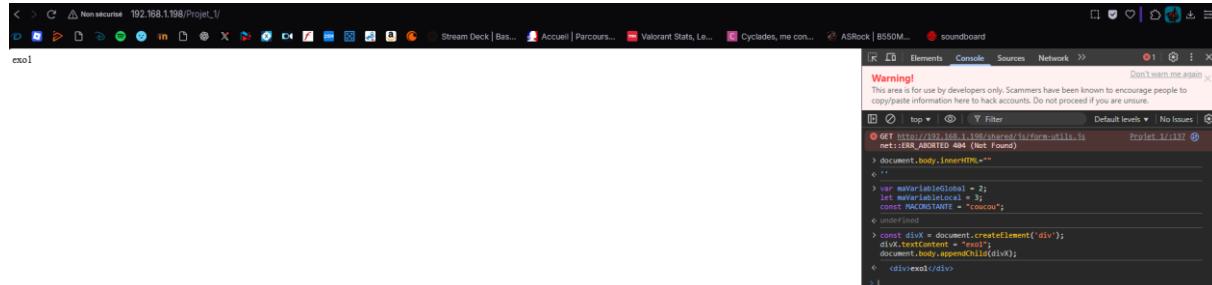


Exo 1

1)

J'ai testé les différentes commandes sur un superbe site internet :)



The screenshot shows a browser window with multiple tabs open. The active tab is a local file at '192.168.1.198/Projet_1/'. The developer tools console is open, displaying a warning message: "Warning! This area is for use by developers only. Scammers have been known to encourage people to copy/paste information here to hack accounts. Do not proceed if you are unsure." Below the warning, the console output shows the following JavaScript code:

```
> document.body.innerHTML=""
< ''
> var maVariableGlobal = 2;
let maVariableLocal = 3;
const MACONSTANTE = "coucou";
< undefined
> const divX = document.createElement('div');
divX.textContent = "exo1";
document.body.appendChild(divX);
<   <div>exo1</div>
>
```

J'ai créé une div nomé divX, et je lui ai mis du texte qui est : exo1. Ce qui fait apparaître le texte sur notre page.

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Non secure 192.168.1.198/Project_1". In the bottom right corner of the browser, there is a developer tools panel. A warning message is displayed: "Warning! This area is for use by developers only. Scammers have been known to encourage people to copy/paste information here to hack accounts. Do not proceed if you are unsure." Below the warning, the developer tools' "Console" tab is selected, showing the following JavaScript code and its execution results:

```
> console.log("la variable est" + maVariableGlobal);
document.body.appendChild(divX);

la variable est2
VM2166:1

< <div>exo1</div>

> divX.textContent = "exo1" + maVariableGlobal + " " + maVariableLocal + "
" + MACONSTANTE;
document.body.appendChild(divX);

< <div>exo12 3 coucou</div>
```

The code uses string concatenation to build a larger string and append it to a new `div` element. The resulting output is visible in the browser's DOM, showing the concatenated text "exo1 maVariableGlobal " " maVariableLocal " "+ MACONSTANTE" followed by the text "exo12 3 coucou". There are several syntax errors highlighted in red, such as "Uncaught SyntaxError: missing) after argument list" and "Uncaught SyntaxError: Unexpected identifier 'maVariableGlobal'", which correspond to the lines where the string is being built.

Il apparaît clairement dans la console ainsi que dans notre div le texte « exo1 » et les valeurs de nos variables.

2)

The screenshot shows a Jupyter Notebook interface with two code cells and their outputs.

Code Cell 1:

```
TP2 > tp2.py ...  
1 maVariable = 2  
2 MACONSTANTE = 3  
3 monText = "Julien Code"  
4  
5 """je me suis arrêté à la question 2 de l'exercice 1"""
```

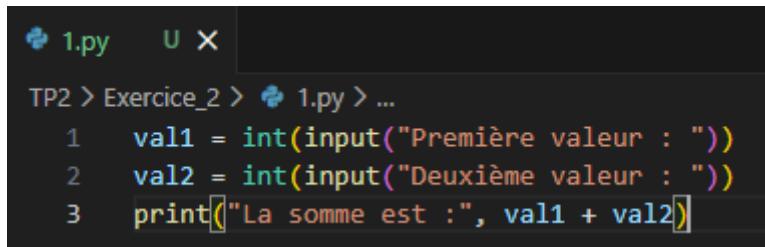
Code Cell 2:

```
Connected to Python 3.9.2  
✓ maVariable = 2 -  
... "je me suis arrêté à la question 2 de l'exercice 1"  
  
print(maVariable)  
✓ 0s  
-- 2  
  
print(MACONSTANTE)  
✓ 0s  
-- 3
```

A status bar at the bottom right indicates: **Press [Enter] to execute.**

Exo 2

1)



```
1.py X
TP2 > Exercice_2 > 1.py > ...
1 val1 = int(input("Première valeur : "))
2 val2 = int(input("Deuxième valeur : "))
3 print("La somme est :", val1 + val2)
```

Mes deux variables ont été créées en indiquant que c'est une entrée (input). Une fois que l'utilisateur a entré ses deux valeurs, le programme affichera la somme des deux valeurs entrées.

J'ai fait deux tests avec comme valeur 15 et 15. Ce qui est censé donner 30 :



Interactive-1 X

Interrupt | Clear All | Restart | Jupyter Variables | Save | Export | Expand | ... | Python 3.9.2

Connected to Python 3.9.2

```
✓ val1 = int(input("Première valeur : ")) ...
...
... La somme est : 30
```

Deuxième test avec comme valeur : 22 et 23



Interactive-1 X

Interrupt | Clear All | Restart | Jupyter Variables | Save | Export | Expand | ... | Python 3.9.2

Connected to Python 3.9.2

```
✓ val1 = int(input("Première valeur : ")) ...
...
... La somme est : 30
```



```
✓ val1 = int(input("Première valeur : ")) ...
...
... La somme est : 45
```

2)

The screenshot shows a browser developer tools console with the following code and output:

```
> const inputA = document.createElement('input');
const inputB = document.createElement('input');
const ButtonA = document.createElement('button');

document.body.appendChild(inputA);
document.body.appendChild(inputB);
document.body.appendChild(ButtonA);

ButtonA.onclick=> {
    const somme = inputA.value + inputB.value;
    console.log("La somme des valeurs entrées est : " + somme);
};

< () => {
    const somme = inputA.value + inputB.value;
    console.log("La somme des valeurs entrées est : " + somme);
}

La somme des valeurs entrées est : 1212
```

The code creates two input fields and a button. The button's onclick event handler adds the values of the two inputs and logs the result. However, it also logs the same result again immediately below it due to a bug in the code structure. The output shows the expected result of 1212.

Ici ça ne fonctionne pas vraiment comme c'est attendu... A revoir