

# **JAVA BASICS**

## **Générer un nombre aléatoire entier**

```
int randomNumber = (int)(Math.random() * (max-min+1)) + min;
```

## **Générer un nombre aléatoire réel**

```
double randomNumber = Math.random() * (max - min) + min;
```

## **Comparer deux chaînes de caractères**

```
String test = "comparedString";  
    if (test.equals(comparedString)) {  
        return true;  
    }else{  
        return false;  
    }  
}
```

## **Buttons**

```
//Désactiver un bouton :  
    myButton.setEnabled(false);  
//Activer un bouton :  
    myButton.setEnabled(true);  
//Rendre un bouton invisible :  
    myButton.setVisible(false);  
//Rendre un bouton visible :
```

```
myButton.setVisible(true);  
//Modifier le texte afficher sur un bouton :  
myButton.setText("Do something");
```

## Label

```
//Afficher une chaîne de caractères dans un libellé :  
myLabel1.setText("Your name is " + name + ".");  
//Afficher un nombre (entier ou réel) dans un libellé :  
myLabel2.setText(String.valueOf(number));
```

## Text Field

```
//Placer le contenu d'un champ de texte dans une variable :  
String var1 = myTextField1.getText();  
//Lorsque le champ de texte contient un nombre entier :  
int var2 = Integer.valueOf(myTextField2.getText());  
//Lorsque le champ de texte contient un nombre réel :  
double var3 = Double.valueOf(myTextField3.getText());
```

## Vérifier qu'aucun des champs de texte n'est pas vide

```
if (!myTextField1.getText().equals("") && !myTextField2.getText()  
    !myTextField3.getText().equals("")) {  
    String var1 = myTextField1.getText();  
    int var2 = Integer.valueOf(myTextField2.getText());  
    double var3 = Double.valueOf(myTextField3.getText());  
}
```

## Progress bar

```
// number de 0 à 100;  
myProgressBar.setValue(Integer.valueOf(number));
```

## Request focus

```
textField.requestFocus();
```

## Trier une liste de nombres en ordre croissant

```
public void sort(){  
  
    //on place à la position i l'élément le plus petit parmi les élém  
    //à partir de on place à la position i l'élément le plus petit po  
    //éléments à partir de la position la position i  
    for(int i=0; i < alNumbers.size()-1; i++){  
  
        //la variable posMin contiendra la position de l'élément le plus  
        int posMin = i;  
        for(int j=i+1; j < alNumbers.size(); j++){  
            if(alNumbers.get(j) < alNumbers.get(posMin)){  
                }    }    posMin = j;  
  
        //on échange les éléments aux position i et posMin si l'élément  
        //plus petit n'est pas déjà à la position i  
        if(i != posMin){  
            Integer temp = alNumbers.get(posMin);  
            alNumbers.set(posMin, alNumbers.get(i));  
            alNumbers.set(i, temp);  
        }  
    }  
}
```

```
}
```

## Trier une liste de nombres en ordre décroissant

```
public void sort(){  
  
    //on place à la position i l'élément le plus petit parmi les élém  
    //à partir de on place à la position i l'élément le plus petit po  
    //éléments à partir de la position la position i  
    for(int i=0; i < alNumbers.size()-1; i++){  
  
        //la variable posMin contiendra la position de l'élément le plus  
        int posMin = i;  
        for(int j=i+1; j < alNumbers.size(); j++){  
            if(alNames.get(j).getNumber().compareTo(alNumbers  
        }    }    posMin = j;  
  
        //on échange les éléments aux position i et posMin si l'élément  
        //plus petit n'est pas déjà à la position i  
        if(i != posMin){  
            Integer temp = alNumbers.get(posMin);  
            alNumbers.set(posMin, alNumbers.get(i));  
            alNumbers.set(i, temp);  
        }  
    }  
}
```

## Mouse Events

```
//Retournent les coordonnées x ou y du point auquel se trouvait  
//le curseur de la souris lorsque l'événement a été déclenché.  
mouseButton = evt.getX();  
mouseButton = evt.getY();  
//
```

```
Retourne le point où se trouvait le curseur de
//la souris lorsque l'événement a été déclenché.
mouseButton = evt.getButton();
//Cette méthode ne fonctionne pas pour l'événement mouseDragged!
mouseButton = evt.getPoint();
```

## Import Timer

```
import javax.swing.Timer;
```

## Initialiser le Timer

```
// 1 = période de la répétition en millisecondes
timer = new Timer(1, myButton.getActionListeners()[0]);
```

## Timer

```
//démarrer le timer
timer.start();
//arrêter le timer
timer.stop();
//définir un délai pour le timer
timer.setDelay(delaySlider.getValue());
```