

# UE12P25 – AP

## Programmation Elixir

Quiz du cours #1

Page	Possible	Score
1	8	
2	8	
3	4	
Total:	20	

Cochez une seule réponse par question.

Les questions 1–12 sont plutôt accessibles, 13–20 un peu plus techniques.

1. Comment affiche-t-on un texte à l'écran en Elixir ? 1  5  1
- ☐ `print("Hello")` ☐ Erreur
- ☐ `echo("Hello")` ☐ 1
- ☐ `IO.puts("Hello")` ☐ 2
- ☐ `Console.log("Hello")` ☐ [1, 2]
2. Lequel de ces éléments est une liste en Elixir ? 2  1
- ☐ {1, 2, 3}
- ☐ [1, 2, 3]
- ☐ (1, 2, 3)
- ☐ <1, 2, 3>
3. Que se passe-t-il si on fait : `x = 1` puis `x = 2` ? 3  1
- ☐ x vaut toujours 1
- ☐ une erreur
- ☐ x vaut 1 et 2 en même temps
- ☐ x vaut 2, mais la valeur 1 n'a pas changé
4. Que représente `:ok` en Elixir ? 4  1
- ☐ Un atome (constante nommée)
- ☐ Un booléen
- ☐ Une fonction spéciale
- ☐ Un raccourci pour `true`
5. Quelle est la valeur de `y` après : `[x, y] = [1, 2]` ? 5  1
- ☐ `a = 1` et `b = nil`
- ☐ Erreur de correspondance
- ☐ `(MatchError)`
- ☐ `a = [1]`, `b = []`
- ☐ `a = 1`, `b = 0`
6. Que se passe-t-il avec : `[a, b] = [1]` ? 6  1
7. Quelle est la sortie ? 7  1
- ```
defmodule M do
  def double(x), do: x * 2
end
IO.puts(M.double(5))
```
- ☐ Erreur
- ☐ 5
- ☐ 10
- ☐ 25
8. Quelle est la sortie ? 8  1
- ```
Enum.map([1, 2, 3], fn x -> x + 1 end)
```
- ☐ [1, 2, 3]
- ☐ [2, 2, 2]
- ☐ 4
- ☐ [2, 3, 4]

9. Quelle est la sortie ?

```
[1, 2, 3]
|> Enum.sum()
```

- ☐ [1, 2, 3]
- ☐ Erreur
- ☐ 6
- ☐ 123

10. Quelle expression retourne true ?

- ☐ if 0 do 0 end
- ☐ if true do true end
- ☐ if nil do true end
- ☐ if false do true end

11. Quelle est la sortie ?

```
defmodule M do
  def fact(0), do: 1
  def fact(n), do: n * fact(n - 1)
end
IO.puts(M.fact(3))
```

- ☐ 3
- ☐ 6
- ☐ 9
- ☐ Erreur

12. Quelle est la valeur de elem({:ok, 42}, 1) ?

- ☐ :ok
- ☐ {:ok, 42}
- ☐ 42
- ☐ Erreur

9

13. Que fait la fonction spawn en Elixir ?

- ☐ Crée un nouveau processus léger
- ☐ Exécute du code plus vite
- ☐ Crée un thread natif du système
- ☐ Crée une fonction anonyme

13

14. Que se passe-t-il avec :

```
send(self(), :hello)
```

puis:

```
receive do
  msg -> msg
end
```

?

- ☐ Erreur
- ☐ Bloque indéfiniment
- ☐ nil
- ☐ :hello

15. Quelle est la sortie ?

```
[1 | [2, 3]]
```

- ☐ [[1], 2, 3]
- ☐ [1, 2, 3]
- ☐ [1 | [2, 3]]
- ☐ Erreur

16. Quelle est la valeur de :

```
%{:a => 1, :b => 2}[:b]
```

- ☐ :b
- ☐ {:b, 2}
- ☐ 2
- ☐ nil

15

16

17. Dans un graphe représenté par une map: <sup>17</sup>

```
%{ a: [:b, :c] }
```

que représente `:b` ?

- ☐ Un poids
- ☐ Un voisin de `a`
- ☐ Un processus
- ☐ Une clé inconnue

18. Quelle est la sortie ? `Enum.filter([1, 2, 3, 4], fn x -> rem(x, 2) == 0 end)` <sup>18</sup>

- ☐ `[2, 4]`
- ☐ `[1, 3]`
- ☐ `[0, 2, 4]`
- ☐ Erreur

19. En Elixir, laquelle de ces comparaisons est vraie ? <sup>19</sup>

- ☐ `1 == :1`
- ☐ `nil > 0`
- ☐ `true != false`
- ☐ `:ok < :error`

20. Quelle est la sortie ? <sup>20</sup>

```
[1, 2, 3]  
|> Enum.map(fn x -> x * x end)  
|> Enum.sum()
```

- ☐ 6
- ☐ 9
- ☐ 12
- ☐ 14