

INTRODUCTION À L'ALGORITHMIQUE

BOUCLES

2014-07

Glenn ROLLAND / @glenux

CONCEPT

- On a besoin d'une condition d'arrêt / de continuation
- Ce qui est entre accolades est la zone ou le code est répété
- Le code répété *devrait* modifier la condition d'arrêt d'une façon ou d'une autre (sinon on a une boucle infinie...)

FAIRE... TANT QUE ...

- Exécute d'abord une fois le code répété
- Test en fin de boucle
- Recommence seulement si la condition est vraie

```
do {  
    // le code répété  
} while ( condition );
```

EXERCICES

- **compte-jusqua10** : Afficher les nombres de 1 à 10.
- **boucles-sum-1to10** : Faire la somme des nombres de 1 à 10 avec l'instruction `do ... while ...`.

TANT QUE ... FAIRE ...

- La condition est testée AVANT d'entrer dans la boucle

```
while ( condition ) {  
    // le code répété  
}
```

EXERCICES

boucles-nbval-somme :

- Demander un nombre $nbVal$ à l'utilisateur.
- En fonction de ce nombre, demander $nbVal$ nouveaux nombres à l'utilisateurs et faire leur somme.

boucle-nan-somme :

- Demander des nombres à l'utilisateur tant qu'il n'entre pas autre chose que des chiffres.
- Afficher la somme de ces chiffres à la fin.

EXERCICES

boucle-jeu-juste-prix

- un chiffre au hasard, l'utilisateur a 10 coups pour le trouver
- FIXME

POUR ... FAIRE ...

```
for (init; condition; transition) {  
    // code répété ici ;-)  
}
```


EXERCICES

- faire la somme des nbVal nombres demandés à l'utilisateur - OK
- faire la somme des chiffres de 1 à 10 (exo de la veille) - OK

COMPARAISONS

AVEC « FOR »

Un compteur qui commence à 1, reste inférieur à 10, avance de 1 en 1 et affiche i à chaque étape.

```
var i;  
for(i = 1; i <= 10; i += 1) {  
    alert(i);  
}
```

AVEC « WHILE ... »

```
var i = 1;  
while(i<=10) {  
    alert(i);  
    i+=1;  
}
```

AVEC « DO ... WHILE ... »

```
var i = 0;  
do {  
    i+=1;  
    alert(i);  
} while (i<10);
```

COMMENT BIEN CHOISIR SA BOUCLE ?

- **for** : si le nombre d'itération est connu d'avance
- **do .. while** : si le traitement répété au moins une fois
- **while** : tout le reste
 - exemple d'un MAUVAIS choix de boucle (répéter + calcul , plutot que tant que)

FIXME: exercices

BREAK : POUR SORTIR D'UNE BOUCLE

Il faut être capable de sortir d'une boucle !

```
var i = 0;
while(true) {
  if (i > 10) { break; }
  i += 1;
}
```

BREAK : EXERCICES

break-on-zero:

- demander à l'utilisateur des nombres et afficher leur carré
- s'arrêter lorsque l'utilisateur saisit zéro

break-sumlimit

- demander à l'utilisateur des nombres
- faire leur somme
- s'arrêter dès que la somme dépasse 500

BOUCLES DANS LES BOUCLES

loop-loop-pascal-triangle

- pour chaque chiffre i de l'intervale $[1, 10]$
- calculer res , la somme des chiffres de l'intervale $[1, i]$
- afficher chaque res pour chaque valeur de i

PROGRAMMATION STRUCTURÉE ET GOTO

FIXME