

HTML

Aspect 2

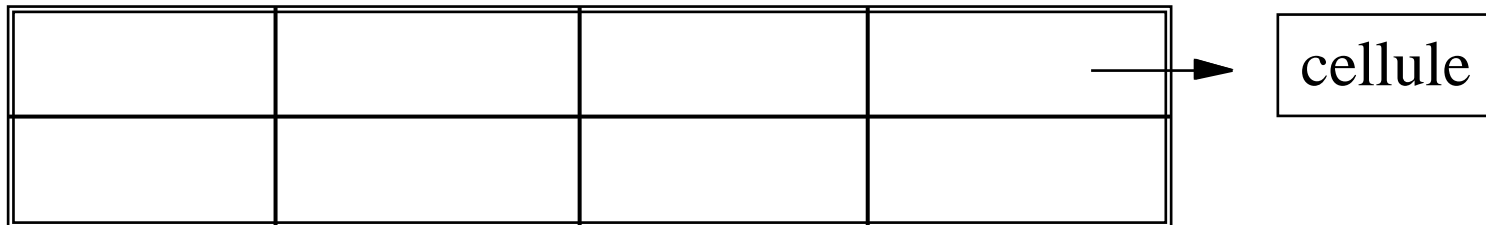
INTRODUCTION

- Nous avons vu HTML aspect 1:
 - structure d'un document html
 - les liens simples et complexes
 - les images
 - l'aspect multimédia
- Nous allons voir aspect 2:
 - les tableaux
 - les scripts
 - les formulaires

LES TABLEAUX

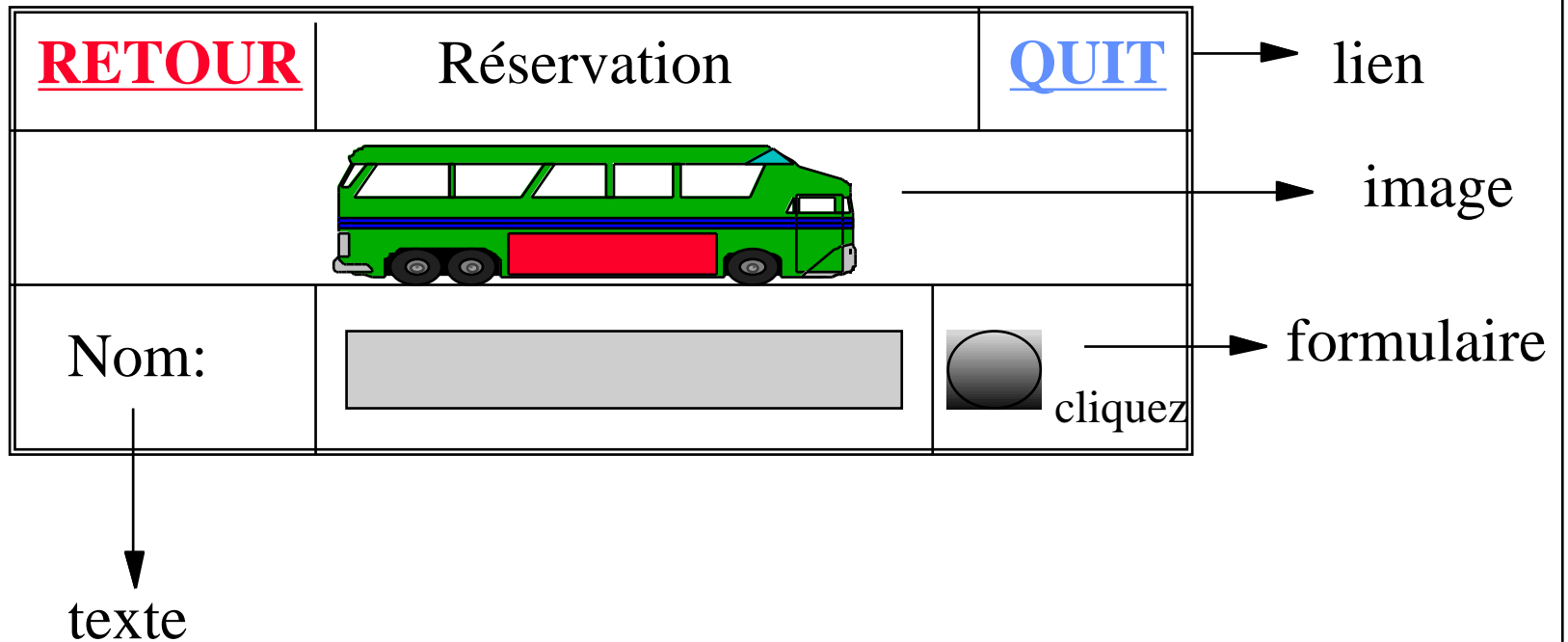
- version HTML 3.0
 - présentation des informations dans un tableau
 - cadre
 - cellules: taille et espacement
 - description ligne/ligne comme les listes html
 - Balise

<TABLE></TABLE>



- Élément d'un tableau

texte liste image liens élément_formulaire



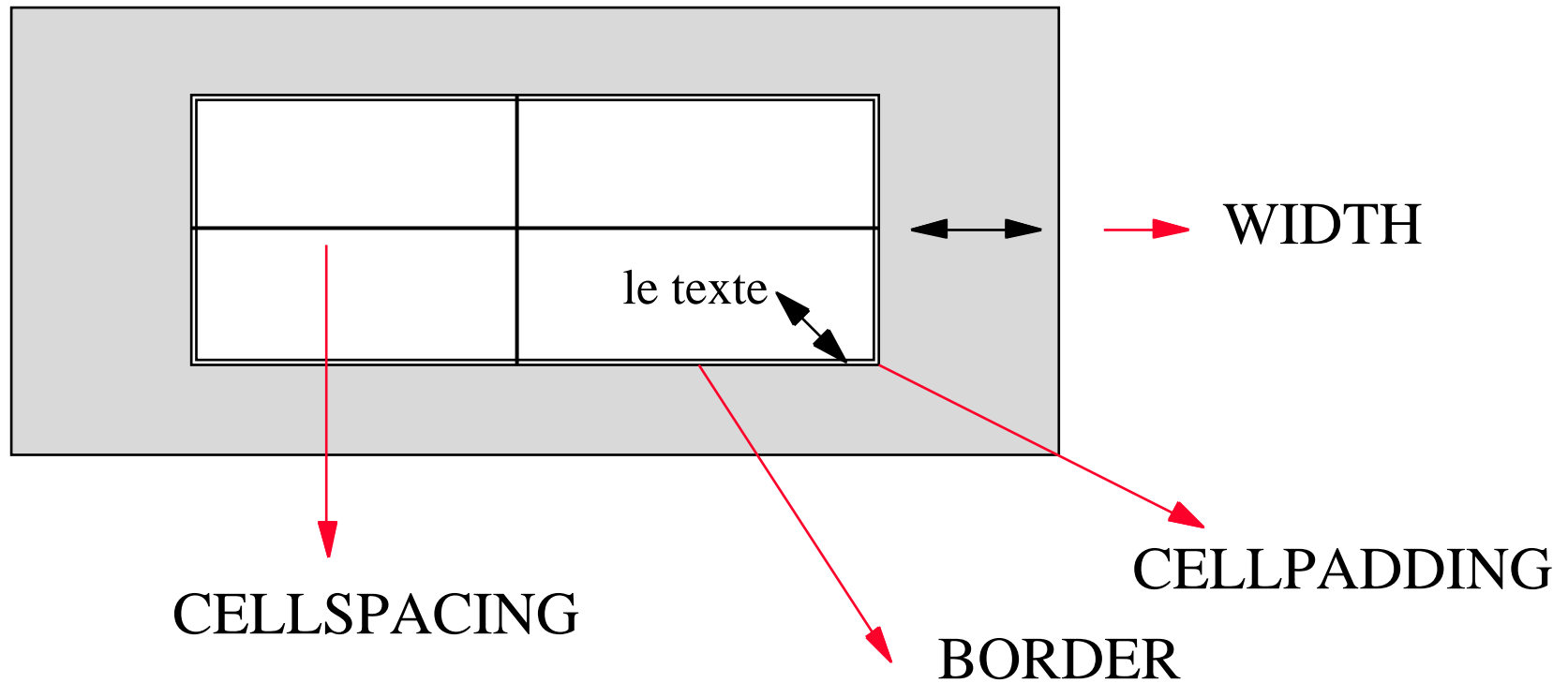
- La présentation
 - propre à chaque navigateur norme HTML 3

- Les balises

<TABLE>.....</TABLE>

- BORDER épaisseur cadre extérieur
- CELLSPACING espace inter-cellule
- CELLPADDING espace autour texte cellule
- WIDTH largeur fenêtre

```
<TABLE BORDER=3 CELLSPACING=2 CELLPADDING=10>  
.....  
.....  
</TABLE>
```



- **CAPTION**

- titre accompagnant le tableau

<CAPTION>.....</CAPTION>

- alignement avec attribut **ALIGN=**

Titre en bas de tableau

```
<CAPTION ALIGN="BOTTOM">  
<B>Titre en bas de tableau</B>  
</CAPTION>
```

- Ligne de tableau

- description d'un tableau ligne/ligne
- description des éléments de chaque ligne
- balises

<TR>.....</TR> **ligne**

<TD>.....</TD> **élément**

```
<TR ALIGN="CENTER">  
  <TD> <A HREF="http://cnam.fr"> CNAM</A> </TD>  
  <TD> <IMG SRC="bus.gif"> </TD>  
</TR>
```


- Attributs de TR ligne
 - ALIGN= alignement horizontal
 - VLIGN= alignement vertical

les valeurs:

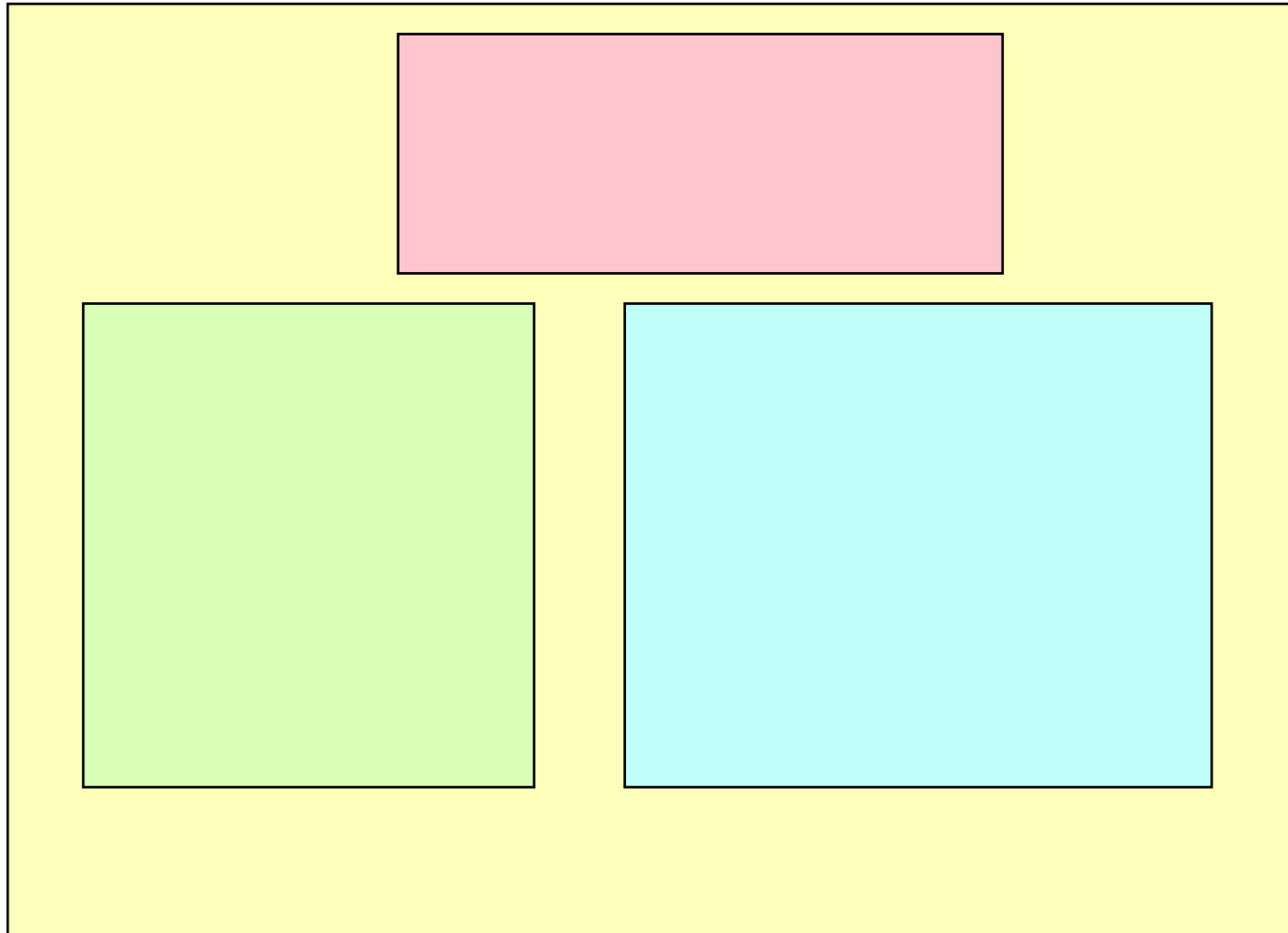
- TOP en haut
- BOTTOM en bas
- MIDDLE au milieu
- RIGHT à droite
- LEFT à gauche
- CENTER au centre

- Attributs de TD élément
 - VALIGN=
 - ALIGN=
 - COLSPAN= multiple de la colonne de base
 - ROWSPAN= multiple de la ligne
 - NOWRAP texte sur la même ligne
- Ajustement automatique taille des cellules
 - sur la longueur du texte le plus long
- Autre balise: <TH>
 - idem <TD> mais texte mis en gras pour le titre des cellules

LES FRAMES

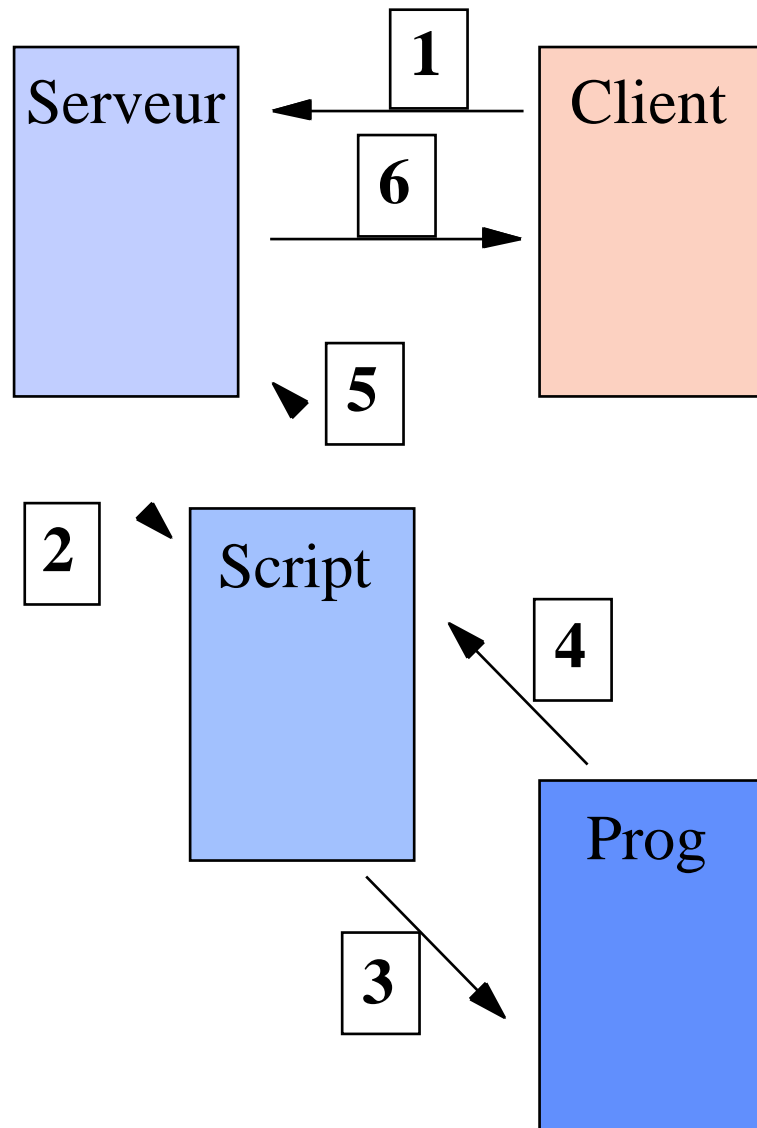
- Ne font partie de la norme HTML
- Principe:
 - facilité d'usage des tableaux
 - l'écran est décomposé en plusieurs zones
 - chaque zone est un document html
 - l'affichage de chaque frame est indépendant
 - différence tableau/frame
 - le tableau est un seul document affiché => retour arrière pour voir la page précédente

La fenêtre du navigateur est décomposée en plusieurs frames



LES SCRIPTS

- Interaction entre le client et le serveur Web
 - lien vers un programme à exécuter sur le serveur
 - le client Web envoie une requête
 - le serveur exécute et retourne les résultats à afficher
 - script cgi (Common Gateway Interface)
- Les langages script
 - Unix C, Fortran, Perl, Shell
 - Mac Applescript
 - PC Visual Basic, Visual C++



- Comment ça marche?

- 1- URL client pointe sur un document_script
- 2- le serveur reconnaît appel script
- 3- le script s'exécute et appel prog
- 4- le résultat de prog est renvoyé à script
- 5- le script prépare le résultat en html
- 6- le serveur envoie au client résultat en html

- Un exemple:

Dans document.html

**afficher la date **

En cliquant sur 'afficher la date': la date du jour s'affiche

cgi-bin répertoire des scripts cgi (défini à la config.)

getdate script cgi

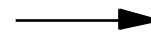
getdate

```
#!/bin/sh
```

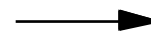
```
echo Content-type: text/plain
```

```
echo
```

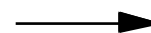
```
/bin/date
```



Commentaire #
Script écrit en shell



En-tête du document
séparateur: ligne



Corps du document

- Remarques:
 - les exécutions de plusieurs scripts ralentissent le serveur
 - les appels de paramètres ne sont pas contrôlés (situation de blocage possible)
 - les scripts doivent être dans un répertoire particulier (cgi-bin)
 - voir la solution de java

- L'en-tête d'un script
 - ne fait pas partie du document
 - type du contenu du document renvoyé:
 - Content-type:text/html en html
 - Content-type:text/plain en texte
 - Content-type:image/gif image gif
 - Content-type:image/jpeg image jpeg
 - Content-type:Application/PostScript en Postscript
 - Content-type:video/mpeg en mpeg
- Content-type de type MIME

- Les résultats du script

Exemple de script de type html

```
#!/bin/sh
echo Content-type:text/html
echo
cat<<EOF
<HTML><HEAD>
<TITLE>Date</TITLE></HEAD>
<BODY><P>La date du jour est:<B>
EOF
/bin/date
cat<<EOF
</B></P></BODY></HTML>
EOF
```

Le résultat renvoyé au client

```
Content-type:text/html

<HTML><HEAD>
<TITLE>Date</TITLE></HEAD>
<BODY><P>La date du jour est:<B>
Wen Nov 17 10:20:00 EDT 1996
</B></P></BODY></HTML>
```

- Les variables d'environnement
 - passage d'informations entre client/serveur

Dépendantes de la requête:

SERVER_PROTOCOL
SERVER_PORT
REQUEST_METHOD
SCRIPT_NAME
REMOTE_HOST
REMOTE_ADDR
AUTH_TYPE
REMOTE_USER
REMOTE_IDENT
CONTENT_TYPE
CONTENT_LENGTH

Hors requête:

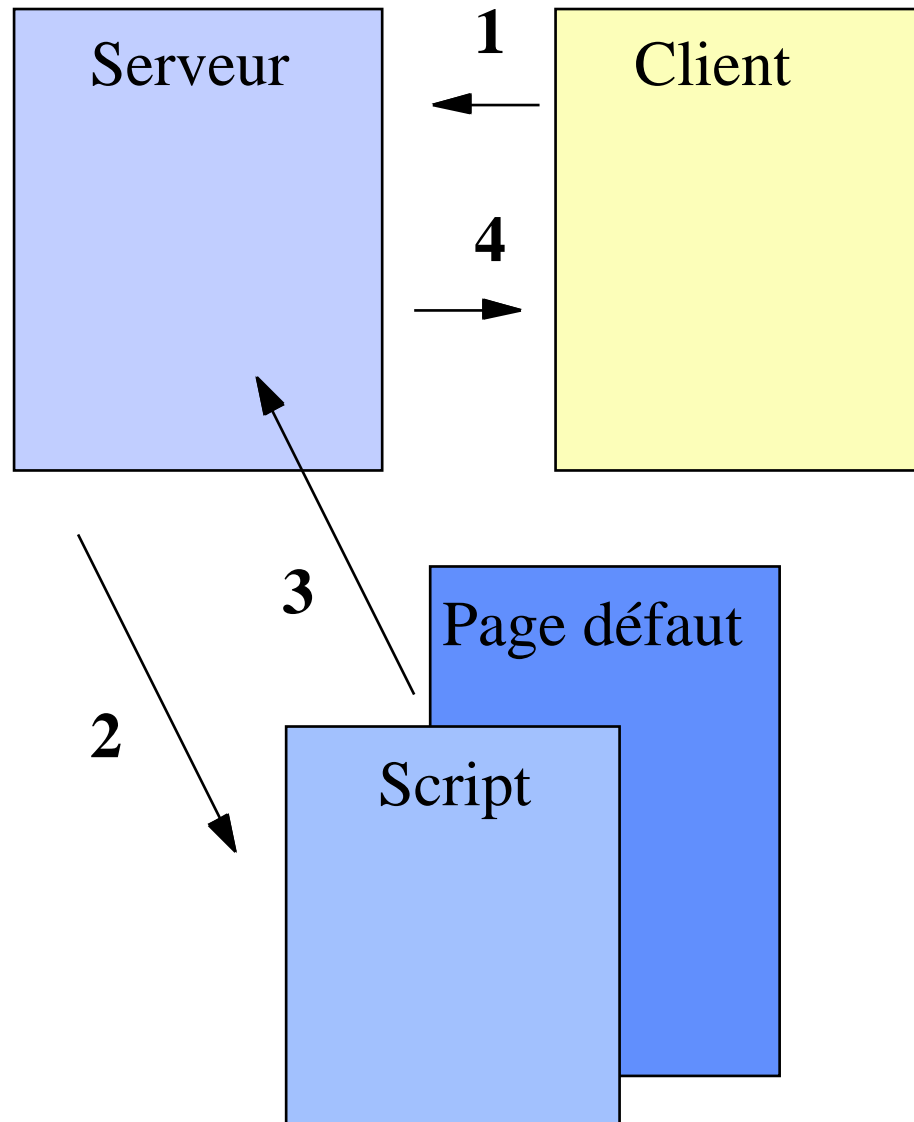
SERVER_SOFTWARE
SERVER_NAME
GATEWAY_INTERFACE

Passage de paramètres:

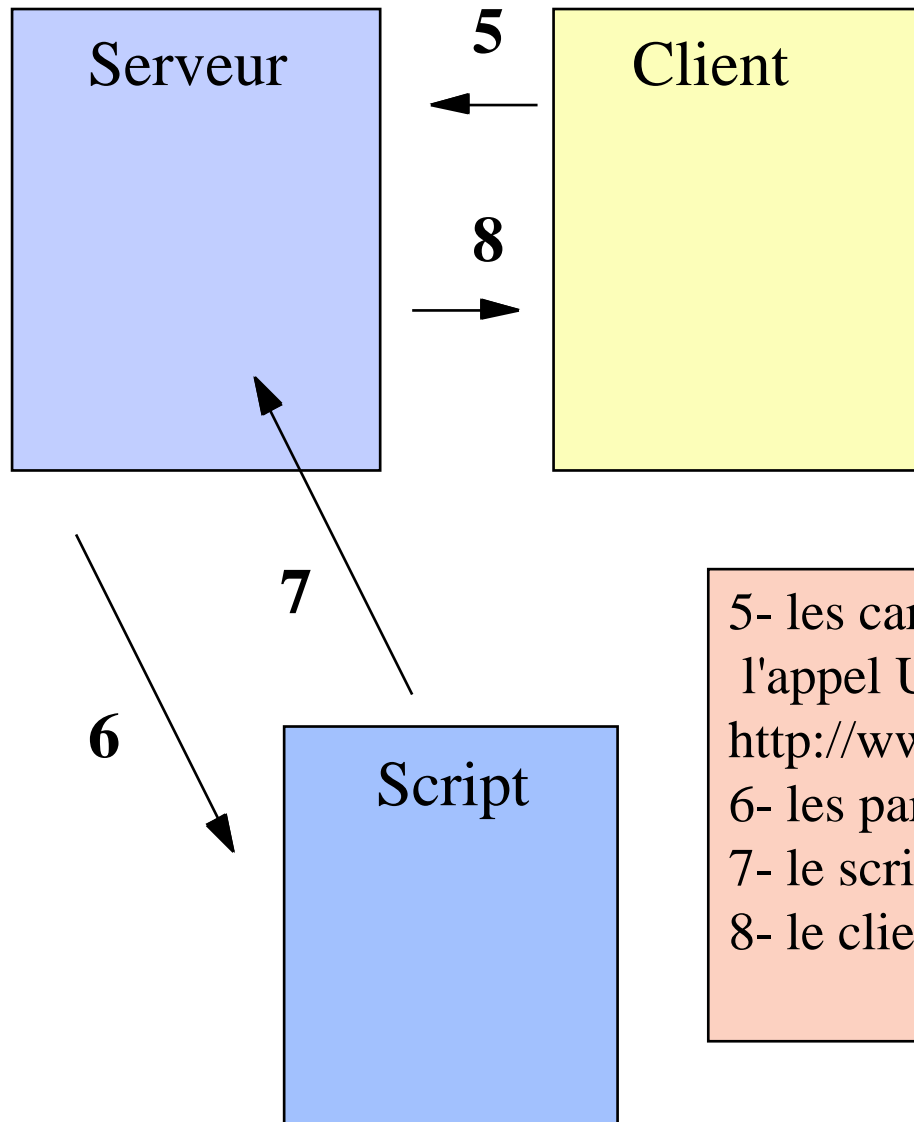
**PATH_INFO
QUERY_STRING**

- L'interactivité entre client et serveur
 - formulation requête dans la fenêtre du navigateur
 - le serveur recherche en fonction de la requête
 - la réponse est affichée
 - Côté serveur
 - un script d'invitation pour entrer la requête
 - un script de réponse à ce questionnaire
 - Questionnaire
 - soit un document html
 - soit un formulaire

- Question dans un document html
 - éléments particuliers:
 - balise <ISINDEX>
 - URL avec réponse
 - passage de paramètres pour le script
 - principe:
 - appel de script sans passage de paramètre
 - erreur: formulation question par envoi d'une page par défaut
 - réponse à la question: passage des paramètres
 - le script récupère les paramètres
 - réponse: page html



- 1- envoi de la requête script
- 2- le navigateur appelle le script sans paramètre
- 3- lorsqu'un script ne reçoit pas de paramètre, il renvoie une page par défaut
- 4- la balise `<ISINDEX>` placée dans la page par défaut provoque une question (demande URL)



5- les caractères entrés se placent à la suite de l'appel URL après ?
`http://www.cnam.fr/cgi-bin/pscript?paramètres`
6- les paramètres parviennent au script
7- le script renvoie la page html réponse
8- le client reçoit la page html et l'affiche

- ISINDEX

- dans la page par défaut provocation question

- <HEAD> <ISINDEX> <TITLE>le nom du chien </TITLE> </HEAD>**

- la réponse se fait par une frappe au clavier

- la frappe des caractères devient un argument

- nouvelle tentative d'accès à l'URL avec l'argument

- ?chaîne de caractères**

- **Un script par défaut**

```
#!/bin/sh
echo Content-type:text/html
echo

cat <<EOF
<HTML><HEAD><TITLE>
EOF
if($#=0);then
cat <<EOF passage d'argument
</TITLE><ISINDEX></HEAD>
<BODY><P>tapez votre argument
EOF
else if($1!="toto"):then
cat<<EOF erreur frappe
</TITLE><ISINDEX></HEAD>
<BODY><P> retapez
EOF
```

```
else
cat<<EOF frappe correcte
</TITLE></HEAD>
<BODY><P>c'est correct
EOF
fi
fi

cat<<EOF
</P></BODY></HTML>
EOF
```

- **Avec Location dans le script**
 - redirection vers un autre URL
 - pas de ligne Content-type
 - pas de corps

```
<A  
  HREF="http://www.cnam.fr/cgi-bin/verslion">  
  afficher la date </A>
```

Script: verslion

```
#!/bin/sh  
echo Location:http://lion.cnam.fr/  
echo
```

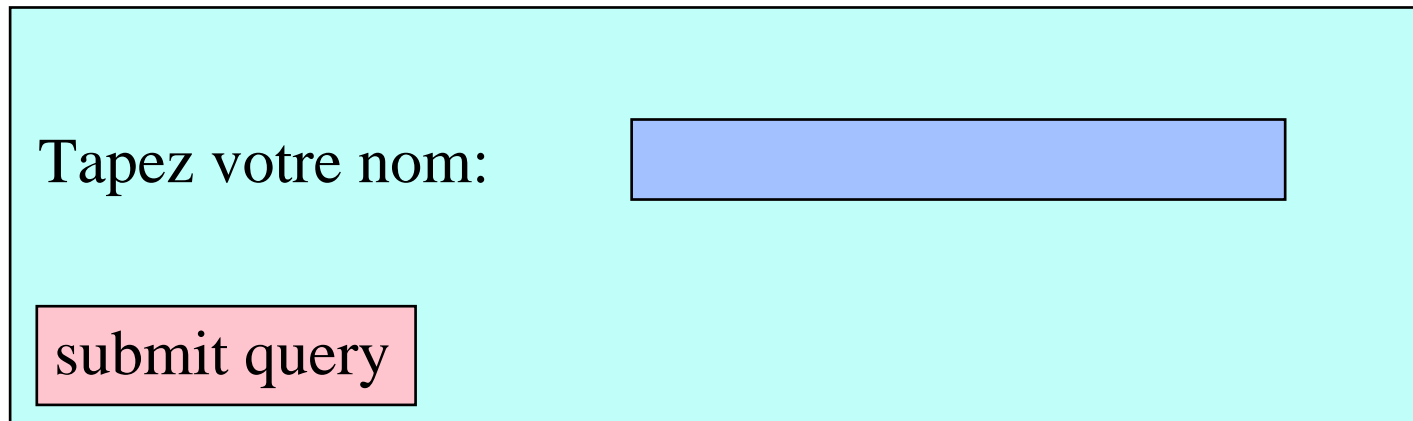
- Passage d'argument sans réponse
 - remplacer Content-type ou Location par **Status: 204 No Response**
 - le code 204 signifie 'ne rien faire'
 - le serveur prend l'argument et le stocke

LES FORMULAIRES

- Deux modes de questionnaires
 - questions dans un document (vu précédemment)
 - questions dans un formulaire
- Version HTML 3.0
- Côté navigateur: norme HTML 3
- Côté serveur: interface CGI
- Différence requête/formulaire
 - requête: arguments séparés par des espaces à la frappe
 - formulaire: une fenêtre par saisie d'argument

Un formulaire:

- une ou plusieurs zones
- un nom par zone
- une valeur par zone



Tapez votre nom:

submit query

The diagram shows a light blue rectangular area representing a form. On the left side of this area, the text "Tapez votre nom:" is displayed. To the right of this text is a blue rectangular input field. Below the input field, on the left side, is a pink rectangular button containing the text "submit query".

- Transfert: formulaire vers serveur
 - "nom=valeur"
 - envoi des données au script
 - GET requête type document
données placées dans l'URL (visible)
puis dans la variable QUERY_STRING
 - POST données sur l'entrée standard du script
sans passer par QUERY_STRING
la longueur des données est dans CONTENT-LENGTH
- Le script
 - si GET invite envoyée par le script lors de la requête
 - si POST le document html contient la requête

- Les requêtes
 - attributs Content-type ou Location ou Status
- Décodage CGI
 - dissociation de la valeur et de la zone
nom=valeur
 - utilitaire cgifparse (HTTPD du CERN)
 - variable au nom FORM_nom
- Balises

<FORM>.....</FORM>

- Attributs de FORM

```
<FORM METHOD="POST" ACTION="../cgi-bin/formule.date">  
</FORM>
```

METHOD	POST ou GET
ACTION	fichier script appelé

- Bouton 'SUBMIT'
 - envoi de donnée
 - un formulaire = 1 bouton SUBMIT
- TYPE="SUBMIT"

Tapez votre nom:

submit query

- Balise SUBMIT

<INPUT TYPE="SUBMIT" VALUE ="intitulé bouton">

- Appel du code du formulaire

test-cgi: formulaire qui renvoie les valeurs vers le serveur

- Action du bouton

– dans la partie formulaire

```
<INPUT TYPE="SUBMIT" VALUE="envoi">  
<FORM METHOD="POST" ACTION="../cgi-bin/test-cgi">  
<INPUT TYPE="SUBMIT">  
</FORM>
```

déclaration

action

- Le texte
 - champ de saisie de type texte

TYPE="TEXT"

- balise

<INPUT TYPE="text" NAME="champ">

– attributs

- TYPE type texte
- NAME nom du champ de saisie
- SIZE long maxi du champ
- MAXLENGTH limite du nb de car saisis

```
<FORM METHOD="POST" ACTION="../cgi-bin/test-cgi">  
login  
<INPUT TYPE="TEXT" NAME="champ_login">  
<BR>  
password  
<INPUT TYPE="TEXT" NAME="champ_pwd">  
<BR>  
</FORM>
```

login

password

- Le bouton radio
 - permet le choix parmi une liste

TYPE="RADIO"

- balise

<INPUT TYPE="RADIO" NAME="temp" VALUE="chaud">

– attributs

- TYPE
- NAME le même nom du champ de saisie pour la liste
- VALUE valeur attribuée si choix

```
<FORM METHOD="POST" ACTION=" ../cgi-bin/test-cgi">
<OL>
<LI><INPUT TYPE="RADIO" NAME="boisson" VALUE="chocolat">
chocolat<BR>
  <OL>
  <LI><INPUT TYPE="RADIO" NAME="temp." VALUE="chaud">chaud
  <LI><INPUT TYPE="RADIO" NAME="temp." VALUE="tiède">tiède
  <LI><INPUT TYPE="RADIO" NAME="temp." VALUE="froid">froid
  </OL>
<LI><INPUT TYPE="RADIO" NAME="boisson" VALUE="thé">
thé<BR>
  <OL>
  <LI><INPUT TYPE="RADIO" NAME="temp." VALUE="chaud">chaud
  <LI><INPUT TYPE="RADIO" NAME="temp." VALUE="froid">froid
  </OL>
</OL>
</FORM>
```

- Les cases à cocher
 - plusieurs cases à cocher si choix
- TYPE="CHECKBOX"

- balise

<INPUT TYPE="CHECKBOX" NAME="maths">

- attributs

- TYPE
- NAME nom du champ de saisie(un par choix)
- VALUE valeur attribuée si choix

```
<FORM METHOD="POST" ACTION="../cgi-bin/test-cgi">
<P>Choisissez vos uvs</P>
<UL>
<LI><INPUT TYPE="CHECKBOX" NAME="maths" VALUE="maths">
mathématiques<BR>
<LI><INPUT TYPE="CHECKBOX" NAME="physique" VALUE="phys">
physique
<LI><INPUT TYPE="CHECKBOX" NAME="chimie" VALUE="chim">
chimie
</UL>
</FORM>
```


- Le bouton Reset

- annulation des choix

- TYPE="RESET"

- balise

- <INPUT TYPE="RESET" VALUE="valeur default">**

- valeur par défaut

- chaque élément d'un formulaire a une valeur par défaut

- texte: valeur de VALUE
 - cases à cocher: valeur de CHECKED
 - bouton radio: valeur de CHECKED

Un exemple complet