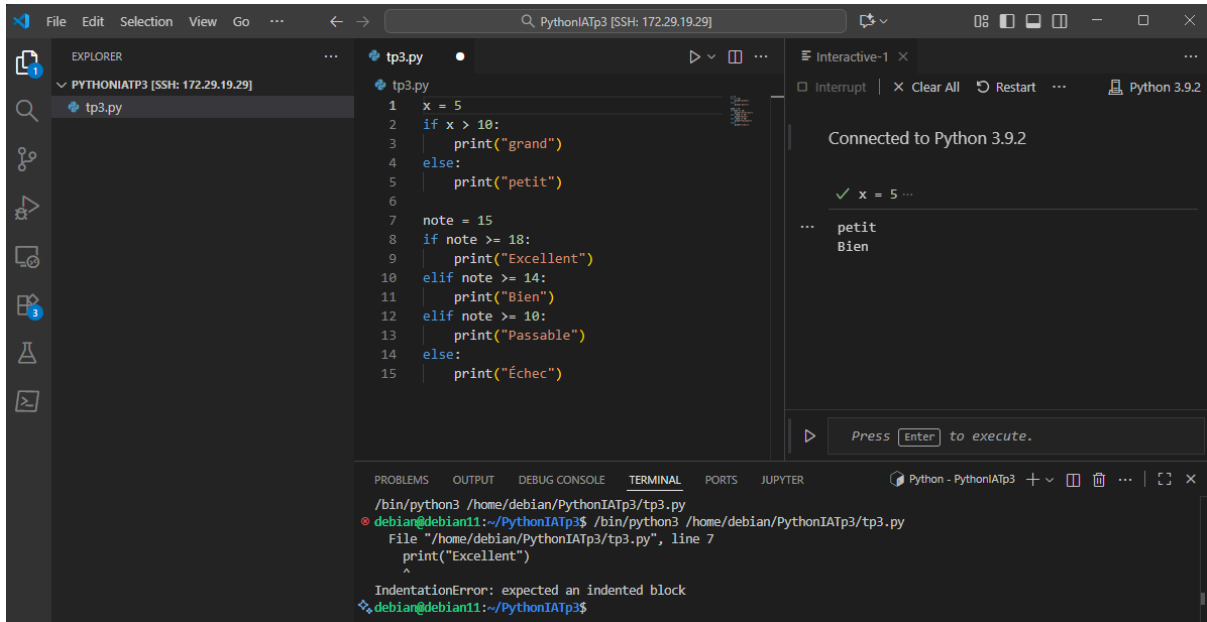


Co Inter Math-Info

TP3 IA

Exemple :

Python : Les condition si sinon



The screenshot shows a Jupyter Notebook interface with a Python script in a cell. The script defines a variable `x` with the value 5 and a variable `note` with the value 15. It uses `if`, `elif`, and `else` statements to print different messages based on the values of `x` and `note`. The output of the cell shows the execution results: `x = 5` is printed, followed by `petit` and `Bien`. The terminal at the bottom shows the command `/bin/python3 /home/debian/PythonIATp3/tp3.py` and the output of the script.

```

1 x = 5
2 if x > 10:
3     print("grand")
4 else:
5     print("petit")
6
7 note = 15
8 if note >= 18:
9     print("Excellent")
10 elif note >= 14:
11     print("Bien")
12 elif note >= 10:
13     print("Passable")
14 else:
15     print("Échec")
  
```

Connected to Python 3.9.2

✓ x = 5 ...

petit
Bien

Press [Enter] to execute.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

/bin/python3 /home/debian/PythonIATp3/tp3.py

debian@debian11:~/PythonIATp3\$ /bin/python3 /home/debian/PythonIATp3/tp3.py


File ~/home/debian/PythonIATp3/tp3.py, line 7

```

print("Excellent")
^
IndentationError: expected an indented block
  
```

debian@debian11:~/PythonIATp3\$

Js: Les condition si sinon



The screenshot shows a code editor with JavaScript code. The code defines a variable `x` with the value 5 and a variable `note` with the value 15. It uses `if`, `else if`, and `else` statements to log different messages based on the values of `x` and `note`. The output of the code shows the execution results: `x = 5` is logged, followed by `petit` and `Bien`.

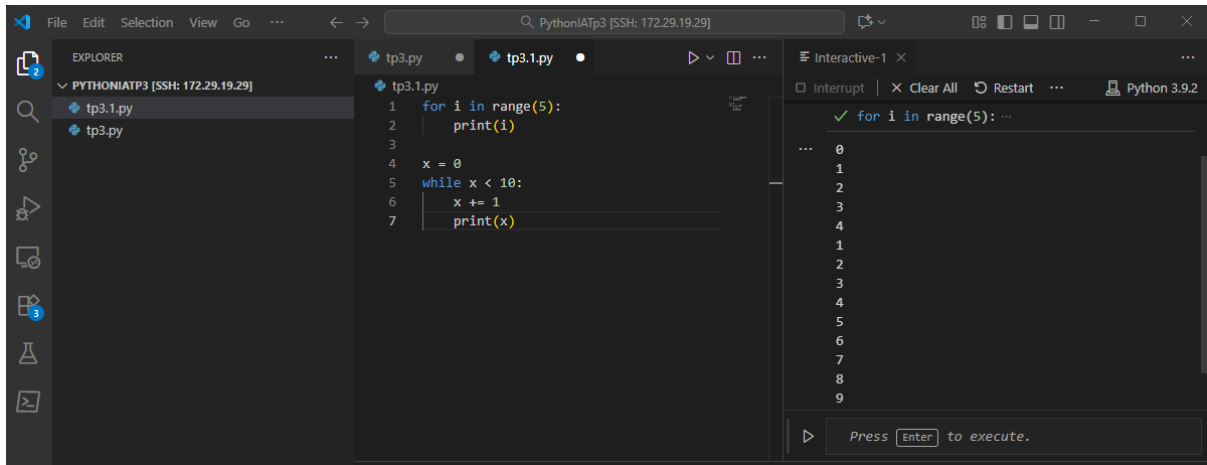
```

if (x > 10) {
  console.log('grand');
} else {
  console.log('petit');
}

let note = 15;
if (note >= 18) {
  console.log("Excellent");
} else if (note >= 14) {
  console.log("Bien");
} else if (note >= 10) {
  console.log("Passable");
} else {
  console.log("Échec");
}

petit
Bien
  
```

Python : Les boucles



The screenshot shows a Jupyter Notebook with two cells. The first cell contains a for loop that prints numbers 0 to 4. The second cell contains a while loop that prints numbers 0 to 9. The output of the first cell is shown in the console, and the output of the second cell is shown in the interactive window.

```

1 for i in range(5):
2     print(i)
3
4 x = 0
5 while x < 10:
6     x += 1
7     print(x)
  
```

Output of the first cell:

```

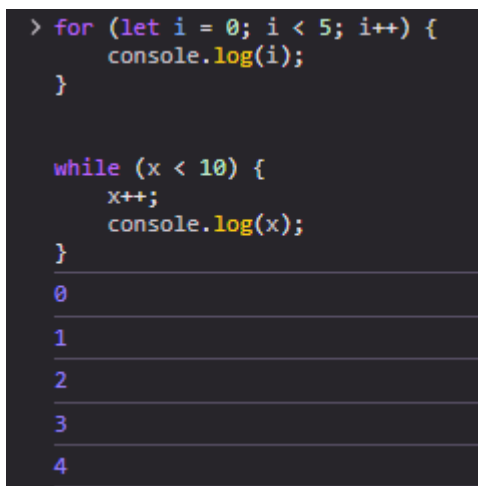
0
1
2
3
4
  
```

Output of the second cell:

```

0
1
2
3
4
5
6
7
8
9
  
```

Js : Les boucle



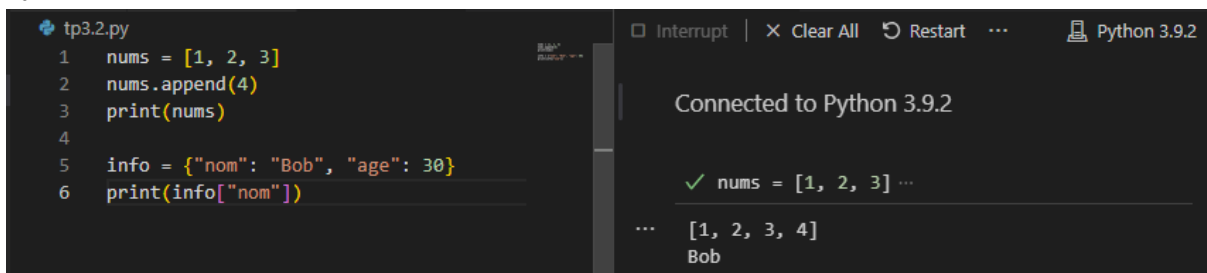
The screenshot shows a JavaScript console with two code blocks. The first block is a for loop that logs numbers 0 to 4. The second block is a while loop that logs numbers 0 to 4. The output of the first block is shown in the console, and the output of the second block is shown in the console.

```

> for (let i = 0; i < 5; i++) {
  console.log(i);
}

while (x < 10) {
  x++;
  console.log(x);
}
0
1
2
3
4
  
```

Python : Les collection de tableau



The screenshot shows a Jupyter Notebook with two cells. The first cell contains code to create a list 'nums' with elements [1, 2, 3], append the value 4, and print the list. The second cell contains code to create a dictionary 'info' with keys 'nom' and 'age', and print the value of 'nom'. The output of the first cell is shown in the console, and the output of the second cell is shown in the console.

```

1 nums = [1, 2, 3]
2 nums.append(4)
3 print(nums)
4
5 info = {"nom": "Bob", "age": 30}
6 print(info["nom"])
  
```

Output of the first cell:

```

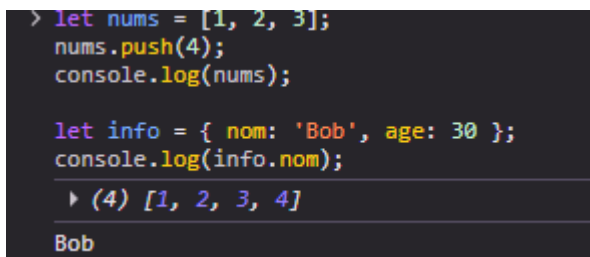
[1, 2, 3, 4]
  
```

Output of the second cell:

```

Bob
  
```

Js : Les collection de tableau



The screenshot shows a JavaScript console with two code blocks. The first block is an array 'nums' with elements [1, 2, 3], push the value 4, and log the array. The second block is an object 'info' with keys 'nom' and 'age', and log the value of 'nom'. The output of the first block is shown in the console, and the output of the second block is shown in the console.

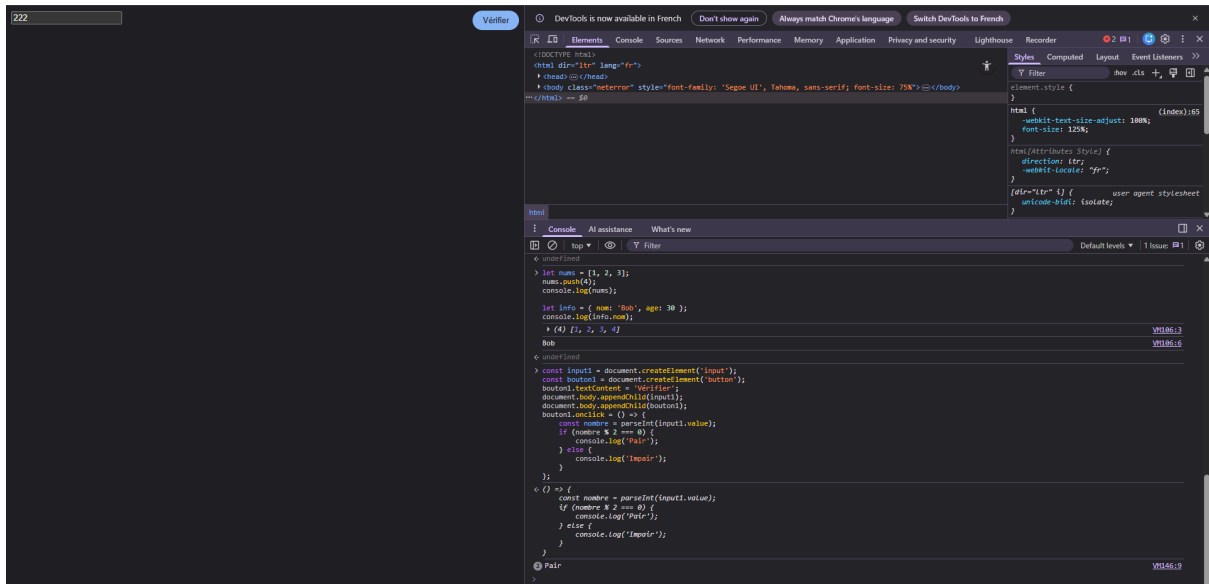
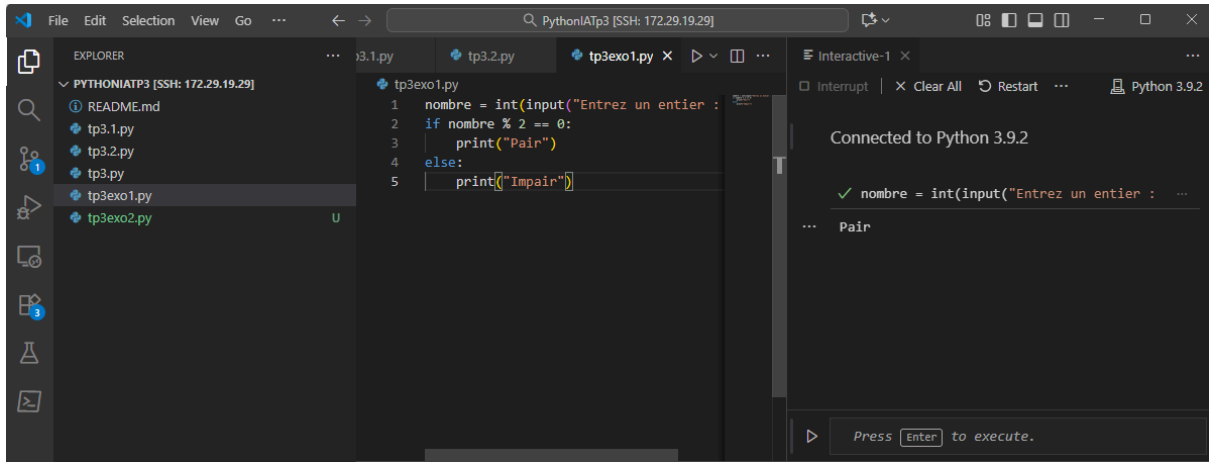
```

> let nums = [1, 2, 3];
  nums.push(4);
  console.log(nums);

let info = { nom: 'Bob', age: 30 };
console.log(info.nom);

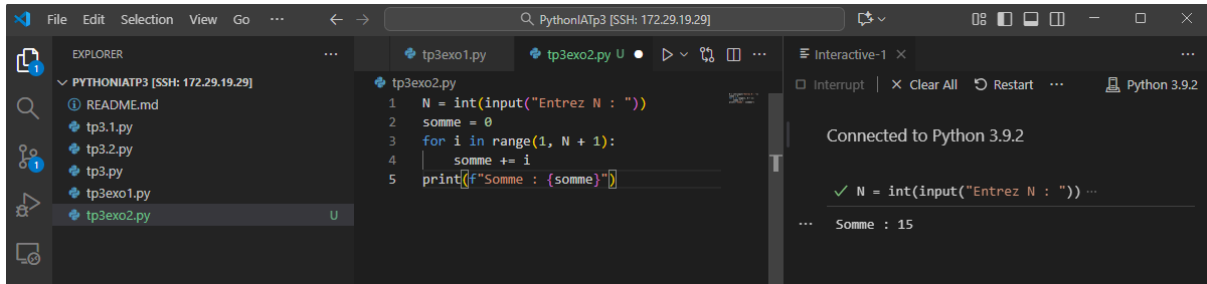
(4) [1, 2, 3, 4]
Bob
  
```

Exercice 1 (Jaune)



```
const input1 = document.createElement('input');
const bouton1 = document.createElement('button');
bouton1.textContent = 'Vérifier';
document.body.appendChild(input1);
document.body.appendChild(bouton1);
bouton1.onclick = () => {
    const nombre = parseInt(input1.value);
    if (nombre % 2 === 0) {
        console.log('Pair');
    } else {
        console.log('Impair');
    }
};
```

Exercice 2 (Bleu)



```

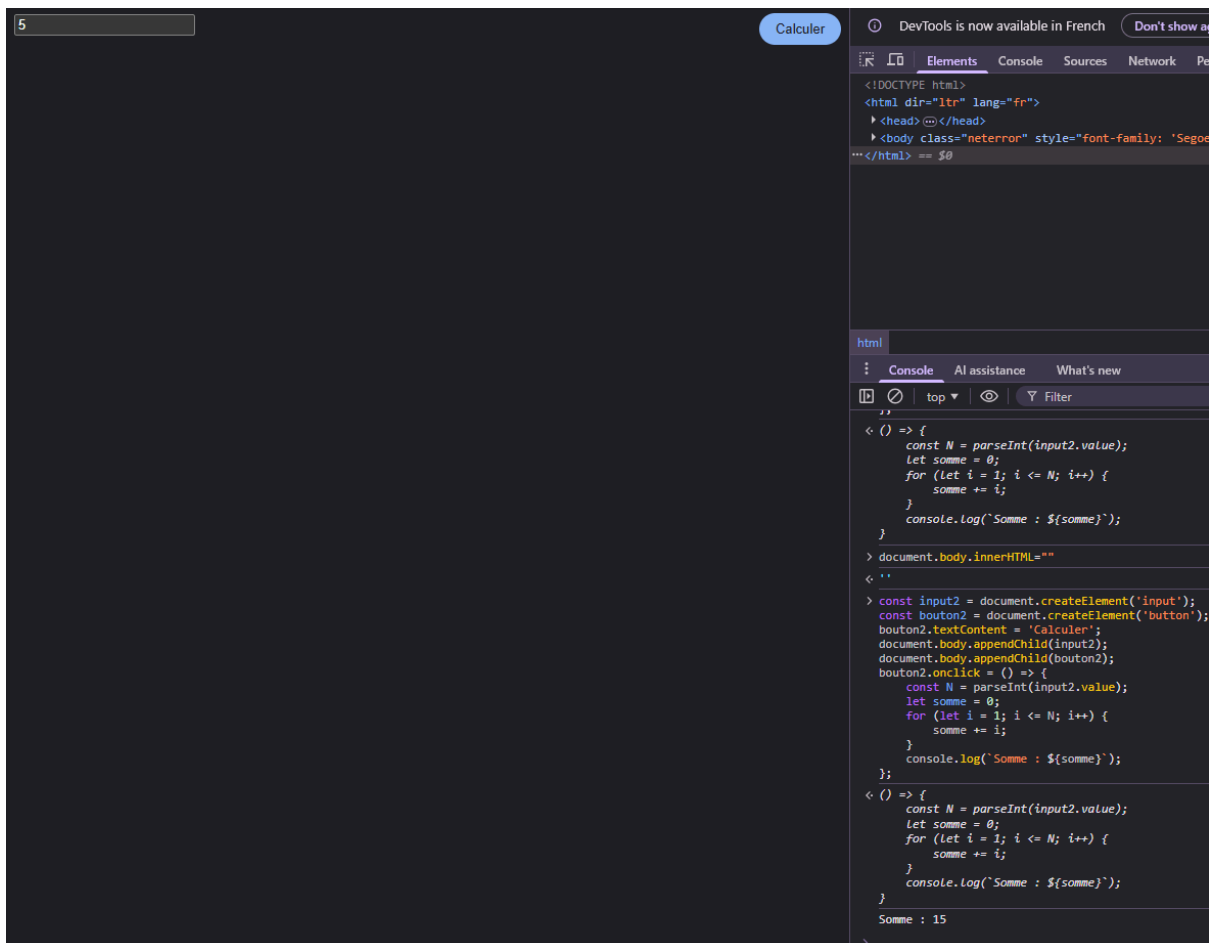
1 N = int(input("Entrez N : "))
2 somme = 0
3 for i in range(1, N + 1):
4     somme += i
5 print(f"Somme : {somme}")

```

Connected to Python 3.9.2

✓ N = int(input("Entrez N : ")) ...

... Somme : 15



```

<!DOCTYPE html>
<html dir="ltr" lang="fr">
  <head>
    <body class="neterror" style="font-family: 'Segoe UI', sans-serif;">

```

```

<() => {
  const N = parseInt(input2.value);
  let somme = 0;
  for (let i = 1; i <= N; i++) {
    somme += i;
  }
  console.log(`Somme : ${somme}`);
}
> document.body.innerHTML=""
<
> const input2 = document.createElement('input');
const bouton2 = document.createElement('button');
bouton2.textContent = 'Calculer';
document.body.appendChild(input2);
document.body.appendChild(bouton2);
bouton2.onclick = () => {
  const N = parseInt(input2.value);
  let somme = 0;
  for (let i = 1; i <= N; i++) {
    somme += i;
  }
  console.log(`Somme : ${somme}`);
};
<() => {
  const N = parseInt(input2.value);
  let somme = 0;
  for (let i = 1; i <= N; i++) {
    somme += i;
  }
  console.log(`Somme : ${somme}`);
}
Somme : 15

```

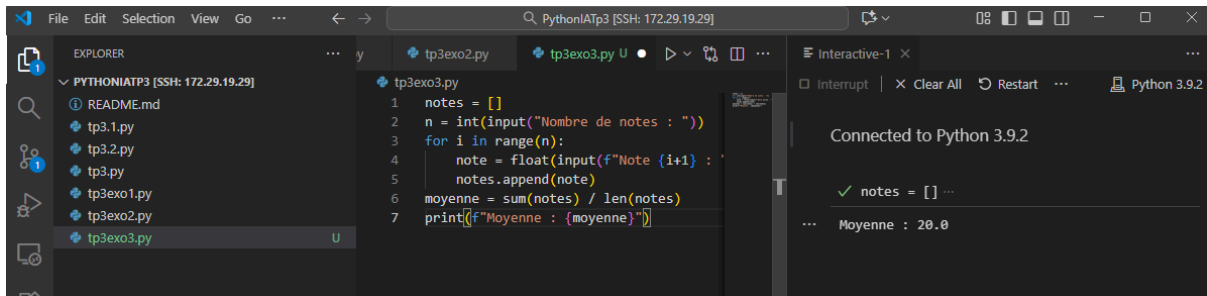
```

const input2 = document.createElement('input');
const bouton2 = document.createElement('button');
bouton2.textContent = 'Calculer';
document.body.appendChild(input2);
document.body.appendChild(bouton2);
bouton2.onclick = () => {
  const N = parseInt(input2.value);
  let somme = 0;
  for (let i = 1; i <= N; i++) {
    somme += i;
  }
  console.log(`Somme : ${somme}`);
}

```

};

Exercice 3 (Bleu) si c'est Optimisé (Vert)

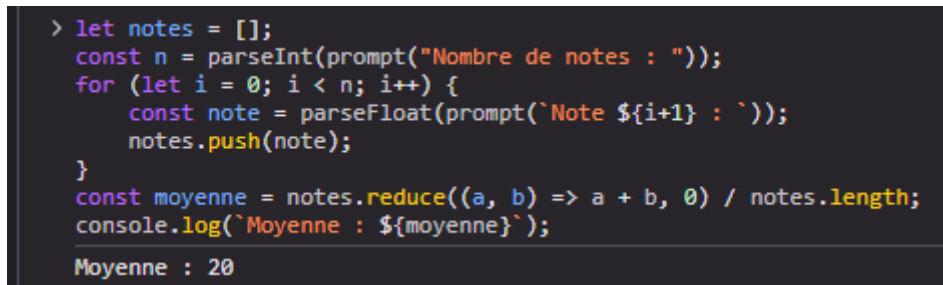


The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal on the right. The file explorer shows a project named 'PYTHONIATP3 [SSH: 172.29.19.29]' with files 'tp3.1.py', 'tp3.2.py', 'tp3.py', 'tp3exo1.py', 'tp3exo2.py', and 'tp3exo3.py'. The code editor shows the content of 'tp3exo3.py' with the following Python code:

```
1 notes = []
2 n = int(input("Nombre de notes : "))
3 for i in range(n):
4     note = float(input(f"Note {i+1} : "))
5     notes.append(note)
6 moyenne = sum(notes) / len(notes)
7 print(f"Moyenne : {moyenne}")
```

The terminal on the right shows the output of the script:

```
Connected to Python 3.9.2
✓ notes = [] ...
... Moyenne : 20.0
```



The screenshot shows a terminal window with the following JavaScript code:

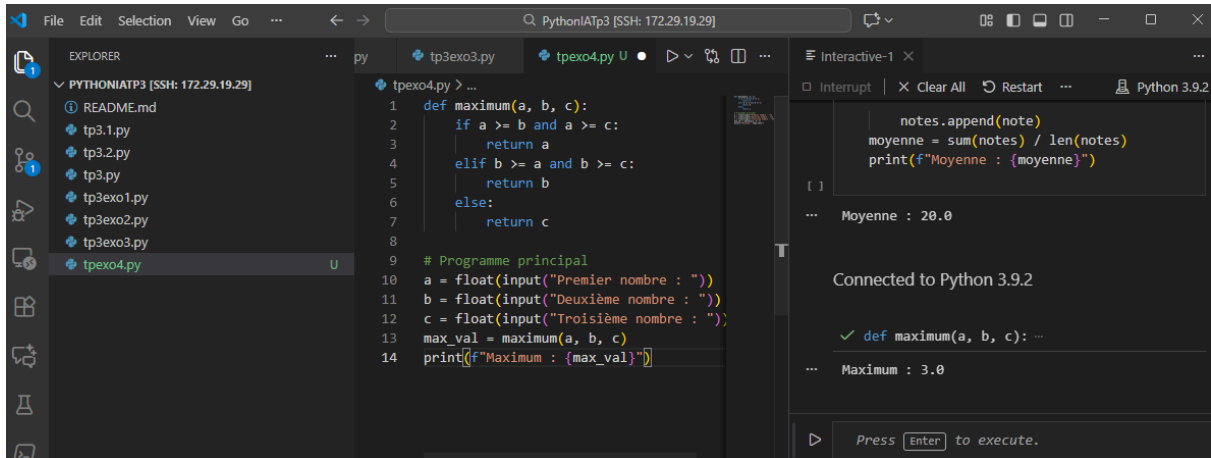
```
> let notes = [];
const n = parseInt(prompt("Nombre de notes : "));
for (let i = 0; i < n; i++) {
    const note = parseFloat(prompt(`Note ${i+1} : `));
    notes.push(note);
}
const moyenne = notes.reduce((a, b) => a + b, 0) / notes.length;
console.log(`Moyenne : ${moyenne}`);
```

The output of the script is:

```
Moyenne : 20
```

```
let notes = [];
const n = parseInt(prompt("Nombre de notes : "));
for (let i = 0; i < n; i++) {
    const note = parseFloat(prompt(`Note ${i+1} : `));
    notes.push(note);
}
const moyenne = notes.reduce((a, b) => a + b, 0) / notes.length;
console.log(`Moyenne : ${moyenne}`);
```

Exercice 4 (Vert)



```

1 def maximum(a, b, c):
2     if a >= b and a >= c:
3         return a
4     elif b >= a and b >= c:
5         return b
6     else:
7         return c
8
9 # Programme principal
10 a = float(input("Premier nombre : "))
11 b = float(input("Deuxième nombre : "))
12 c = float(input("Troisième nombre : "))
13 max_val = maximum(a, b, c)
14 print(f"Maximum : {max_val}")

```

```

> function maximum(a, b, c) {
    if (a >= b && a >= c) {
        return a;
    } else if (b >= a && b >= c) {
        return b;
    } else {
        return c;
    }
}

// Programme principal
const a1 = parseFloat(prompt("Premier nombre : "));
const b1 = parseFloat(prompt("Deuxième nombre : "));
const c1 = parseFloat(prompt("Troisième nombre : "));
const maxVal1 = maximum(a1, b1, c1);
console.log(`Maximum : ${maxVal1}`);

Maximum : 1000
< undefined

```

```

function maximum(a, b, c) {
    if (a >= b && a >= c) {
        return a;
    } else if (b >= a && b >= c) {
        return b;
    } else {
        return c;
    }
}

```

```

// Programme principal
const a1 = parseFloat(prompt("Premier nombre : "));
const b1 = parseFloat(prompt("Deuxième nombre : "));
const c1 = parseFloat(prompt("Troisième nombre : "));
const maxVal1 = maximum(a1, b1, c1);
console.log(`Maximum : ${maxVal1}`);

```