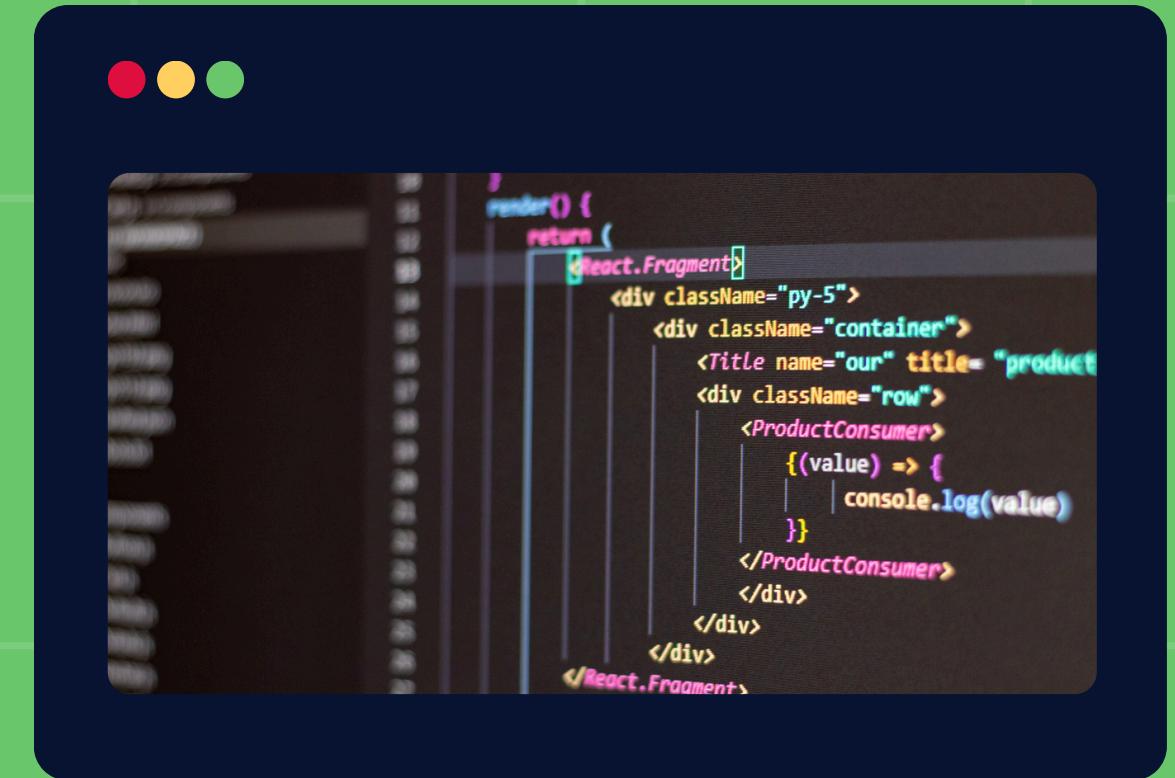
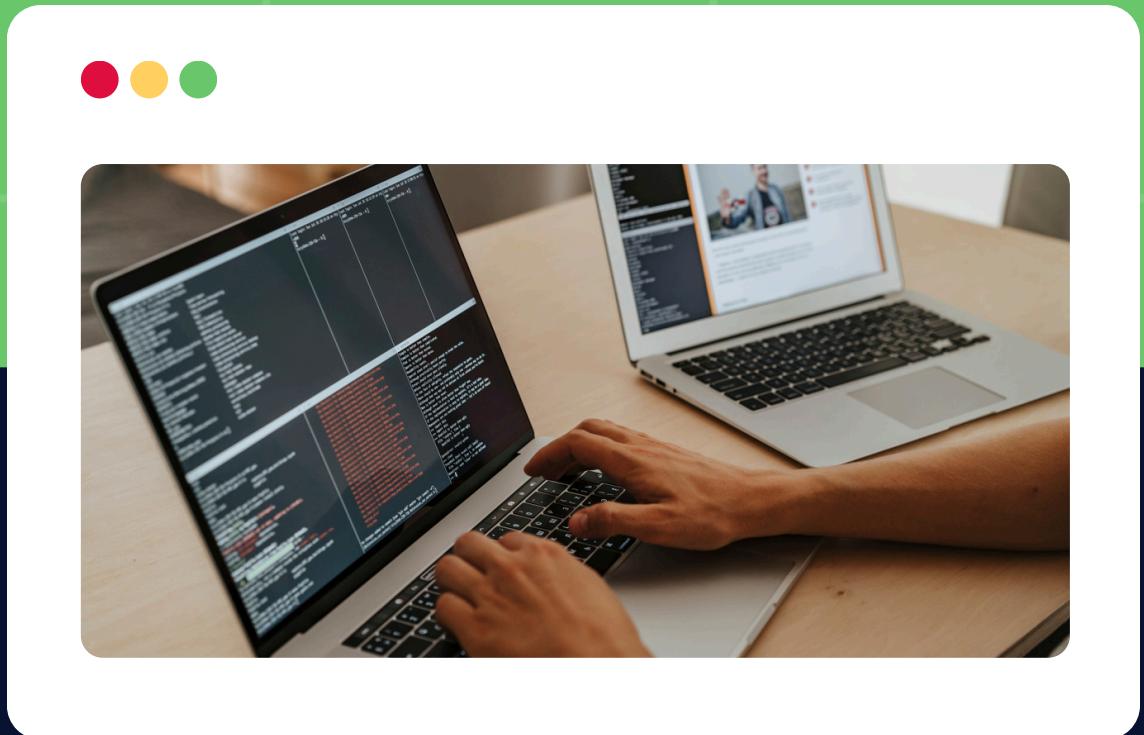




Seance 3: Python



A screenshot of a code editor window showing a React component. The code is as follows:

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title= "product" />
          <div className="row">
            <ProductConsumer>
              {(value) => {
                console.log(value)
              }}
            </ProductConsumer>
          </div>
        </div>
      <React.Fragment>
    
```

Laffet takwa



Exercices (partie 1)

Créer un List représentant 3
livre (titre, auteur, année).
Modifier l'année de 2e Livre

Dictionnaires

Un dictionnaire est une collection d'éléments sous forme de clé:valeur.

```
etudiant = {  
    "nom": "Sara",  
    "age": 21,  
    "filiere": "Informatique"  
}
```

```
print(etudiant["nom"]) # Ali  
print(etudiant.get("age")) # 20
```

Dictionnaires

#Ajouter un élément

```
etudiant["filiere"] = "Informatique"  
print(etudiant)  
# {'nom': 'Ali', 'age': 20, 'ville':  
'Tunis', 'filiere': 'Informatique'}
```

#Supprimer un élément

```
del etudiant["ville"]  
print(etudiant)  
# {'nom': 'Ali', 'age': 21,  
'filiere': 'Informatique'}
```

```
#Modifier une valeur  
etudiant["age"] = 21  
print(etudiant)  
# {'nom': 'Ali', 'age': 21, 'ville':  
'Tunis', 'filiere': 'Informatique'}
```

#Supprimer un élément

```
filiere = etudiant.pop("filiere")  
print(etudiant)    # {'nom': 'Ali', 'age':  
21}  
print(filiere)    # Informatique
```

Dictionnaires

```
#Supprimer un élément  
etudiant.popitem()  
print(etudiant)  
#{'nom': 'Ali'}
```

```
#Vider complètement le  
dictionnaire  
etudiant.clear()  
print(etudiant)  #{'}
```

```
etudiant.keys()      # renvoie toutes les clés  
etudiant.values()   # renvoie toutes les valeurs  
etudiant.items()    # renvoie clés et valeurs
```

Les Ensembles (set)

Un ensemble est une collection non ordonnée et sans doublons.

Idéal pour les opérations mathématiques (union, intersection, etc.).

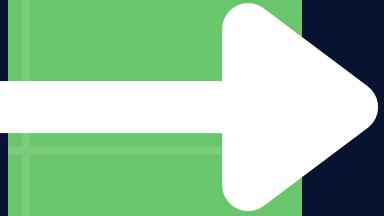
```
nombres = {1, 2, 3, 3, 4}  
print(nombres)  # {1, 2, 3, 4}  
  
A = {1, 2, 3}  
B = {3, 4, 5}  
  
print(A | B)  # union -> {1, 2, 3, 4, 5}  
print(A & B)  # intersection -> {3}  
print(A - B)  # différence -> {1, 2}
```

Exercices

1. Créer un dictionnaire représentant un livre (titre, auteur, année).
2. Modifier l'année et ajouter une clé pages.
3. Créer deux ensembles A et B puis afficher leur union et intersection.

Fonctions

Une fonction est un bloc de code qui exécute une tâche spécifique.



```
def saluer():
    print("Bonjour !")
saluer()
```

Fonctions avec paramètres

```
def addition(a, b):  
    return a + b  
  
print(addition(3, 5))
```

#Paramètres par défaut

```
def dire_bonjour(nom="Utilisateur"):  
    print("Bonjour", nom)  
  
dire_bonjour("Ali")      # Bonjour Ali  
dire_bonjour()           # Bonjour Utilisateur
```

#Valeurs de retour multiples

```
def calculs(a, b):  
    return a+b, a-b, a*b  
  
somme, diff, prod = calculs(10, 5)  
print(somme, diff, prod)
```

Exercices



- Crée une fonction qui retourne le carré d'un nombre.
- Crée une fonction qui calcule la moyenne d'une liste de notes.
- Crée une fonction qui retourne le plus grand de deux nombres.

Gestion des erreurs



A. Problème

Sans gestion :

```
x = int("abc") # ValueError
```

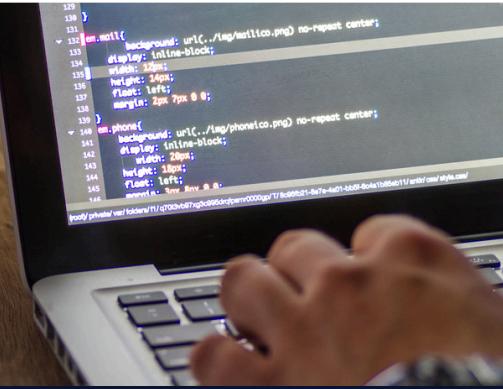
Structure de base

```
try:  
    x = int("abc") #ValueError  
except:  
    print("⚠ Erreur !")
```

```
try:  
    f = open("fichier.txt")  
except FileNotFoundError:  
    print("Fichier introuvable")  
finally:  
    print("Programme terminé")
```

Exercices

- Demander un nombre à l'utilisateur et gérer s'il entre une chaîne.
- Écrire une fonction de division qui gère la division par zéro.



Thank You

