

# COURS APPLET JAVA



# QU'EST CE QU'UNE APPLET



- Une **Applet** est une application java exécutée dans un navigateur Web (Netscape, Internet Explorer,...)
- Une Applet ne possède pas de point d'entrée `main()` car celui-ci se trouve déjà dans le navigateur
  - Le navigateur contient en interne:
  - La machine virtuelle
  - Sa propre méthode

**`public static void main(String[] args)`**

- Le navigateur doit comprendre java et en plus effectuer des contrôles sur le code qu'il va exécuter (C'est pourquoi on ne laisse pas le code de l'Applet contrôler la machine virtuelle)
- Le navigateur fait donc tourner une application indépendante qui se charge d'exécuter les Applets après avoir effectué les contrôles nécessaires.

# QU'EST CE QU'UNE APPLET (SUITE)

## ■ Mais alors où commence l'exécution de mon Applet ?

Il a été défini un ensemble de méthodes qui sont utilisées comme différents points d'entrées en fonction du cycle de vie de l'Applet :

- `public void init( )`
- `public void start( )`
- `public void stop( )`
- `public void destroy ( )`
- `public void paint(Graphics g)...`

## ■ Pourquoi plusieurs points d'entrée ?

Une Applet est une application qui s'exécute dans un browser son utilisation dépend du bon vouloir de l'utilisateur du browser.

# CREER UNE APPLET



Pour créer une Applet, il suffit d'hériter de la classe **java.applet.Applet** ou **javax.swing.JApplet** (hérite de Applet) et surcharger les méthodes **init ( )**, **start ( )**, etc...

## Exemple :

```
import java.applet.*;
public class MonAppletextendsApplet
{
// surcharge des méthodes de notre choix
public void init ( )
{
.....
}
public void start ( )
{
.....
}
public void stop ( )
{
.....
}
}
```

# LA CLASSE APPLET



- Par héritage la classe MonApplet est devenue une Applet
- Ma classe appartient donc à toutes les classes de l'arbre d'héritage de la classe Applet:

`java.lang.Object`

`java.awt.Component`

`java.awt.Container`

`java.awt.Panel`

`java.applet.Applet`

`MonApplet`

- **Object**: fournit le savoir-vivre d'une entité Java
- **Component**: permet d'apparaître comme un composant graphique à l'écran
- **Container**: fournit la possibilité d'incorporer d'autres entités visuelles
- **Panel**: autorise l'alignement des entités visuelles incorporées
- **Applet**: complète ce qui précède pour fournir les points d'entrée appelés par le navigateur web
- **MonApplet**: ça dépend de ce qu'on va y mettre!

# LES MÉTHODES DE LA CLASSE APPLET APPELÉES PAR LE BROWSER



- **init ( )** : appelée à l'initialisation de l'Applet par le navigateur. Elle n'est exécutée qu'une seule fois. On y met toutes les opérations qui doivent être réalisées avant l'exécution de l'Applet (ex: chargement des graphismes, des sons, ...)
- **start ( )**: appelée par le navigateur chaque fois que l'Applet est interrompue et réactivée (ex: on quitte la page contenant l'Applet et en suite on revient dessus). Elle est appelée soit après init ( ), soit après un appel à stop ( )
- **stop ( )** : appelée à chaque fois que le navigateur arrête l'exécution de l'Applet (ex: changement de page, navigateur icônifié, ..). On peut penser ici à arrêter ici par exemple la musique, ...
- **destroy ( )**: est appelée lorsqu'on quitte le navigateur
- **paint( Graphics g)**: appelée chaque fois que l'on redessine l'Applet

## *Remarque :*

Si on définit un constructeur pour la classe qui hérite de Applet, celui ci est appelé en premier avant init ( )

# EXÉCUTER UNE APPLET

- Une Applet s'exécute au sein d'un navigateur, et est chargée et appelée à travers un fichier HTML
- Dans ce fichier HTML, un **tag APPLET** permet de spécifier la classe de base de cette Applet et les paramètres nécessaires à son lancement.
- La JVM se charge de la récupération des classes utiles à l'exécution de l'Applet. Les classes de l'API sont déjà présentes dans le navigateur
- **Exemple :**

...

**<applet**

code = «MonApplet.class»

name= «MonApplet»

width= 500

height= 500

codebase = «http://www.monserveur.org/java/»

>

**</applet>**

# LE TAG <APPLET>

## Syntaxe :

**<applet ... liste des attributs ...>**

**</applet>**

## ■ Quelques attributs :

**code :** nom de la classe principale

**name:** nom de l'applet

**width:** longueur de la zone d'affichage de l'Applet dans la page HTML

**height:** largeur de la zone d'affichage de l'Applet dans la page HTML

**codebase:** répertoire contenant le chemin de la classe principale (par défaut le répertoire courant)

## ■ Exemple :

**<applet**

code = «MonApplet.class»

name= «MonApplet»

width= 500

height= 500

codebase= «http://www.monserveur.org/java/»

**>**

**</applet>**



# LE TAG <APPLET>

## ■ Passage de paramètres à l'Applet

Dans la page HTML :

```
<applet ... liste des attributs ...>
```

```
<param name= nbre value = «4»>
```

```
</applet>
```

## ■ Récupération des paramètres

Dans le fichier java :

```
...
```

```
public void init ( )
```

```
{
```

```
int nombre;
```

```
String param1 = getParametre(«nbre»);
```

```
if (param1 == null) nombre = 0;
```

```
elsenombre = Integer.parseInt(param1)
```

```
}
```

```
...
```

# MÉTHODES UTILES DE LA CLASSE APPLET



## ■ **public URL getDocumentBase():**

retourne l'URL du document HTML contenant l'Applet  
(La classe **java.net.URL** a une méthode **getFile** qui retourne le nom du fichier de l'URL)

## ■ **public URL getCodeBase():**

Retourne l'URL du répertoire de chargement de l'Applet

### ***Remarque:***

Une URL (Uniform Resource Locator) est une adresse qui permet de localiser une ressource sur un réseau

Elle contient:

- Le protocole utilisé: http, ftp,...
- Le nom du serveur ou son adresse IP
- Le chemin et nom du fichier contenant les informations

Ex: <http://www.atos.com/fichier.html>

<http://www.ibp.fr/pub/outils/util.exe>

# L'INTERFACE APPLETCONTEXT



- Toute applet peut disposer d'un objet de type **AppletContext** qui représente le contexte dans lequel s'exécute cette applet.
- L'objet **AppletContext** permet en fait d'interagir avec le navigateur.
- Pour récupérer cet objet, il suffit d'utiliser la méthode de la classe Applet:

## **AppletContext getAppletContext()**

- Quelques méthodes de AppletContext:
  - **void showStatus(String mess):** permet d'afficher un message dans la barre de status du navigateur
  - **void showDocument(URL u):** permet d'afficher le document désigné dans la fenêtre courante du navigateur
  - **Image getImage(URL u) :** permet de charger une image dans un objet Image (supporte les formats JPEG et GIF)
  - **AudioClip getAudioClip(URL u):** permet de charger une séquence audio ...

# EXEMPLE 1



```
import java.awt.*;
import java.applet.*;
import java.net.* ;
public class Test extends Applet {
    public void init ( ) {
        AppletContext ctxt = getAppletContext();
        ctxt.showStatus("Cette applet: Affiche un
            document");
        try
        {
            ctxt.showDocument(new
                URL("http://www.elsnet.org/acl2001-
                arabic.html")); }
        catch (MalformedURLException)
        {
        }
    }
}
```

## EXEMPLE 2



```
import java.awt.*;
import java.applet.*;
public class ChargeImageextendsApplet
{
    Private Image image;
    public void init ( )
    {
        try
        {
            image = getImage(new URL(getCodeBase( ).toString()+
                «image.gif»));
        }
        catch (MalformedURLException) { }
    }
    public void paint(Graphics g)
    {
        g.drawImage(image, 10, 10, this);
    }
}
```

■ **public Image getImage( URL u ) :**

retourne une image résidant en mémoire et pouvant être affichée par Graphics.drawImage

# AVANTAGES/ INCONVÉNIENTS



## ■ Avantages

- rendre les pages web dynamiques
- enrichir les fonctionnalités d'un poste client
- Portabilité: les applets étant écrites en Java, elles sont par conséquent portables
- Possibilité de communication avec d'autres technologies basées sur le langage Java: les **servlets** et les **Java Server Pages**

## ■ Inconvénients

- Plusieurs navigateurs ne prennent plus en charge les plugins Java, l'utilisation d'applets Java sur le Web est dépréciée.
- L'élément HTML **<applet>**, pour les applets, définit l'intégration d'une applet Java. Cet élément est désormais déprécié.