

## UE17 : Les structures de contrôle en JavaScript

### Objectifs pédagogiques/ Exemples d'actions :

- Traduire des algorithmes séquentiels, conditionnels et itératifs en Javascript

### Contrôle de prerequisites :

1. Définir **structure de contrôle** : C'est un mécanisme qui permet de déterminer l'ordre d'exécution des instructions dans un algorithme ou un programme.

### Situation problème :

Un club de sport souhaite automatiser le calcul de la moyenne d'âge de ses membres. Le programme doit :

1. Demander à l'utilisateur d'entrer le nombre de membres ;
2. Permettre à l'utilisateur de saisir l'âge de chaque membre ;
3. Calculer la moyenne des âges ;
4. Afficher la moyenne.

De plus : Si la moyenne d'âge est inférieure à 18 ans, afficher : « Groupe Jeune » ; Sinon, afficher : « Groupe Adulte ».

### Consignes

Déduire les structures de contrôles nécessaires pour résoudre ce problème.

### Trace écrite

#### 1. Traduction de la structure linéaire en JavaScript

##### Algorithme Produit

```
Var a,b,p : Reel ;
Debut
  a ← 4 ;
  b ← 5 ;
  p ← a*b ;
Ecrire(p) ;
```

Fin

##### En JavaScript

```
<script language="JavaScript">
  var a=4 ;
  var b=5 ;
  p=a*b ;
  alert(p) ;
</script>
```

#### 2. Traduction des structures alternatives en JavaScript

**Une structure alternative** est une structure qui permet d'exécuter une instruction ou un groupe d'instructions selon une condition donnée.

##### a. La structure alternative réduite

###### Sa syntaxe en algorithmique :

```
Si (condition) alors
  Instruction1 ;
  Instruction2 ;
  ...
  Instruction n ;
Fin si
```

Exemple : En algorithmique

```
Algorithme negatif
  Var x : Entier ;
Debut
  Ecrire("Entrer un nombre") ;
  Lire (x) ;
  Si(x<0) alors
    Ecrire("Nombre négatif") ;
  Finsi
```

Fin

###### Sa syntaxe en JavaScript :

```
if (condition) {
  Instruction1 ;
  Instruction2 ;
  ...
  Instruction n ;
}
```

En javascript

```
<script language="JavaScript">
  x=parseInt(prompt(" Entrer un nombre"));
  if(x<0){
    alert("Nombre négatif") ;
  }
</script>
```

## b. La structure alternative complète

### Sa syntaxe en algorithmique :

```
Si (condition) alors
  Liste d'instructions 1 ;
Sinon
  Liste d'instructions 2 ;
Finsi
```

Exemple : En algorithmique

```
Algorithme negatif
  Var x : Entier ;
Debut
  Ecrire("Entrer un nombre") ;
  Lire (x) ;
  Si(x<0) alors
    Ecrire("Nombre négatif") ;
  Sinon
    Ecrire("Nombre positif") ;
  Finsi
Fin
```

### Sa syntaxe en JavaScript :

```
if (condition){
  Liste d'instructions 1 ;
}else{
  Liste d'instructions 2 ;
}
```

En javascript

```
<script language="JavaScript">
  x=parseInt(prompt(" Entrer un nombre"));
  if(x<0){
    alert("Nombre négatif") ;
  }else{
    alert("Nombre positif") ;
  }
</script>
```

#### 3. Traduction des structures itératives (Repete/ TantQue / Pour) en JavaScript

**Une structure itérative ou boucle** est une instruction composée permettant d'exécuter une suite d'instructions de façon répétitive en fonction d'une condition.

**Nombre d'itération** est le nombre d'exécution d'une boucle.

Il existe trois types de boucle à savoir : **la boucle pour ...finpour**, **la boucle tantque ...fintantque** et **la**

**boucle repeter...jusqu'à**

##### a. La boucle pour

La boucle **Pour...Faire** est une structure de contrôle qui répète un bloc d'instructions pour un nombre défini d'itérations.

###### Syntaxe en Algorithmique :

```
Pour variable allant de début à fin Faire
  Instructions
Fin Pour
```

###### Syntaxe en JavaScript :

Exemple :

En Algorithmique

```
s ← 0 ;
Pour i allant de 1 à 5 faire
  s ← s+1 ;
Finpour
```

```
for( var variable=début ; variable<=fin ;
variable++){
  Instructions
}
```

En javascript

```
<script language="JavaScript">
  s=0 ;
for( var i=1 ;i<=5 ;i++){
  s=s+i ;
}
</script>
```

##### b. La boucle tantque

La boucle **TantQue...Faire** est une structure de contrôle qui répète une instruction ou un bloc d'instructions tant qu'une condition est vraie.

###### Syntaxe en Algorithmique :

```
TantQue condition Faire
```

Instructions ;

**FinTantQue**

**Syntaxe en JavaScript :**

```
While( condition) {  
  Instructions ;  
}
```

**Exemple :**

**En Algorithmique**

```
i ← 1 ; s ← 0 ;  
Tantque i ≤ 5 faire  
  s ← s + i ;  
  i ← i + 1 ;  
Fintantque
```

**En javascript**

```
<script language="JavaScript">  
  i=1; s=0 ;  
  while(i<=5){  
    s=s+i ;  
    i=i+1 ;  
  }  
</script>
```

### c. La boucle Repeter

La boucle **Répéter...Jusqu'à** est une structure de contrôle qui exécute des instructions au moins une fois, jusqu'à ce qu'une condition soit vraie.

**Syntaxe en Algorithmique :**

```
Répéter  
  Instructions  
Jusqu'à condition ;
```

**Exemple :**

**En Algorithmique**

```
i ← 1 ; s ← 0 ;  
reperer  
  s ← s + i ;  
  i ← i + 1 ;  
jusqu'à ( i=5) ;
```

**Syntaxe en JavaScript :**

```
do  
  Instructions  
While( condition );
```

**En javascript**

```
<script language="JavaScript">  
  i=1; s=0 ;  
  do{  
    s=s+i ;  
    i=i+1 ;  
  }while(i !=5) ;  
</script>
```

**Exercice d'application**

Ecrire le programme de la situation problème.