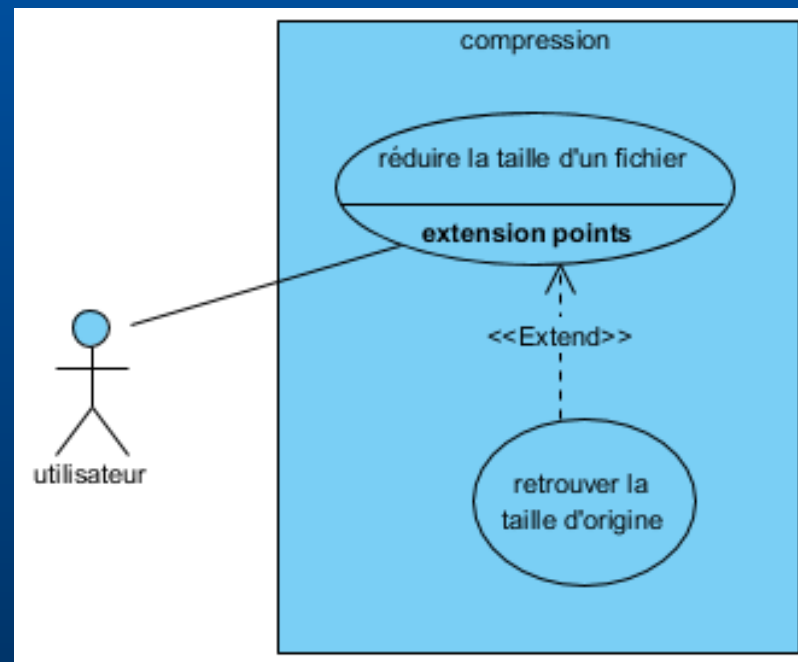
# Présentation

Informatique et Science du Numérique



## Cas d'utilisation

Définition du besoin



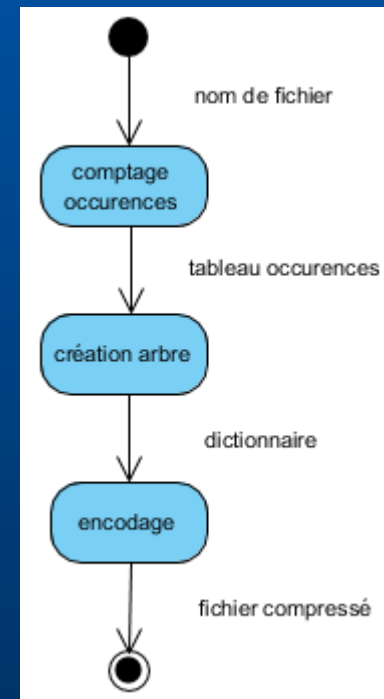
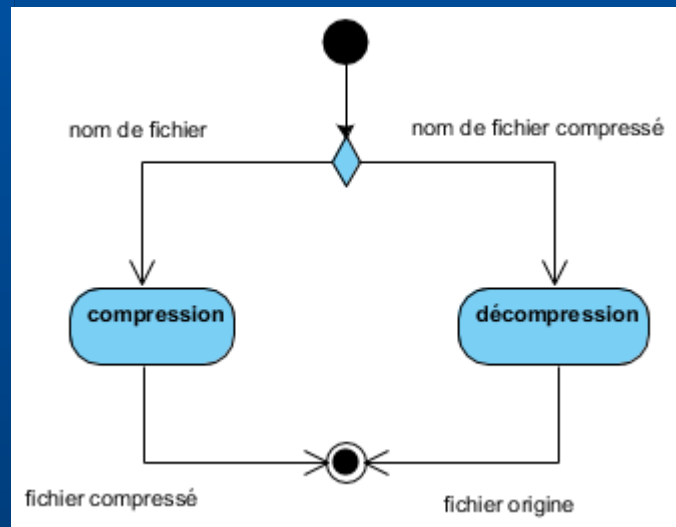
# Présentation

Informatique et Science du Numérique



## Diagramme d'activités

Formalisation et conception



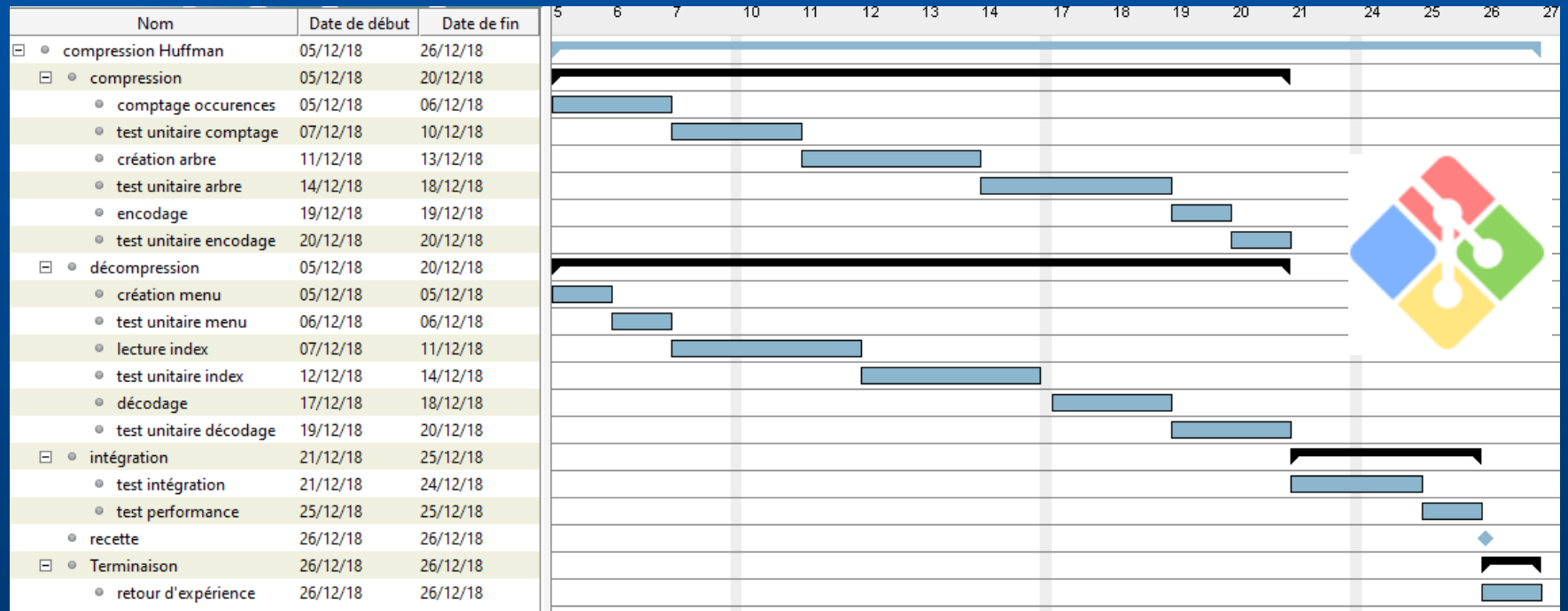
# Présentation

## Informatique et Science du Numérique



### Planification

Qui fait quoi ?



# Présentation

Informatique et Science du Numérique



## Algorithme

Choisir une fonctionnalité

```
fonction compte_occurrences(texte : chaîne, car : caractère) : entier
Début
    total := 0
    pour chaque lettre dans texte faire
        si (lettre = car) alors
            total := total + 1
    retourner total
Fin
```





# Présentation

## Informatique et Science du Numérique

### Implémentation

```
def compte_occurrences(texte : str, car : str) -> int:
```

```
    """
```

```
    compte les occurrences d'une lettre dans un texte
```

```
    Arguments:
```

```
    texte -- texte de recherche
```

```
    car -- lettre à chercher
```

```
    Retour:
```

```
    nombre d'occurrences de car dans texte
```

```
    """
```

```
    if type(texte) != str or type(car) != str:
```

```
        return 0
```

```
    total = 0
```

```
    for lettre in texte :
```

```
        if lettre == car:
```

```
            total = total + 1
```

```
    return total
```

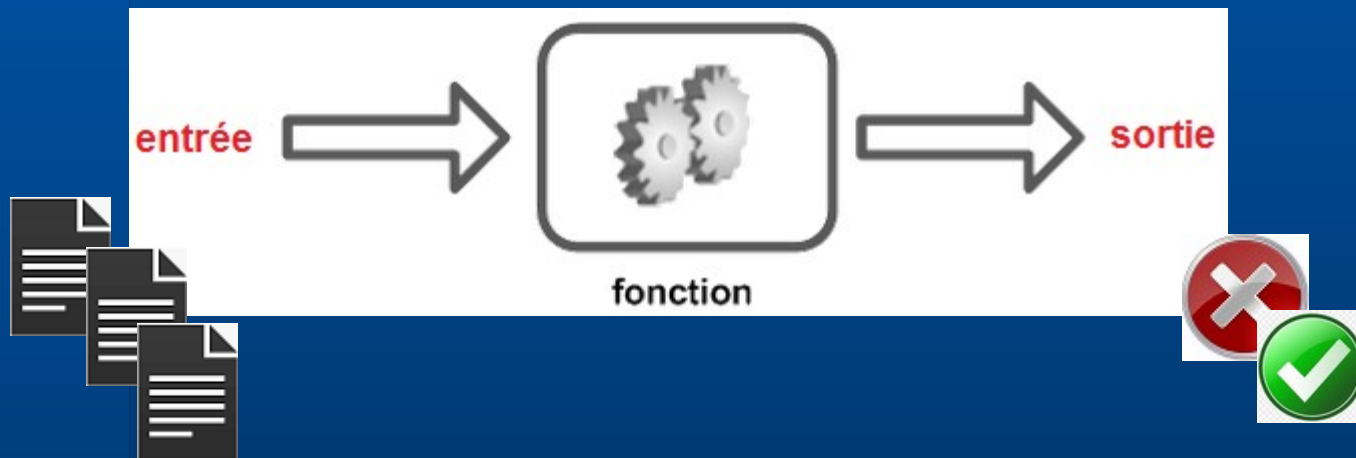
# Présentation

Informatique et Science du Numérique



## Les tests unitaires

Une série de tests ne prouvera jamais qu'une fonction est correcte



Classes de tests

FAILED ou PASSED

# Présentation

Informatique et Science du Numérique



## Implémentation tests unitaires

```
assert compte_occurrences("", 'a') == 0
```

```
assert compte_occurrences("a", 'a') == 1
```

```
assert compte_occurrences("aba", 'a') == 2
```

```
assert compte_occurrences("aba", 1) == 0
```

```
assert compte_occurrences(123456, 'a') == 0
```

```
assert compte_occurrences(123456, 1) == 0
```

```
assert compte_occurrences(123456, "") == 0
```

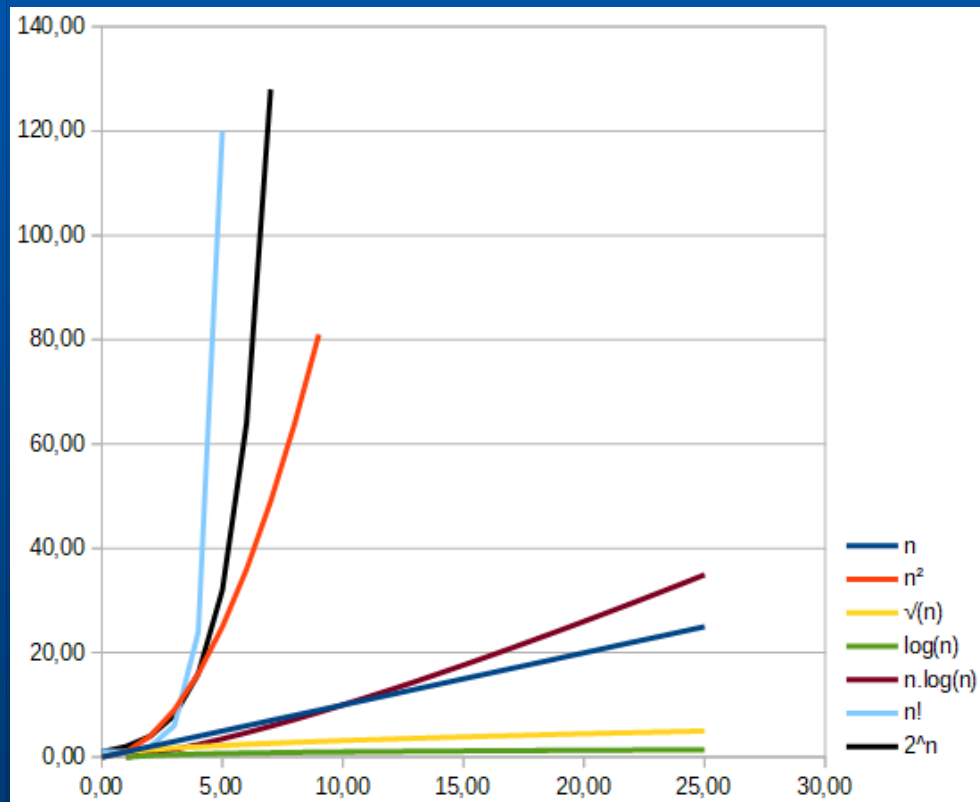
```
assert compte_occurrences("aba", "") == 0
```

# Présentation

Informatique et Science du Numérique



## Tests de performance



# Présentation

Informatique et Science du Numérique



## Les langages

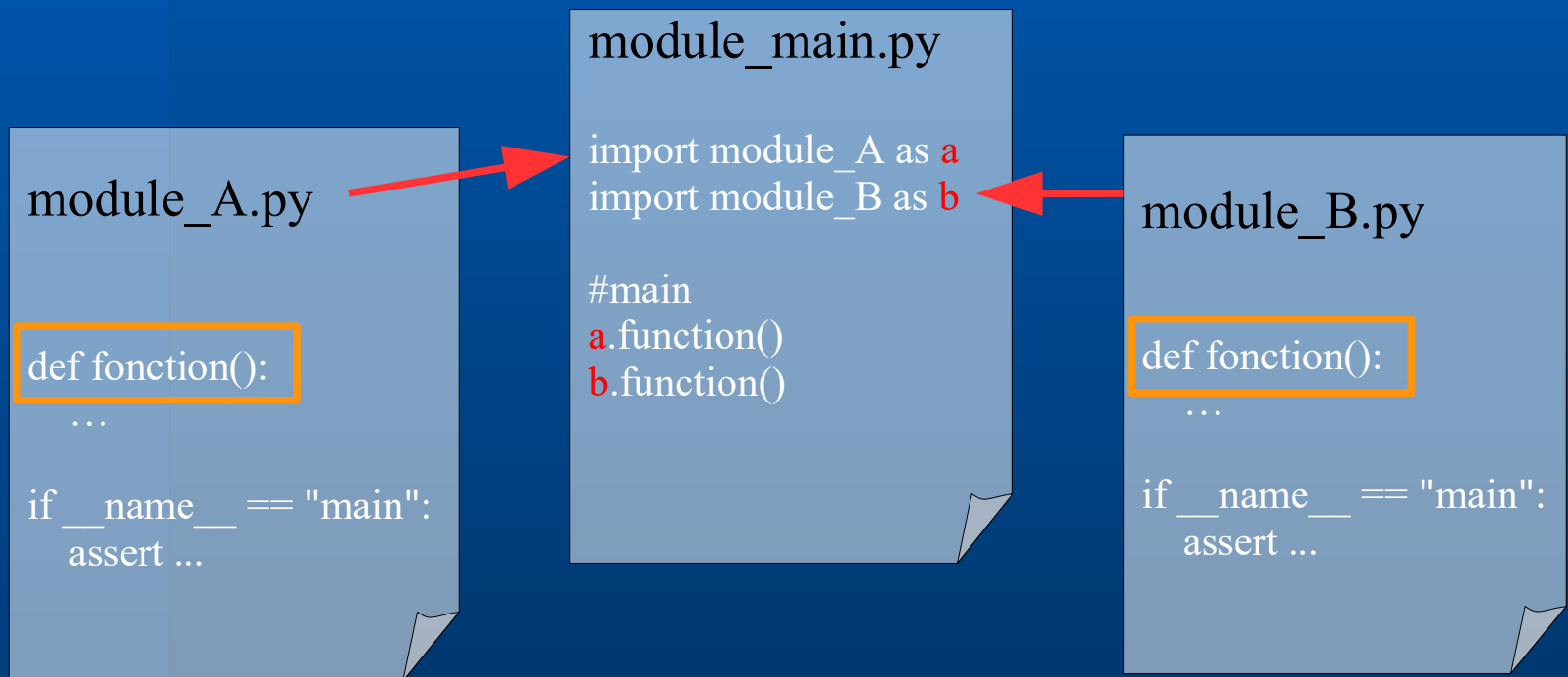


# Présentation

Informatique et Science du Numérique



## Programmation modulaire



instruction **import**

# Présentation

Informatique et Science du Numérique



## Évaluation

8 pts revues de projet

12 pts épreuve bac

- \* 10' exposé

- \* 10' questions

# Présentation

Informatique et Science du Numérique



A vous de jouer !!