

---

# Les Applets

---

ou  
“Java rencontre le Web”

Principes:

- partie dynamique d'un document Web
- exécution dans la machine locale
- chargement de code via le réseau
- code minimal nécessaire à la réalisation de la tâche
- protection “anti-virale” forte

---

# Documents WWW

---

## Un document = un noeud hypertexte

Document structuré => langage de marquage (HTML)

titre:                   <TITLE> C'est le titre </TITLE>  
en-tête niveau 1: <H1> La théorie du Tout </H1>  
liste:                    <UL>  
                          <LI> animé  
                          <LI> inanimé  
                          <LI> autre  
                          </UL>  
paragraphe: Il est nécessaire de tout  
                          comprendre avant de ...

## Liens entre documents = ancre + désignation

<A HREF=document-référencé> texte de l'ancre </A>

La recherche en <A HREF="/db-research/index.html">  
bases de données</A> &agrave; l'université  
cité;

Le document référencé peut être obtenu par “calcul”

tout sur la <A HREF=  
"/cgi-bin/get-index?x=botanique> botanique </A>

---

## Inclusion d'autres documents (IMG)

---

Dans le standard HTML seule l'inclusion d'images (en général en format GIF) est prévue:

`<IMG SRC=localisation de l'image>`

Si vous voyez le panneau `<IMG SRC="/images/stop.gif">` vous devez vous arrêter.

Ces images peuvent être calculées:

Vous êtes le visiteur no.

`<IMG SRC="/cgi-bin/compteur?cpt=c1">` sur ce serveur.

Impossible d'inclure un autre document HTML!

---

# Document dynamique sur serveur

---

Un document statique HTML apparaît tel qu'écrit par l'auteur.

Un serveur HTTP peut gérer des documents dynamiques :

Tout le contenu est calculé (généré) par le serveur, formaté en HTML, transmis au client.

Exemple: serveurs de recherche (Alta Vista, etc.)

`http://altavista.digital.com/cgi-bin/  
query?pg=q&what=web&fmt=.&q=argumentation+fluide`

Le client/navigateur WWW reste passif

Le serveur doit être très puissant (p.ex. Alta Vista = 4Gb RAM)

---

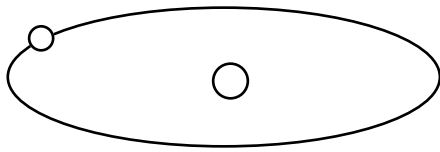
# Document dynamique avec Java

---

- Une partie du document est gérée par l'exécution d'un programme Java.
- Cette partie est délimitée par une fenêtre (au départ)
- La partie dynamique peut interagir avec le lecteur

Cours d'astronomie

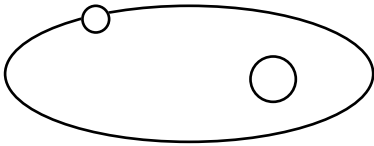
Texte théorique sur la dynamique des corps ...



<exécution d'une simulation>

... encore du texte ...

<simulation interactive>



start

stop

$V_0 =$         $g =$

---

## Le tag APPLET

---

But: inclure dans un document HTML un espace pour l'exécution d'une petite application.

```
<APPLET
  [CODEBASE = localisation_programme]
  CODE=nom_fichier_programme
  WIDTH=largeur_fenêtre
  HEIGHT=hauteur_fenêtre
  autres>

<PARAM NAME=nom1 VALUE=valeur1>
<PARAM NAME=nom2 VALUE=valeur2>
...
</APPLET>
```

Le tag PARAM permet d'envoyer des paramètres (strings) à l'applet.

---

## Cycle de vie d'une applet

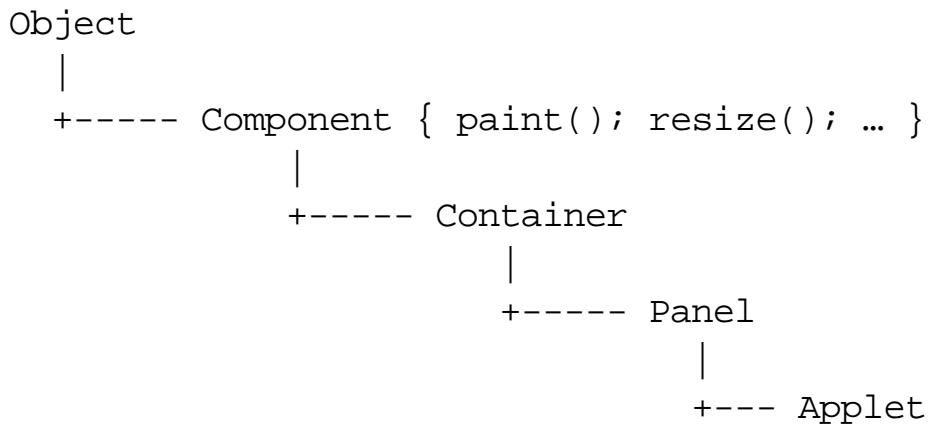
---

- Les applets sont sous le contrôle du navigateur WWW (Netscape, Hot Java, etc.)
- L'interface décide quand charger les applets d'une page HTML (=> état inactif)
- L'interface (re)démarre une applet quand sa fenêtre est visible sur l'écran (=> état actif)
- L'interface arrête l'applet quand elle disparaît de l'écran (=> état inactif)
- L'interface efface l'applet quand elle n'en a plus besoin

---

# La classe Applet

---



- Une applet est donc un *objet graphique*.
- Mais c'est aussi un *objet actif* créé et contrôlé par le navigateur Web

La classe Applet possède deux catégories de méthodes:

- les méthodes d'interface graphique (héritées):

```
public void paint(Graphics g);
public boolean mouseDown(Event evt, int x,
    int y);
public boolean action(Event evt,
    Object what);
etc.
etc.
```

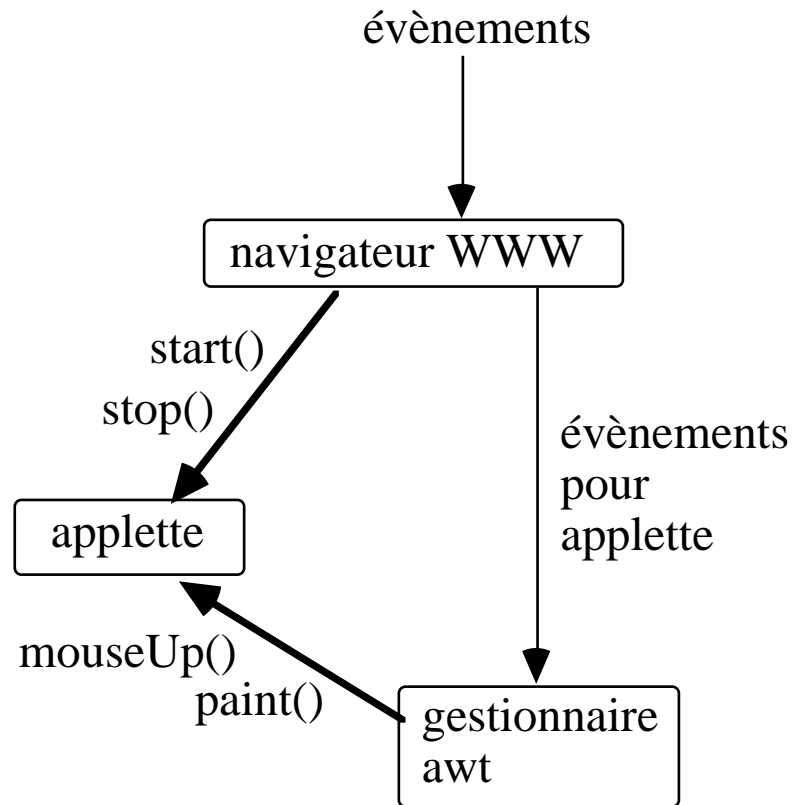
- les méthodes de contrôle d'exécution

```
public void init();
public void start();
public void stop();
public void destroy();
etc.
```

---

# Interaction d'une applet

---



---

# Écrire une applet

---

```
import java.applet.*;
import java.awt.*;

public class <NomApplet> extends Applet {

    public void init() {
        <Initialisations>
        <Démarrage de processus>
    }
    public void start() {
        <Démarrer l'applet, la page Web est visitée>
    }
    public void paint(Graphics g) {
        <Dessiner le contenu actuel de l'applet>
    }
    public void stop() {
        <Arrêter l'applet, la page Web est quittée>
    }
    public void destroy() {
        <Relâcher les ressources, l'applet va quitter
        la mémoire>
    }
}
```

## Cycle de vie d'une applet

```
init() ( start() paint()* stop() )* destroy()
```

---

# Applet : exemples

---

Ecrire une applet == créer une sous-classe de Applet  
pour  
redéfinir certaines méthodes

## **Applet minimale:** ne redéfinit que paint()

```
import java.applet.*;
import java.awt.*;

public class MonApplet extends Applet {
    // redéfinit la méthode paint()
    public void paint(Graphics g) {
        int px[] = {5, 10, 0, 20, 40, 20};
        int py[] = {5, 20, 40, 60, 40, 5};
        g.drawPolygon(px, py, 6);
        g.drawString("Une applet !", 50, 50);
    }
}
```

---

# Paramètres

---

L'applet peut lire les paramètres avec la méthode

```
public String getParameter(String name)
    (null si le paramètre n'existe pas)
```

Comme les paramètres sont des Strings il faut les convertir selon ce qu'ils représentent (int, float, etc.)

```
public void init() {

    String s;

    s = getParameter("level");
    if (s != null) fractLevel = Integer.parseInt(s);

    s = getParameter("incremental");
    if (s != null)
        incrementalUpdates = s.equals("true");

    s = getParameter("delay");
    if (s != null)
        repaintDelay = Integer.parseInt(s);

    s = getParameter("startAngle");
    if (s != null)
        startAngle = Float.valueOf(s).floatValue();

    s = getParameter("rotAngle");
    if (s != null)
        rotAngle = Float.valueOf(s).floatValue();
```

---

## Extension de HTML par les applets

---

HTML permet essentiellement de structurer du texte

Rien n'est prévu pour les graphiques, formules (math., chimiques, etc.), circuits électroniques, ...

Une applet peut prendre en charge l'affichage d'une catégorie de données et son animation

p.ex.

Dans le graphe ci-dessous

```
<APPLET CODE="Graphe.class" ...>  
  <PARAM NAME=noeuds VALUE="s1 s2 s5 s12 s44">  
  <PARAM NAME=arcs  
    VALUE="s1-s12 s44-s1 s44-s2 s2-s12">  
</APPLET>
```

etc. etc. etc.

La molécule

```
<APPLET CODE="FChimique.class" ...>  
  <PARAM NAME="formule" VALUE="C6H12O6">  
</APPLET>
```

présente une forme qui ...

---

# **Interactions applet - environnement**

---