

## Méthodes d'un objet

### Rappels

Pour l'instant, l'objet *fraction* possède trois propriétés, son numérateur, son dénominateur et sa valeur approchée (calculée avec le maximum de précision permise par *JavaScript*). Pour la suite, on aura besoin d'un algorithme de calcul de pgcd, et c'est l'algorithme d'Euclide avec division qui a été choisi :

```
function pgcd(a,b){
  while(b>0){
    var c=a%b;
    a=b;
    b=c;
  }
  return a;
}

fraction=function(n,d){
  this.numérateur=n;
  this.dénominateur=d;
  this.valeur=n/d;
}
```

Cette fois-ci, on va fournir à l'objet *fraction*, en plus de ses propriétés (qui sont typiquement des nombres), des *méthodes*, qui seront des algorithmes.

### Exercice

Rajouter une nouvelle propriété à l'objet *fraction* : Son signe, égal à "+" si elle est positive, et à "-" si elle est négative.

```
fraction=function(n,d){
  this.numer=n;
  this.denom=d;
  this.valeur=n/d;
  if( ){
    this.signe= ;
  } else {
    this.signe= ;
  }
}
```

On pourra tester cet algorithme avec `var f=new fraction(2,-3); puis Println(f.signe);`.

### Affichage

Un premier exemple de méthode, importante pour l'objet *fraction*, est son affichage. Voici un algorithme possible :

```

fraction=function(n,d){
  this.numer=n;
  this.denom=d;
  this.valeur=n/d;
  this.affichage=function(){
    Println(n+"/"+d);
  }
}

```

Avec cette méthode, `f=new fraction(5,8);` suivi de `f.affichage();` donne l'affichage de *f* sous la forme 5/8 (ce qui est améliorable, le trait de fraction n'étant pas horizontal).

### **Simplification**

La simplification d'une fraction est aussi une méthode de celle-ci. Rédiger sous forme de méthode, un algorithme de simplification :

```

fraction=function(n,d){
  this.numer=n;
  this.denom=d;
  this.valeur=n/d;
  this.affichage=function(){
    Println(n+"/"+d);
  }
  this.simplifie=function(){

    return new fraction( , );
  }
}

```

Après avoir entré `var f=new fraction(48,32);` puis `f.simplifie().affichage();`, quelle fraction sera affichée? .....

### **Autres méthodes**

- 1°) Rédiger un algorithme qui implémente une méthode *inverse* pour l'objet *fraction*. Par exemple, avec `var f=new fraction(7,3);`, on veut que `f.inverse().affichage();` donne 3/7.
- 2°) Rédiger un algorithme qui implémente une méthode *opposé* pour l'objet *fraction*. Par exemple, on veut que `f.oppose().affichage();` donne 3/8 pour chacune des fractions suivantes :
  - a) `var f=new fraction(-3,8);`
  - b) `var f=new fraction(3,-8);`
 et que `f.oppose().affichage();` donne -3/8 pour chacune des fractions suivantes :
  - a) `var f=new fraction(3,8);`
  - b) `var f=new fraction(-3,-8);`
- 3°) Proposer d'autres méthodes pour l'objet *fraction*.