

# PHP

Cours #01. Hello world.

**LE SAVAIS-TU?**



# INTERNET ?

INTERCONNECTED  
NETWORKS

# INTERNET ?

INTERCONNECTED  
NETWORKS

INTERNET ?

URL ?

INTERCONNECTED  
NETWORKS

# INTERNET ?

## URL ?

UNIFORM  
RESOURCE  
LOCATOR

<http://subdomain.domain.tld:port/folder/file.ext>

INTERCONNECTED  
NETWORKS

# INTERNET ?

## URL ?

UNIFORM  
RESOURCE  
LOCATOR

## TLD ?

<http://subdomain.domain.tld:port/folder/file.ext>



# INTERNET ?

## URL ?

UNIFORM  
RESOURCE  
LOCATOR

## TLD ?

TOP  
LEVEL  
DOMAIN

<http://subdomain.domain.tld:port/folder/file.ext>

# INTERNET ?

## URL ?

UNIFORM  
RESOURCE  
LOCATOR

## TLD ?

TOP  
LEVEL  
DOMAIN

<http://subdomain.domain.tld:port/folder/file.ext>

## PORT par défaut d'un serveur web ?

# INTERNET ?

## URL ?

UNIFORM  
RESOURCE  
LOCATOR

## TLD ?

TOP  
LEVEL  
DOMAIN

http://subdomain.domain.tld:port/folder/file.ext

## PORT par défaut d'un serveur web ?

http://subdomain.domain.tld:80/folder/file.ext

**URL = UNIFORM RESOURCE LOCATOR**  
(= «ADRESSE WEB»)

URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>

# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>
  - http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)

# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>
  - ✦ http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)
  - ✦ ensuite, de droite à gauche jusqu'au premier «slash» (/)  
(du général au particulier)

# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>
  - http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)
  - ensuite, de droite à gauche jusqu'au premier «slash» (/)  
(du général au particulier)
    - .be = Top Level Domain



# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>
  - ✦ http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)
  - ✦ ensuite, de droite à gauche jusqu'au premier «slash» (/)  
(du général au particulier)
    - ✦ .be = Top Level Domain
    - ✦ lesoir = domain name (nom de domaine)

# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

- Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>
  - ✦ http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)
  - ✦ ensuite, de droite à gauche jusqu'au premier «slash» (/)  
(du général au particulier)
    - ✦ .be = Top Level Domain
    - ✦ lesoir = domain name (nom de domaine)
    - ✦ www = sous-domaine

# URL = UNIFORM RESOURCE LOCATOR

(= «ADRESSE WEB»)

· Ex: <http://www.lesoir.be/exclusif/prince-philippe-est-une-fille.html>

- ✦ http:// = le protocole (idem: ftp:// , file://, mailto: )  
( http = hypertext transfer protocol, ftp = file transfer protocol)
- ✦ ensuite, de droite à gauche jusqu'au premier «slash» (/)  
(du général au particulier)
  - ✦ .be = Top Level Domain
  - ✦ lesoir = domain name (nom de domaine)
  - ✦ www = sous-domaine
- ✦ Cet URL correspond à dire à la machine:  
«va chercher le fichier prince-philippe-est-une-fille.html se trouvant dans le dossier « /exclusif/ » sur l'ordinateur **be.lesoir.www** via le protocole http»

# Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

# Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

• WWW.LESOIR.BE = ?



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

# Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

- lesoir.be ip = 81.246.2.9

• WWW.LESOIR.BE = ?



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

# Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

• lesoir.be ip =  
81.246.2.9

• WWW.LESOIR.BE = ?



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

## Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

• lesoir.be ip =  
81.246.2.9

• WWW.LESOIR.BE = ?



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

## Etape 2



## Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

- lesoir.be ip = 81.246.2.9

• WWW.LESOIR.BE = ?



client

serveur web

81.246.2.9



## Etape 2

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

## Etape 1

I.S.P

serveur DNS de l'ISP



195.13.15.25

- lesoir.be ip = 81.246.2.9

• WWW.LESOIR.BE = ?



client

serveur web



81.246.2.9

## Etape 2

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

Etape 1

I.S.P  
serveur DNS de l'ISP



195.13.15.25

• WWW.LESOIR.BE = ?

• lesoir.be ip =  
81.246.2.9

• SALUT, JE VOUDRAIS QUE TU ME SERVES LE  
FICHIER LESOIR.BE/INDEX.PHP ET SES  
FICHIERS ASSOCIÉS.

serveur web



81.246.2.9



client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

Etape 2

Etape 1

I.S.P  
serveur DNS de l'ISP



195.13.15.25

• WWW.LESOIR.BE = ?

• lesoir.be ip =  
81.246.2.9



client

• SALUT, JE VOUDRAIS QUE TU ME SERVES LE  
FICHIER LESOIR.BE/INDEX.PHP ET SES  
FICHIERS ASSOCIÉS.

serveur web



81.246.2.9

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

Etape 2

Etape 1

I.S.P  
serveur DNS de l'ISP



195.13.15.25

• WWW.LESOIR.BE = ?

• lesoir.be ip =  
81.246.2.9

• SALUT, JE VOUDRAIS QUE TU ME SERVES LE  
FICHIER LESOIR.BE/INDEX.PHP ET SES  
FICHIERS ASSOCIÉS.

serveur web



81.246.2.9

• LES VOICI!



Etape 2

client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

Etape 1

I.S.P  
serveur DNS de l'ISP



195.13.15.25

• WWW.LESOIR.BE = ?

• lesoir.be ip =  
81.246.2.9

• SALUT, JE VOUDRAIS QUE TU ME SERVES LE  
FICHIER LESOIR.BE/INDEX.PHP ET SES  
FICHIERS ASSOCIÉS.

serveur web



81.246.2.9

• LES VOICI!



Etape 2

client

195.13.233.456.dhcp-belgacom.be (adresse dynamique fournie par l'ISP lors de la connexion)

# DONC, POUR METTRE UN SITE EN LIGNE

# DONC, POUR METTRE UN SITE EN LIGNE

- il faut trois choses :



# DONC, POUR METTRE UN SITE EN LIGNE

- il faut trois choses :

1. un nom de domaine (ex: "becode.org" ) chez un enregistreur de domaines («name registrar»)

# DONC, POUR METTRE UN SITE EN LIGNE

- il faut trois choses :
  1. un nom de domaine (ex: "becode.org" ) chez un enregistreur de domaines («name registrar»)
  2. un espace sur un serveur pour héberger ses fichiers (un «hébergeur»)

# DONC, POUR METTRE UN SITE EN LIGNE

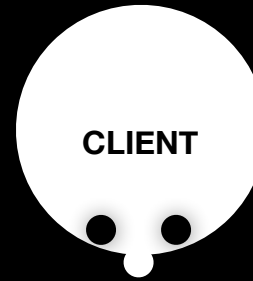
- il faut trois choses :
  1. un nom de domaine (ex: "becode.org" ) chez un enregistreur de domaines («name registrar»)
  2. un espace sur un serveur pour héberger ses fichiers (un «hébergeur»)
  3. les fichiers constituant le site en lui-même

## **schéma en 5", par deux**

Groupe et organise sous forme de couches superposées les différentes technologies et concepts suivants:

html, javascript, php, clientside, css,  
serveur apache, serveur dns, flash, WebGL,  
jquery, jQuery plugin, python, ruby on  
rails, serveur mysql, fichier .htaccess,  
node.js, serverside, ASP, TCP/IP .

# LE WEB EST UNE LASAGNE



CLIENT

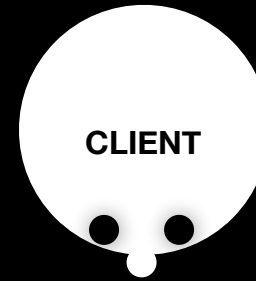
le web browser

le web server

# LE WEB EST UNE LASAGNE



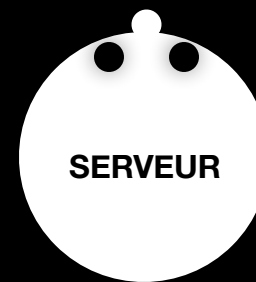
# LE WEB EST UNE LASAGNE



CLIENT

le web browser

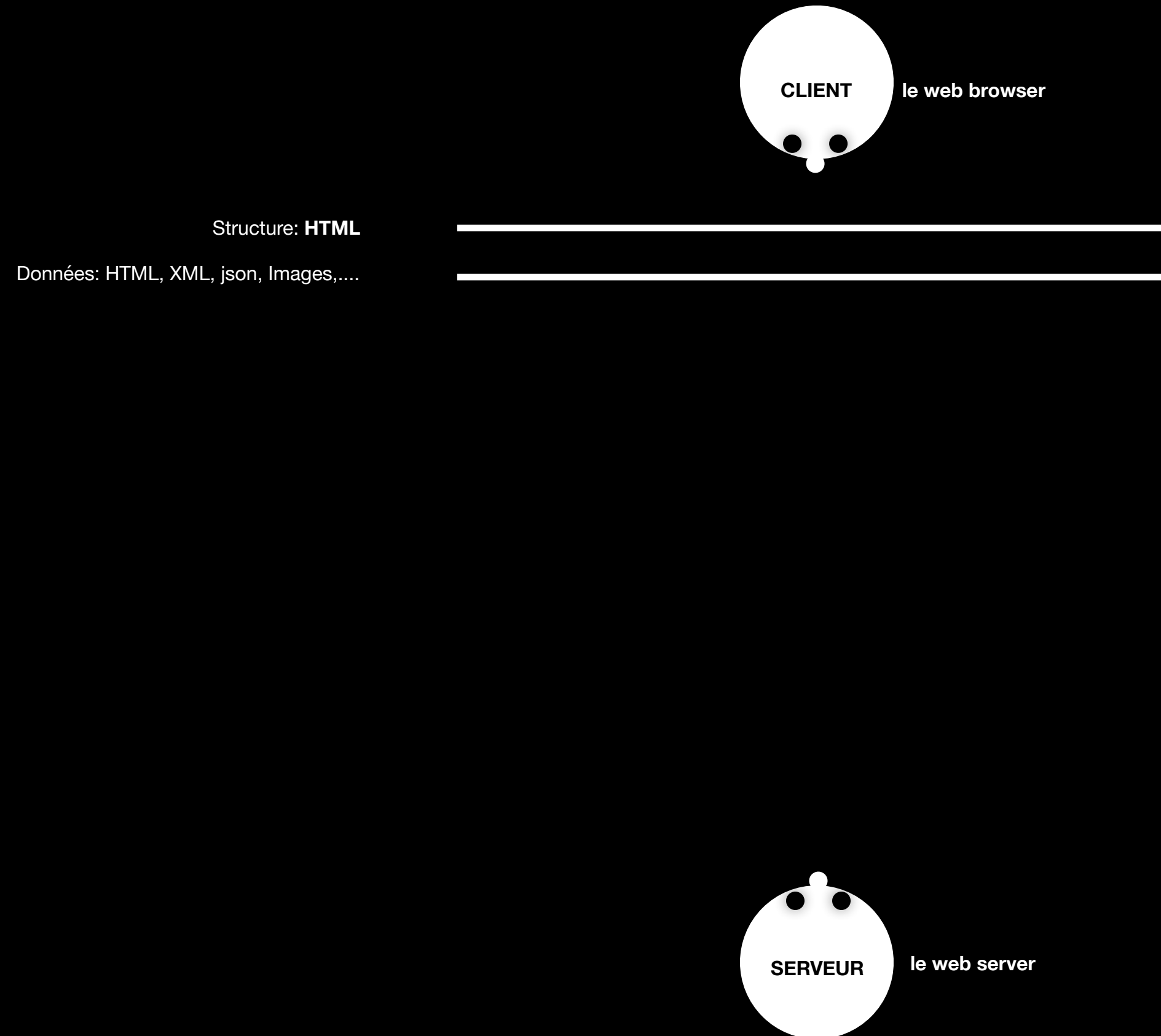
Structure: **HTML**



SERVEUR

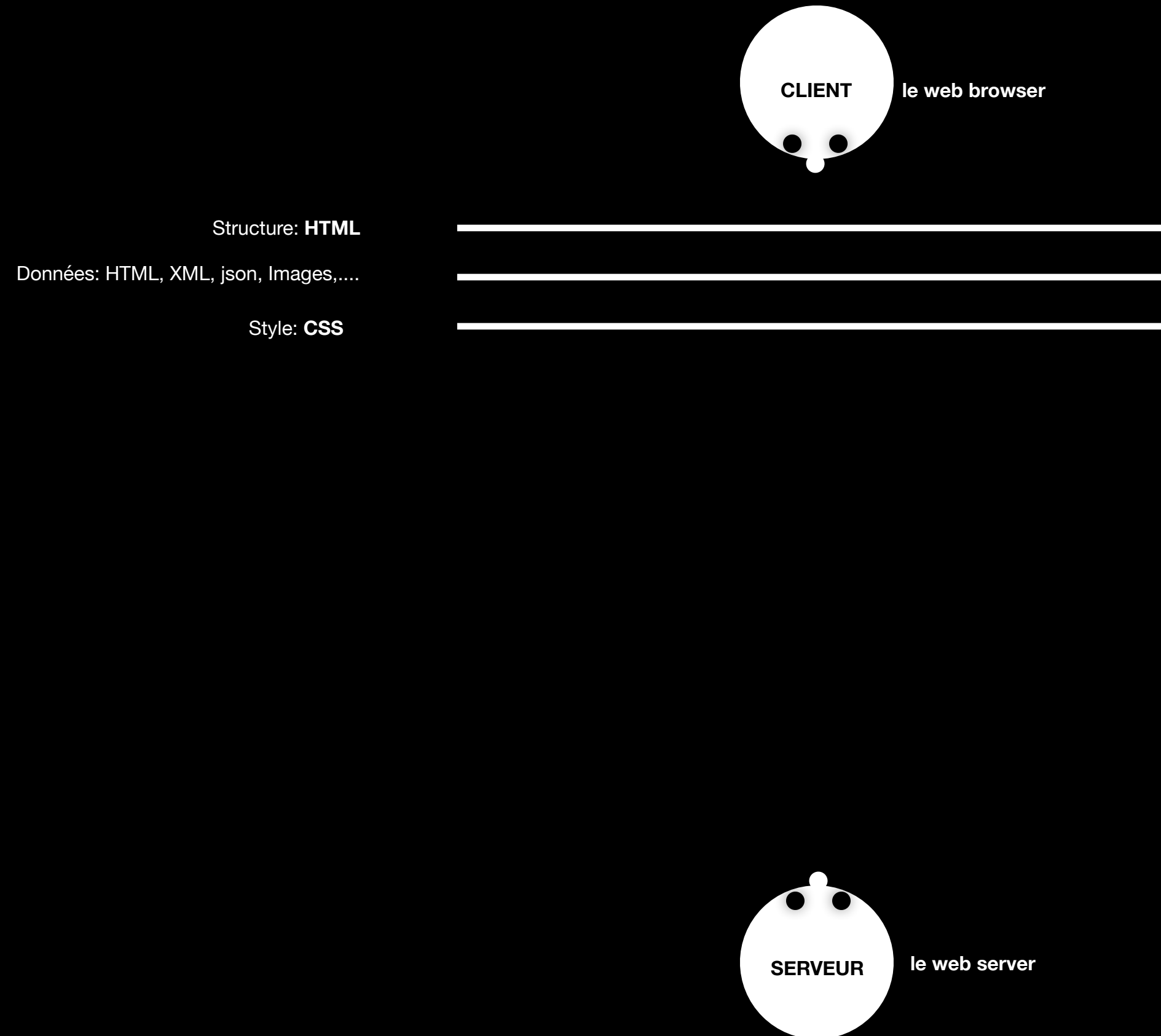
le web server

# LE WEB EST UNE LASAGNE

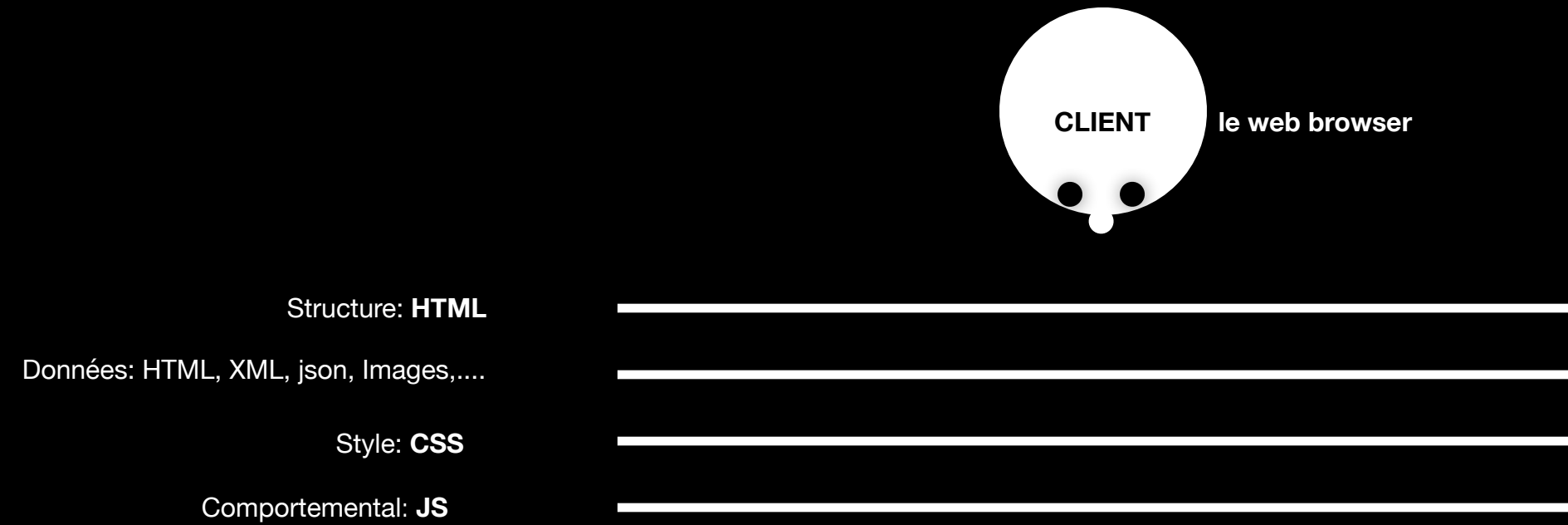




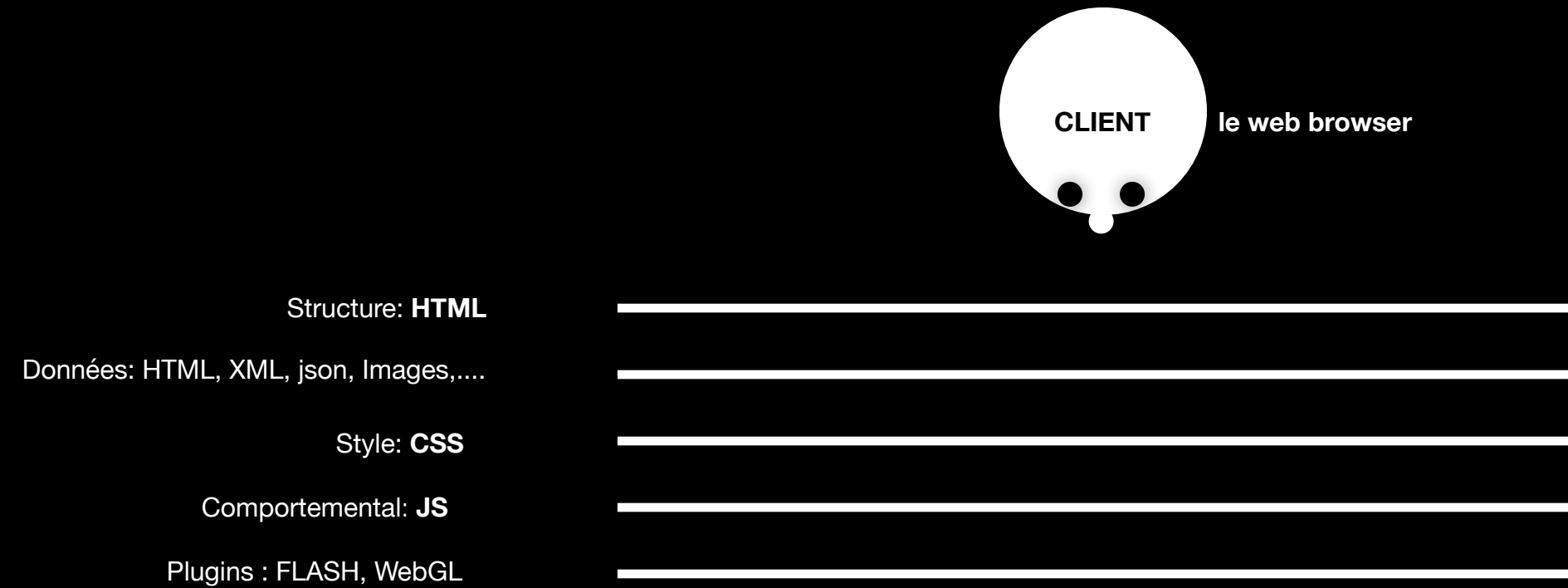
# LE WEB EST UNE LASAGNE



# LE WEB EST UNE LASAGNE



# LE WEB EST UNE LASAGNE



# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

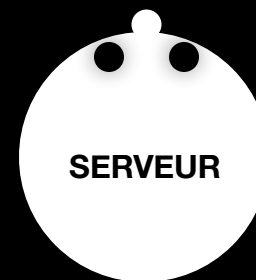
Données: HTML, XML, json, Images,....

Style: **CSS**

Comportemental: **JS**

Plugins : FLASH, WebGL

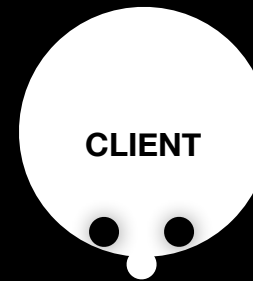
processing script : **PHP**, Python, ASP, RoR...



SERVEUR

le web server

# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

Données: HTML, XML, json, Images,....

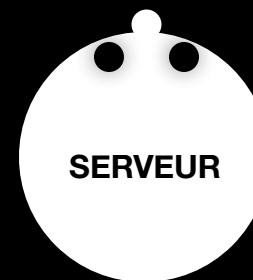
Style: **CSS**

Comportemental: **JS**

Plugins : FLASH, WebGL

processing script : **PHP**, Python, ASP, RoR...

Base de données: **MySQL**, Oracle...



SERVEUR

le web server

# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

Données: HTML, XML, json, Images,....

Style: **CSS**

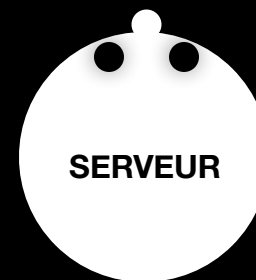
Comportemental: **JS**

Plugins : FLASH, WebGL

processing script : **PHP**, Python, ASP, RoR...

Base de données: **MySQL**, Oracle...

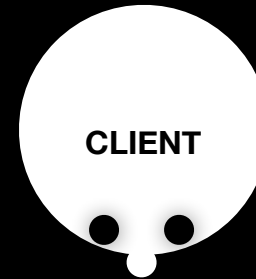
Serveur: **APACHE** ou **NGINX**



SERVEUR

le web server

# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

Données: HTML, XML, json, Images,....

Style: **CSS**

Comportemental: **JS**

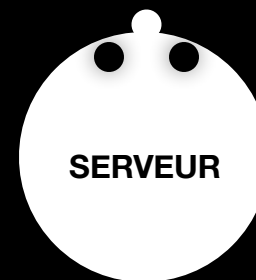
Plugins : FLASH, WebGL

processing script : **PHP**, Python, ASP, RoR...

Base de données: **MySQL**, Oracle...

Serveur: **APACHE** ou **NGINX**

DNS: BIND



SERVEUR

le web server

# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

Données: HTML, XML, json, Images,....

Style: **CSS**

Comportemental: **JS**

Plugins : FLASH, WebGL

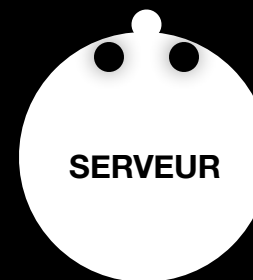
clientside

processing script : **PHP**, Python, ASP, RoR...

Base de données: **MySQL**, Oracle...

Serveur: **APACHE** ou **NGINX**

DNS: BIND



SERVEUR

le web server



# LE WEB EST UNE LASAGNE



CLIENT

le web browser

Structure: **HTML**

Données: HTML, XML, json, Images,....

Style: **CSS**

Comportemental: **JS**

Plugins : FLASH, WebGL

clientside

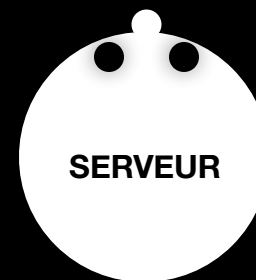
processing script : **PHP**, Python, ASP, RoR...

Base de données: **MySQL**, Oracle...

Serveur: **APACHE** ou **NGINX**

DNS: BIND

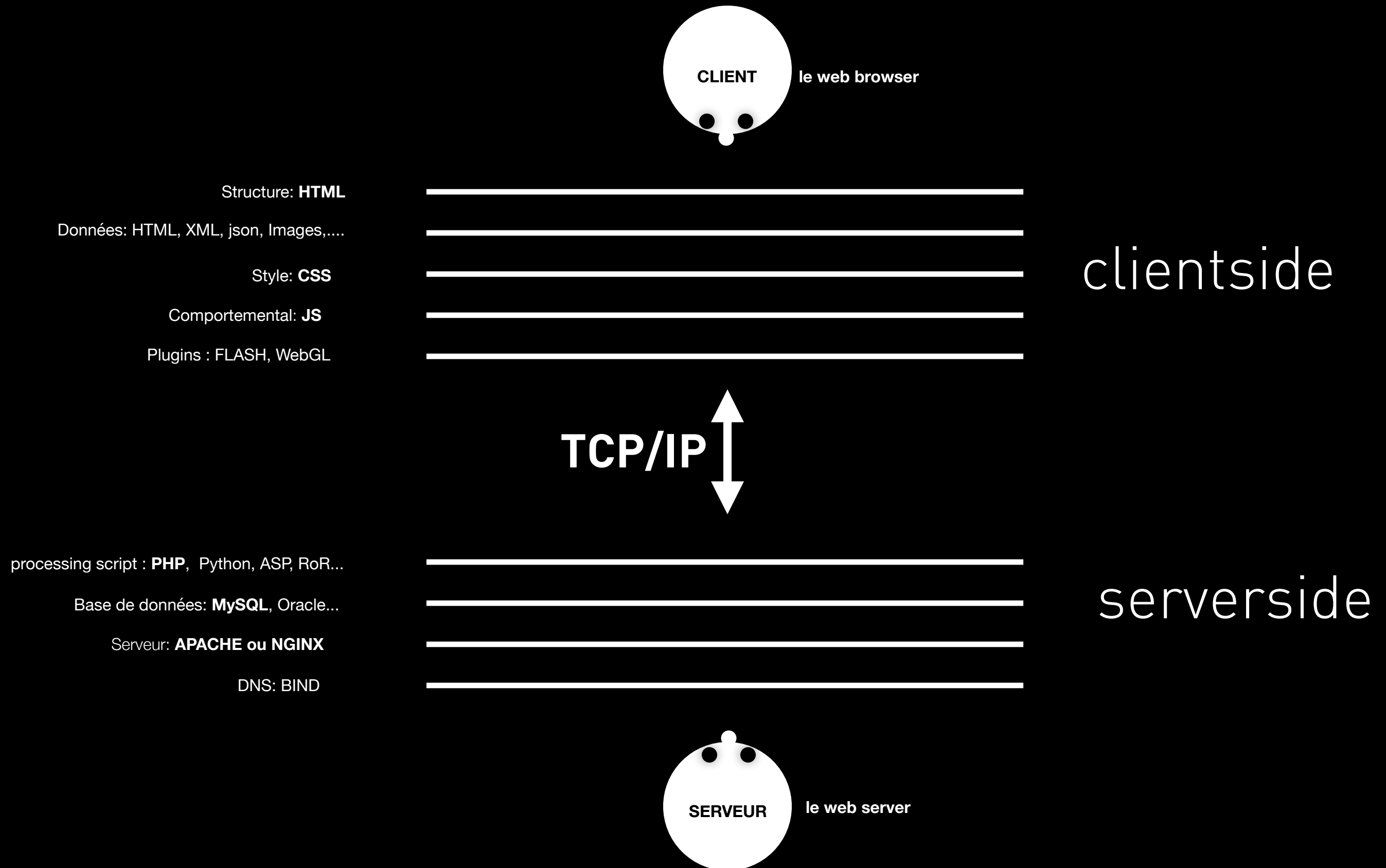
serverside



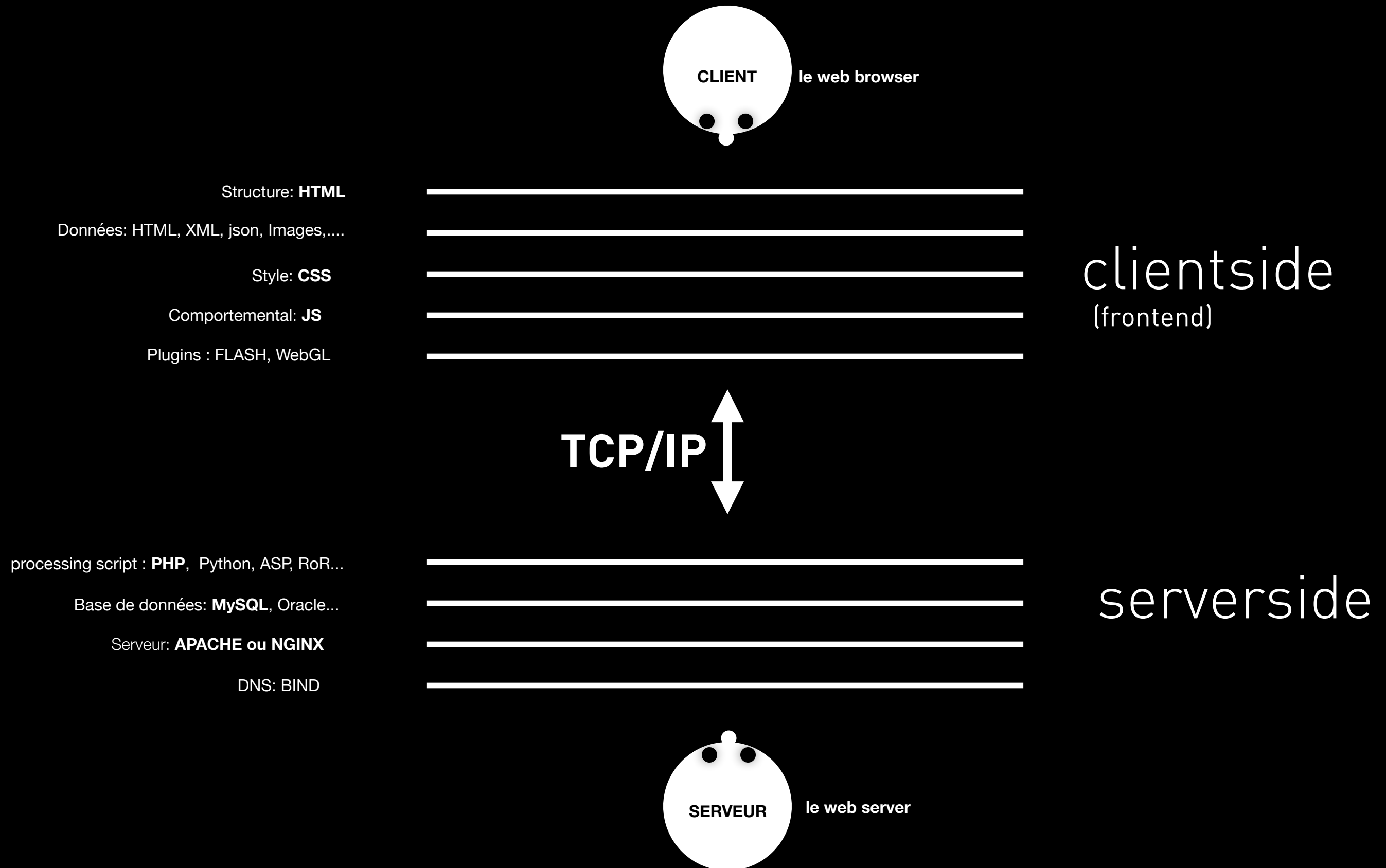
SERVEUR

le web server

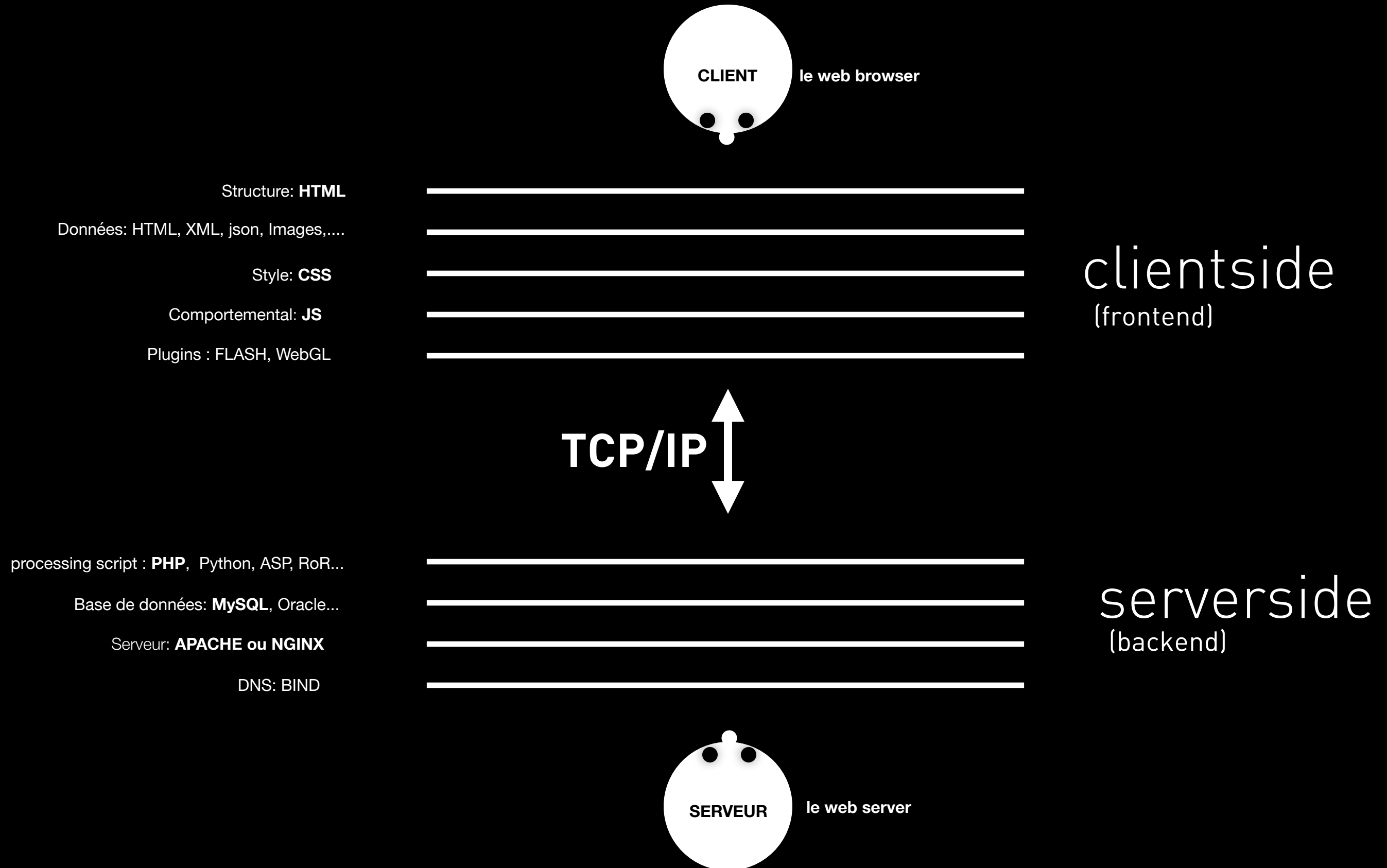
# LE WEB EST UNE LASAGNE



# LE WEB EST UNE LASAGNE



# LE WEB EST UNE LASAGNE



cet empilement de technologies entraîne une complexité  
qui nous amène à un concept important...

# AMÉLIORATION PROGRESSIVE



# AMÉLIORATION PROGRESSIVE





# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.





# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

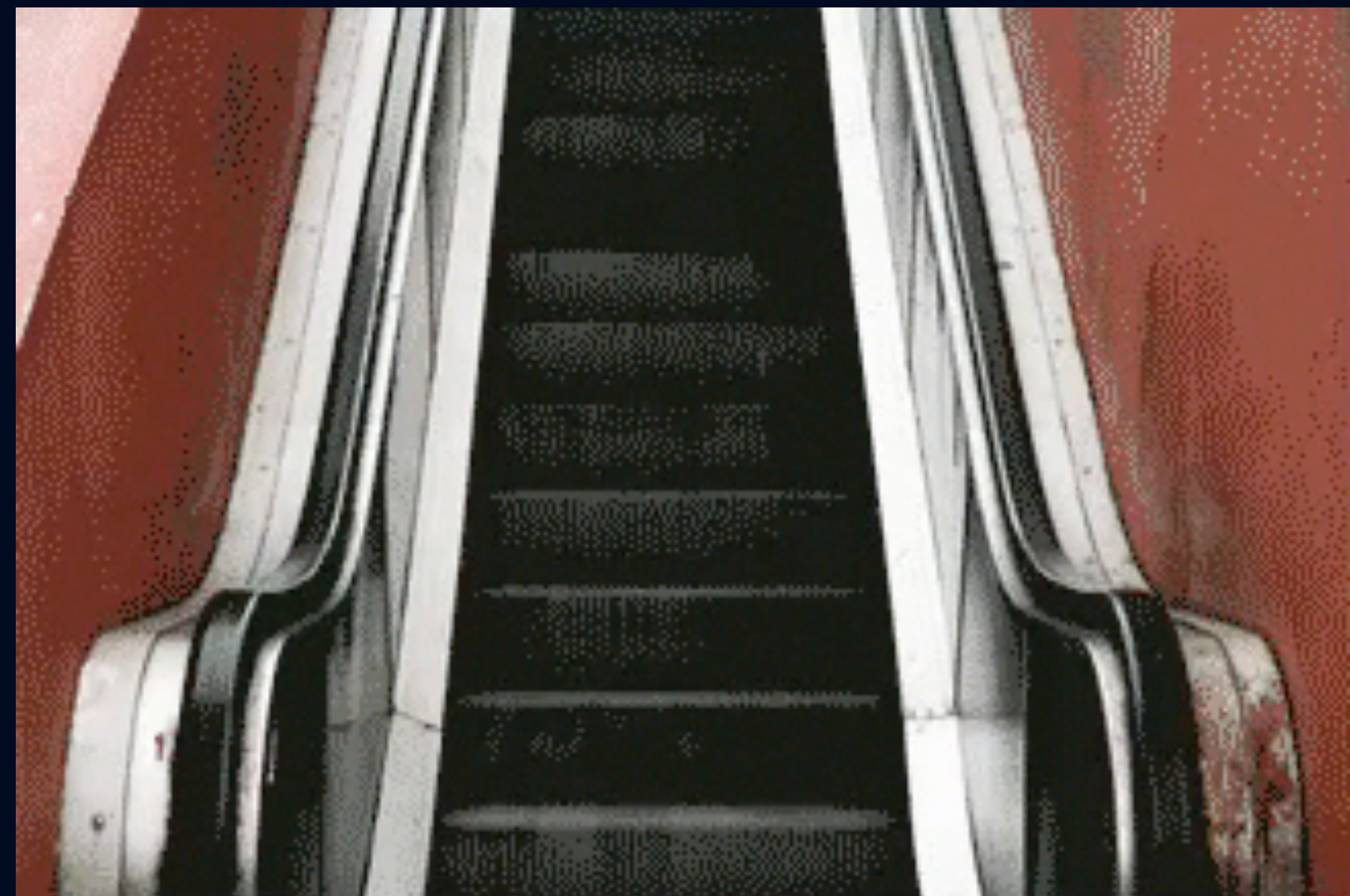


# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

On code pour le pire scénario, càd, un navigateur qui ne disposerait que du html (pas de css ni de JS) peut accéder à l'information essentielle du site / la fonctionnalité de base.





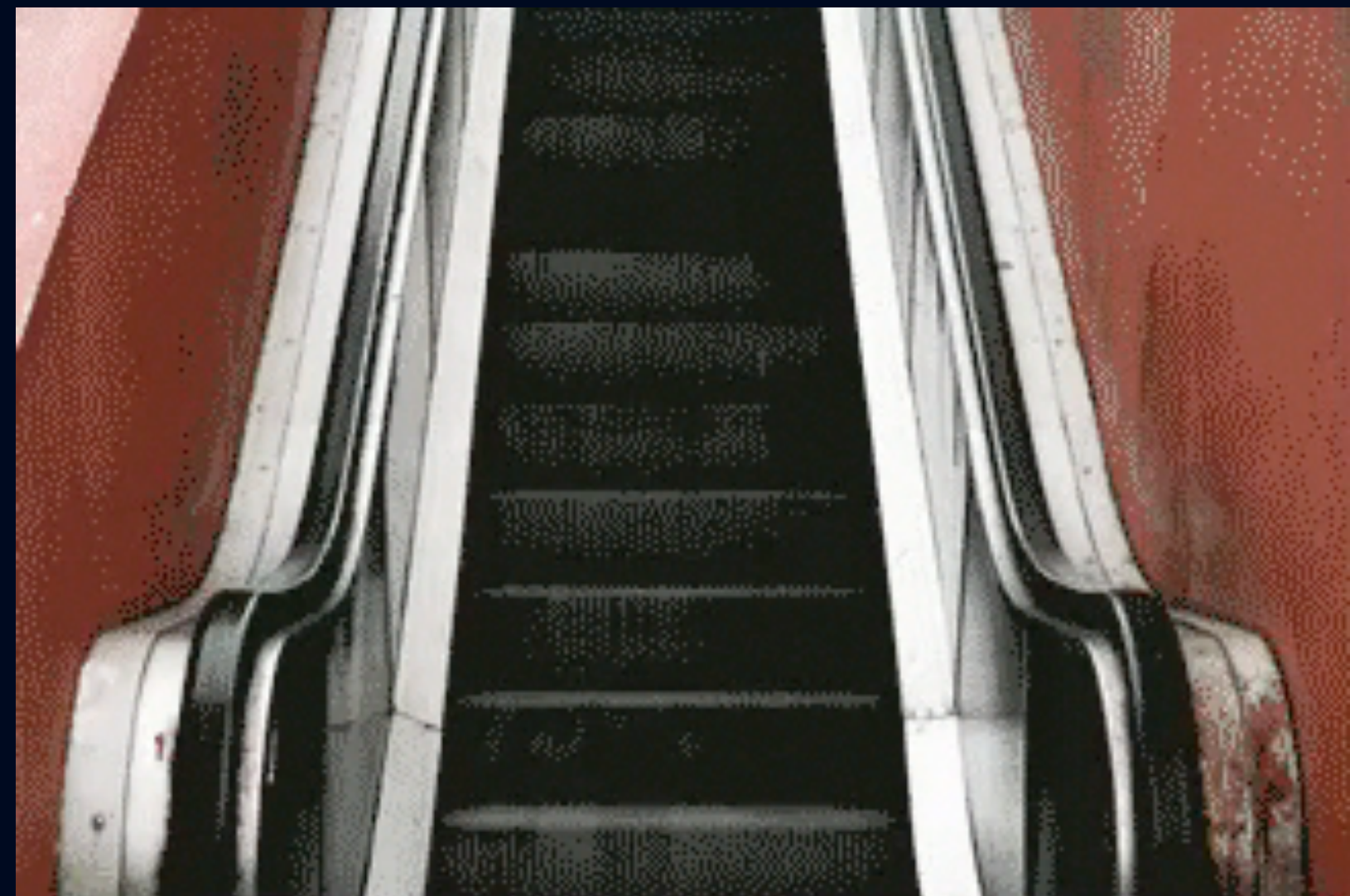
# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

On code pour le pire scénario, càd, un navigateur qui ne disposerait que du html (pas de css ni de JS) peut accéder à l'information essentielle du site / la fonctionnalité de base.

Si disponible, CSS améliore le look et l'ergonomie.



# AMÉLIORATION PROGRESSIVE

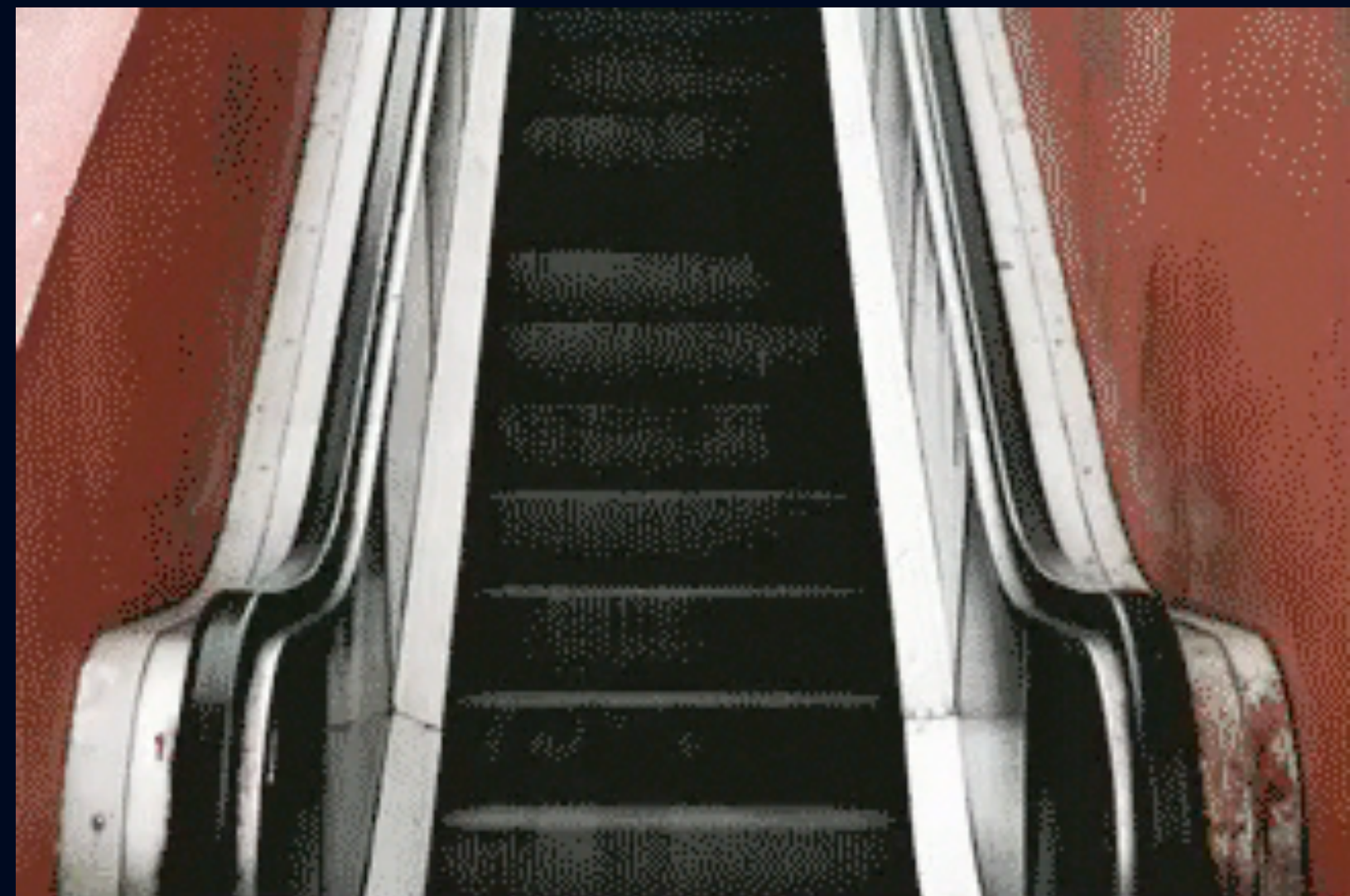
Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

On code pour le pire scénario, càd, un navigateur qui ne disposerait que du html (pas de css ni de JS) peut accéder à l'information essentielle du site / la fonctionnalité de base.

Si disponible, CSS améliore le look et l'ergonomie.

Si disponible, JS améliore l'ergonomie et fluidifie l'expérience.





# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

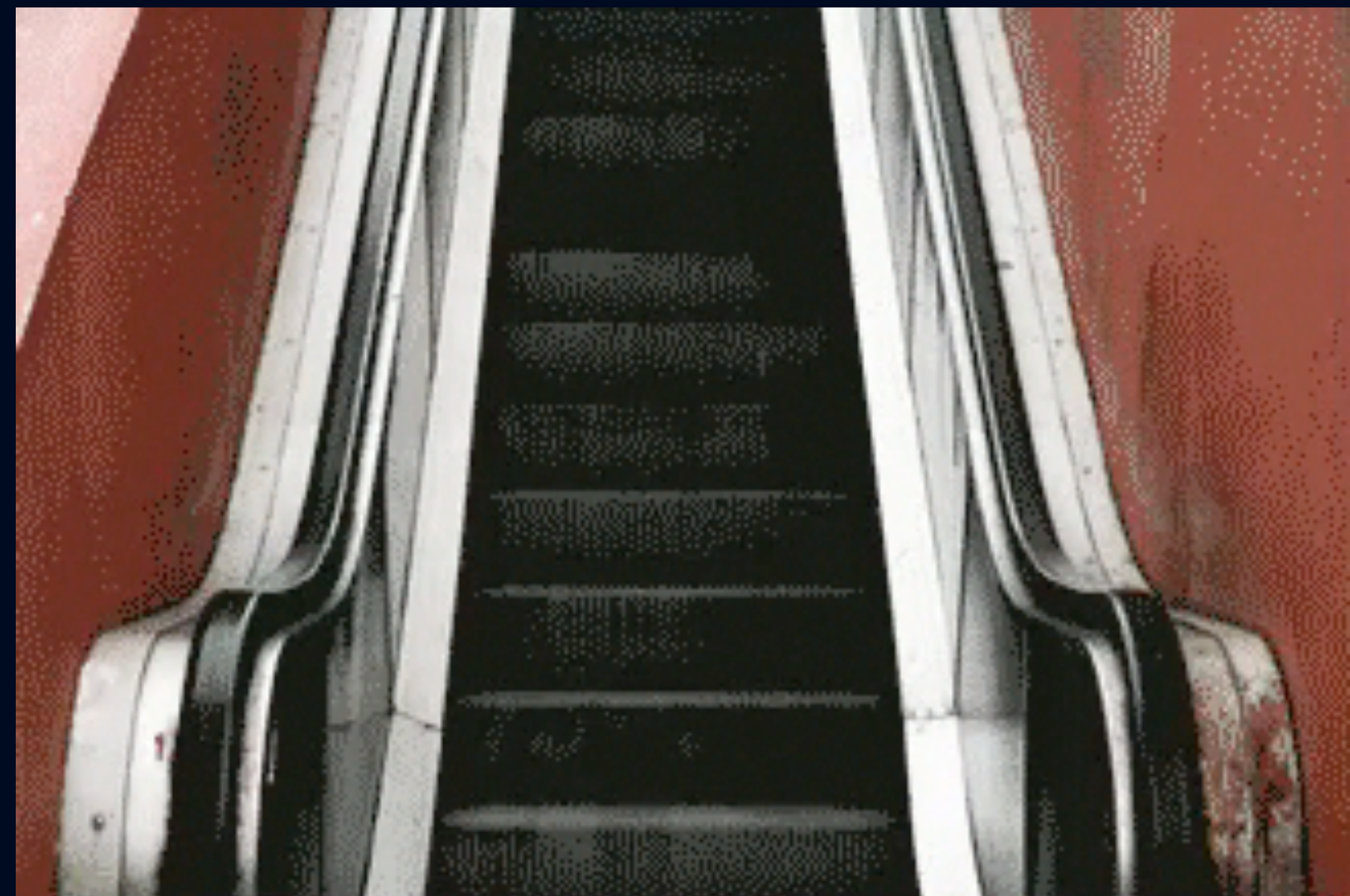
L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

On code pour le pire scénario, càd, un navigateur qui ne disposerait que du html (pas de css ni de JS) peut accéder à l'information essentielle du site / la fonctionnalité de base.

Si disponible, CSS améliore le look et l'ergonomie.

Si disponible, JS améliore l'ergonomie et fluidifie l'expérience.

«progressive enhancement» versus «graceful degradation»



# AMÉLIORATION PROGRESSIVE

Quelle que soit la configuration technique de l'utilisateur, il doit avoir accès à l'information.

L'information doit être accessible pour tous, même le fonctionnaire du SPF Finances coincé sur son vieil IBM avec écran TFT 800x600 vert navigant sur Internet Explorer 4.

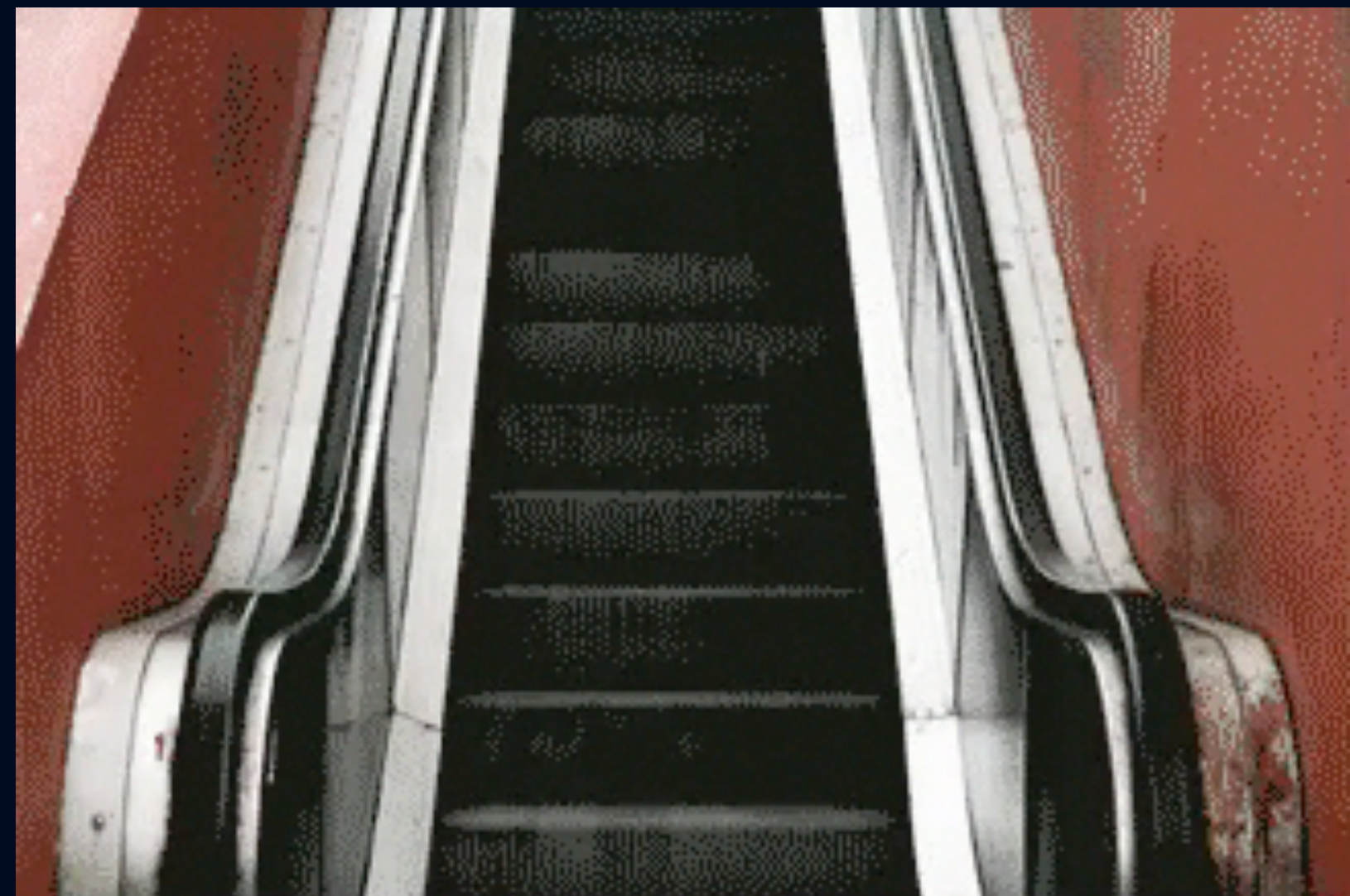
On code pour le pire scénario, càd, un navigateur qui ne disposerait que du html (pas de css ni de JS) peut accéder à l'information essentielle du site / la fonctionnalité de base.

Si disponible, CSS améliore le look et l'ergonomie.

Si disponible, JS améliore l'ergonomie et fluidifie l'expérience.

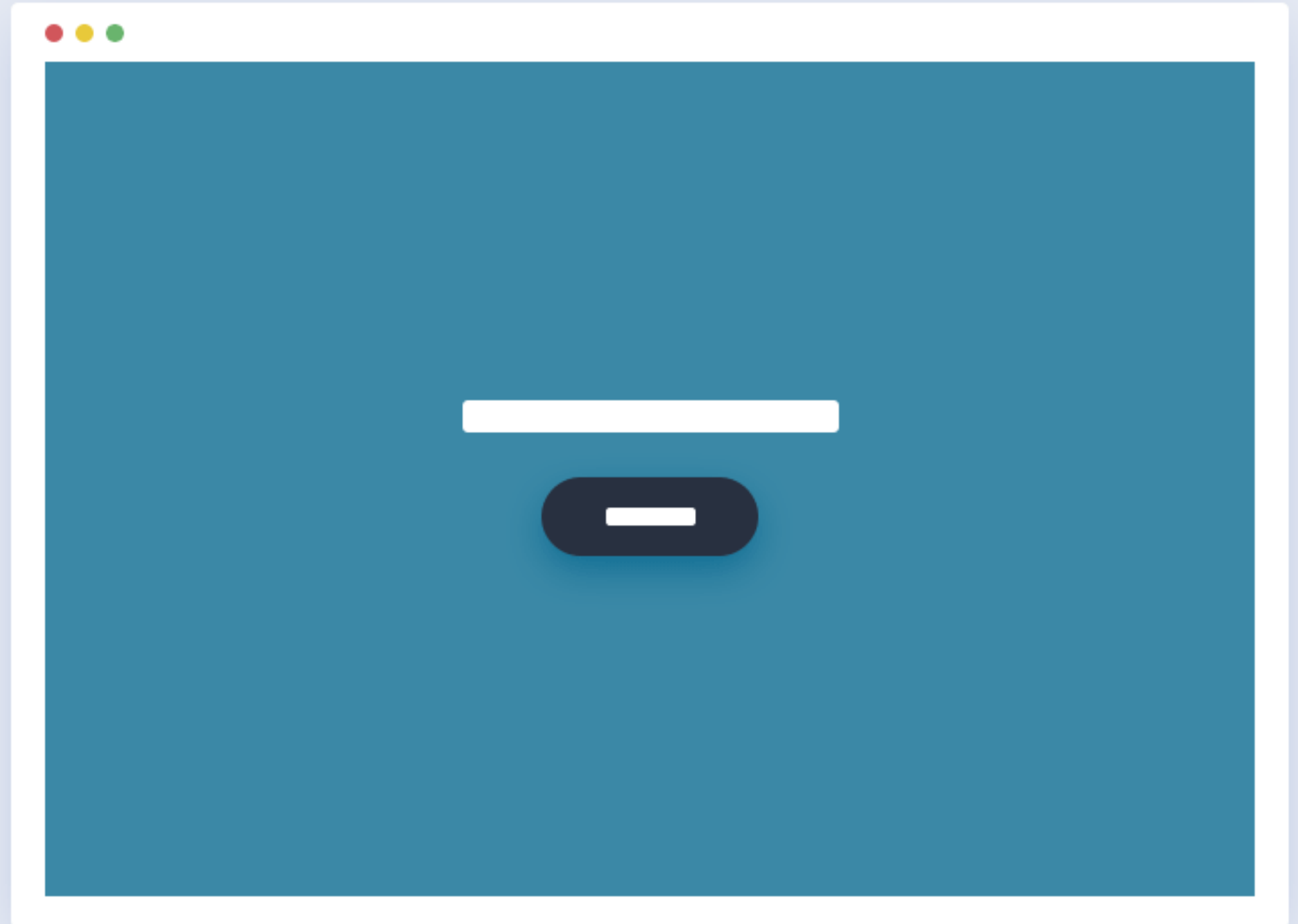
«progressive enhancement» versus «graceful degradation»

cela rend votre travail "futureproof"



# EXERCICE

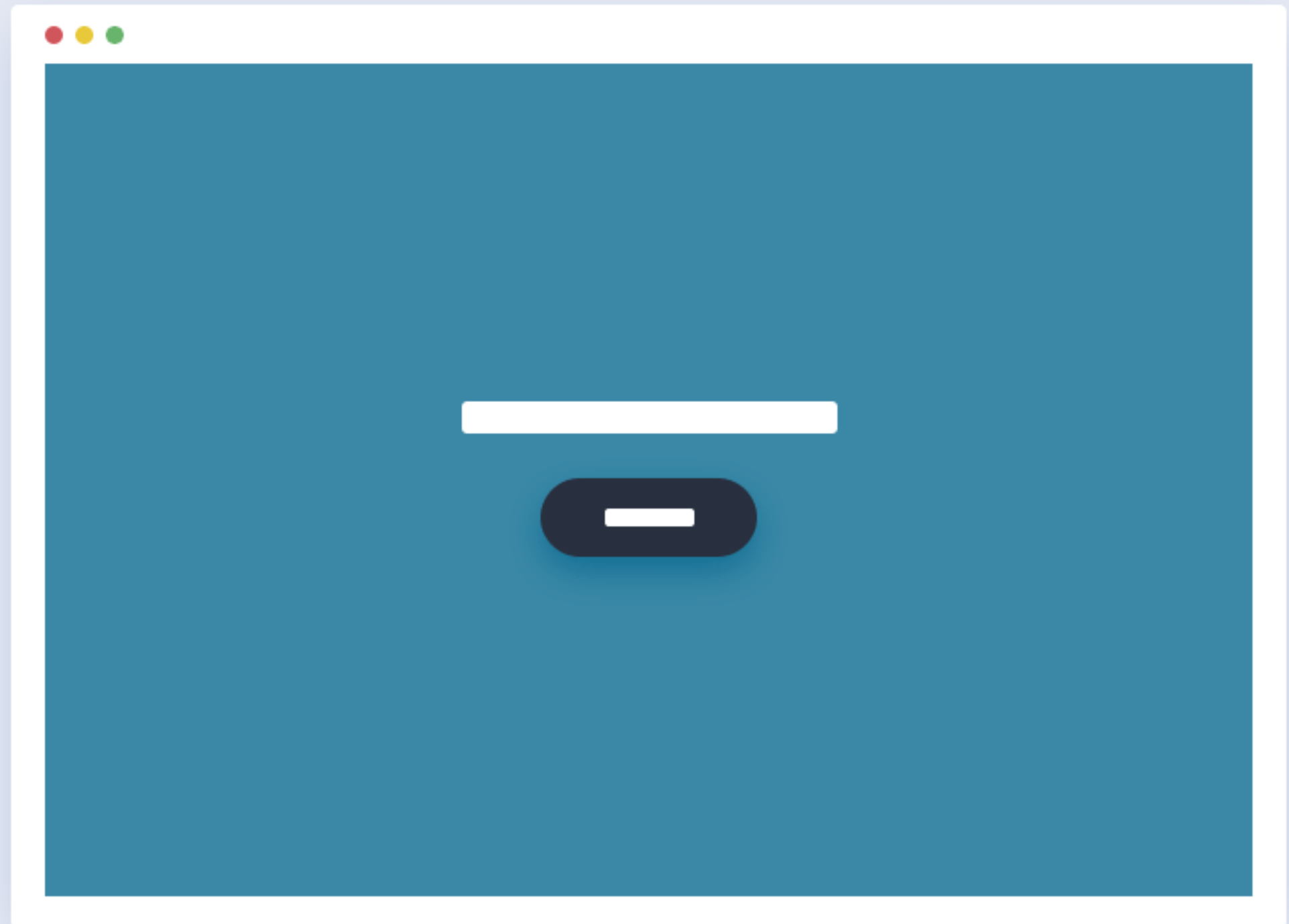
- Ton client souhaite qu'un écran "chargement en cours" avec une jolie animation s'affiche durant le chargement de chaque page de son site.
- Comment mettre cela en place en respectant l'amélioration progressive ?
- Décris la manière dont tu le ferais, de sorte que l'employé du SPF tout comme le web designer californien aient tous deux accès à l'information via la meilleure UX possible en fonction de leur configuration technique.





# EXERCICE

- Ton client souhaite qu'un écran "chargement en cours" avec une jolie animation s'affiche durant le chargement de chaque page de son site.
- Comment mettre cela en place en respectant l'amélioration progressive ?
- Décris la manière dont tu le ferais, de sorte que l'employé du SPF tout comme le web designer californien aient tous deux accès à l'information via la meilleure UX possible en fonction de leur configuration technique.





# PHP Development Server

(depuis php 5.4)

# PHP BUILT-IN DEVELOPMENT SERVER

- ouvrir le Terminal

- taper

`php -S localhost:5000`

- visiter <http://localhost:5000>

- Bien pour les petits projets ne nécessitant pas de base de données.

```
Last login: Mon Jan 26 00:30:03 on ttys001
You have mail.
Airdealexandre:~ alexandreplennevaux$ php -S localhost:8000
PHP 5.6.2 Development Server started at Mon Jan 26 10:00:57 2015
Listening on http://localhost:8000
Document root is /Users/alexandreplennevaux
Press Ctrl-C to quit.
[Mon Jan 26 10:01:12 2015] ::1:52428 [200]: /
[Mon Jan 26 10:01:13 2015] ::1:52429 [200]: /Accessibilite%CC%81%20web_files/reveal.css
[Mon Jan 26 10:01:13 2015] ::1:52430 [200]: /Accessibilite%CC%81%20web_files/anysurfer.css
[Mon Jan 26 10:01:13 2015] ::1:52431 [200]: /Accessibilite%CC%81%20web_files/simple.css
[Mon Jan 26 10:01:13 2015] ::1:52432 [200]: /Accessibilite%CC%81%20web_files/fullscreen-img.css
[Mon Jan 26 10:01:13 2015] ::1:52433 [200]: /Accessibilite%CC%81%20web_files/zenburn.css
[Mon Jan 26 10:01:13 2015] ::1:52434 [200]: /Accessibilite%CC%81%20web_files/head.js
[Mon Jan 26 10:01:13 2015] ::1:52435 [200]: /Accessibilite%CC%81%20web_files/reveal.js
[Mon Jan 26 10:01:13 2015] ::1:52436 [200]: /Accessibilite%CC%81%20web_files/highlight.js
[Mon Jan 26 10:01:13 2015] ::1:52437 [200]: /Accessibilite%CC%81%20web_files/zoom.js
[Mon Jan 26 10:01:13 2015] ::1:52438 [200]: /Accessibilite%CC%81%20web_files/notes.js
[Mon Jan 26 10:01:13 2015] ::1:52439 [200]: /Accessibilite%CC%81%20web_files/jquery.js
[Mon Jan 26 10:01:13 2015] ::1:52440 [200]: /Accessibilite%CC%81%20web_files/insertion_modules.js
[Mon Jan 26 10:01:13 2015] ::1:52441 [200]: /Accessibilite%CC%81%20web_files/logo_AS_square.png
[Mon Jan 26 10:01:13 2015] ::1:52442 [200]: /Accessibilite%CC%81%20web_files/logo_as_baseline_FR.png
[Mon Jan 26 10:01:13 2015] ::1:52443 [200]: /Accessibilite%CC%81%20web_files/labeled.jpg
[Mon Jan 26 10:01:13 2015] ::1:52444 [200]: /Accessibilite%CC%81%20web_files/opleidingslokaal.jpg
[Mon Jan 26 10:01:13 2015] ::1:52445 [200]: /Accessibilite%CC%81%20web_files/erkendebouwers.jpg
[Mon Jan 26 10:01:13 2015] ::1:52446 [200]: /Accessibilite%CC%81%20web_files/overzicht_scholen_2013.png
[Mon Jan 26 10:01:13 2015] ::1:52447 [200]: /Accessibilite%CC%81%20web_files/whitecane.jpg
[Mon Jan 26 10:01:13 2015] ::1:52448 [200]: /Accessibilite%CC%81%20web_files/parking.png
[Mon Jan 26 10:01:13 2015] ::1:52449 [200]: /Accessibilite%CC%81%20web_files/rolstoelingang.png
[Mon Jan 26 10:01:13 2015] ::1:52450 [200]: /Accessibilite%CC%81%20web_files/telecoil.png
[Mon Jan 26 10:01:13 2015] ::1:52451 [200]: /Accessibilite%CC%81%20web_files/draaideur.png
[Mon Jan 26 10:01:13 2015] ::1:52452 [200]: /Accessibilite%CC%81%20web_files/browser_logo.png
[Mon Jan 26 10:01:13 2015] ::1:52453 [200]: /Accessibilite%CC%81%20web_files/smartphones.png
[Mon Jan 26 10:01:13 2015] ::1:52454 [200]: /Accessibilite%CC%81%20web_files/docs.png
[Mon Jan 26 10:01:13 2015] ::1:52455 [200]: /Accessibilite%CC%81%20web_files/Woman_using_computer_bajstock.jpg
[Mon Jan 26 10:01:13 2015] ::1:52456 [200]: /Accessibilite%CC%81%20web_files/who.png
```

php built-in

<http://www.sitepoint.com/taking-advantage-of-phps-built-in-server/>

# PHP: HYPERTEXT PREPROCESSOR

# PREMIER SCRIPT EN PHP

```
<h1>Ceci</h1>
```

```
<?php echo "<p>Hello World</p>"; ?>
```

```
<h1>... est la même chose que ceci:</h1>
```

```
<p>Hello World</p>
```

## INSTRUCTIONS

- à la racine de ton serveur, crée un fichier test.php
- écris le code à gauche
- Va sur <http://localhost/test.php>
- observe le code source de la page générée.

# PREMIER SCRIPT EN PHP

```
<h1>Ceci</h1>
```

```
<?php echo "<p>Hello World</p>"; ?>
```

```
<h1>... est la même chose que ceci:</h1>
```

```
<p>Hello World</p>
```

## INSTRUCTIONS

- à la racine de ton serveur, crée un fichier test.php
- écris le code à gauche
- Va sur <http://localhost/test.php>
- observe le code source de la page générée.

## QU' OBSERVES - TU?

# PREMIER SCRIPT EN PHP

```
<h1>Ceci</h1>
```

```
<?php echo "<p>Hello World</p>"; ?>
```

```
<h1>... est la même chose que ceci:</h1>
```

```
<p>Hello World</p>
```

## INSTRUCTIONS

- à la racine de ton serveur, crée un fichier test.php
- écris le code à gauche
- Va sur <http://localhost/test.php>
- observe le code source de la page générée.

## QU' OBSERVES - TU?

- html et php sont mélangés

# PREMIER SCRIPT EN PHP

```
<h1>Ceci</h1>
```

```
<?php echo "<p>Hello World</p>"; ?>
```

```
<h1>... est la même chose que ceci:</h1>
```

```
<p>Hello World</p>
```

## INSTRUCTIONS

- à la racine de ton serveur, crée un fichier test.php
- écris le code à gauche
- Va sur <http://localhost/test.php>
- observe le code source de la page générée.

## QU' OBSERVES - TU?

- html et php sont mélangés
- le php commence par `<?php` et termine par `?>`

# PREMIER SCRIPT EN PHP

```
<h1>Ceci</h1>
```

```
<?php echo "<p>Hello World</p>"; ?>
```

```
<h1>... est la même chose que ceci:</h1>
```

```
<p>Hello World</p>
```

## INSTRUCTIONS

- à la racine de ton serveur, crée un fichier test.php
- écris le code à gauche
- Va sur <http://localhost/test.php>
- observe le code source de la page générée.

## QU' OBSERVES - TU?

- html et php sont mélangés
- le php commence par `<?php` et termine par `?>`
- echo sert à afficher ce qui vient immédiatement à sa droite.



# PHP: HYPERTEXT PREPROCESSOR

# PHP: HYPERTEXT PREPROCESSOR

Langage généraliste et Open Source, conçu pour le web. Il peut être intégré facilement au HTML.

Le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat.

[Lire la définition complète](#)

Le savais-tu? Facebook est programmé en php.

# VARIABLES

une variable est un **conteneur**.

Comme une boîte en carton, elle peut contenir n'importe quoi: des livres, des objets, ou même des pizzas.

```
$boite = "livres";  
$boite = "objets";  
$boite = "pizza";
```



# «JEAN-PIERRE»

```
<?php $prenom = 'Jean-Pierre'; ?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<?php
// on change la valeur de la variable
$prenom = 'Marie-Chantale'; ?>

<h1>Salut <?php echo $prenom ; ?> ! Cha va?</h1>
```

// Visite la page. Tu dois voir ceci:

## Résultat

```
<h1>Salut Jean-Pierre ! Cha va?</h1>
<h1>Salut Marie-Chantale ! Cha va?</h1>
```

# «JEAN-PIERRE»

```
<?php $prenom = 'Jean-Pierre'; ?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<?php
// on change la valeur de la variable
$prenom = 'Marie-Chantale'; ?>

<h1>Salut <?php echo $prenom ; ?> ! Cha va?</h1>
```

// Visite la page. Tu dois voir ceci:

## Résultat

```
<h1>Salut Jean-Pierre ! Cha va?</h1>
<h1>Salut Marie-Chantale ! Cha va?</h1>
```

## INSTRUCTIONS

- écris le code à gauche
- Va sur <http://localhost:8000/test.php>

# «JEAN-PIERRE»

```
<?php $prenom = 'Jean-Pierre'; ?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<?php
// on change la valeur de la variable
$prenom = 'Marie-Chantale'; ?>

<h1>Salut <?php echo $prenom ; ?> ! Cha va?</h1>

// Visite la page. Tu dois voir ceci:
```

## Résultat

```
<h1>Salut Jean-Pierre ! Cha va?</h1>
<h1>Salut Marie-Chantale ! Cha va?</h1>
```

## INSTRUCTIONS

- écris le code à gauche
- Va sur <http://localhost:8000/test.php>

## OBSERVATIONS

- une variable commence par le signe \$
- je suis libre de choisir le nom de la variable (a-zA-Z1-9\_ ). Pas d'accents.
- sensible à la casse:  
\$Prenom ≠ \$prenom
- La variable, ici, contient du texte. Il est donc mis entre guillemets simples. (')
- Tu peux changer la valeur d'une variable durant l'exécution de ton script.

# «JEAN-PIERRE»

```
<?php $prenom = 'Jean-Pierre'; ?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<?php
// on change la valeur de la variable
$prenom = 'Marie-Chantale'; ?>

<h1>Salut <?php echo $prenom ; ?> ! Cha va?</h1>
```

**EXERCICE:**  
 Salue une troisième personne de  
 ton choix par son prénom.

```
// Visite la page. Tu dois voir ceci.
```

## Résultat

```
<h1>Salut Jean-Pierre ! Cha va?</h1>
<h1>Salut Marie-Chantale ! Cha va?</h1>
```

## INSTRUCTIONS

- écris le code à gauche
- Va sur <http://localhost:8000/test.php>

## OBSERVATIONS

Une variable commence par le signe \$  
 Tu es libre de choisir le nom de la  
 variable (a-zA-Z1-9\_). Pas d'accents.

- sensible à la casse:  
 \$Prenom ≠ \$prenom
- La variable, ici, contient du texte. Il est donc mis entre guillemets simples. (')
- Tu peux changer la valeur d'une variable durant l'exécution de ton script.

**une variable est utile lorsqu'un  
contenu doit pouvoir changer**

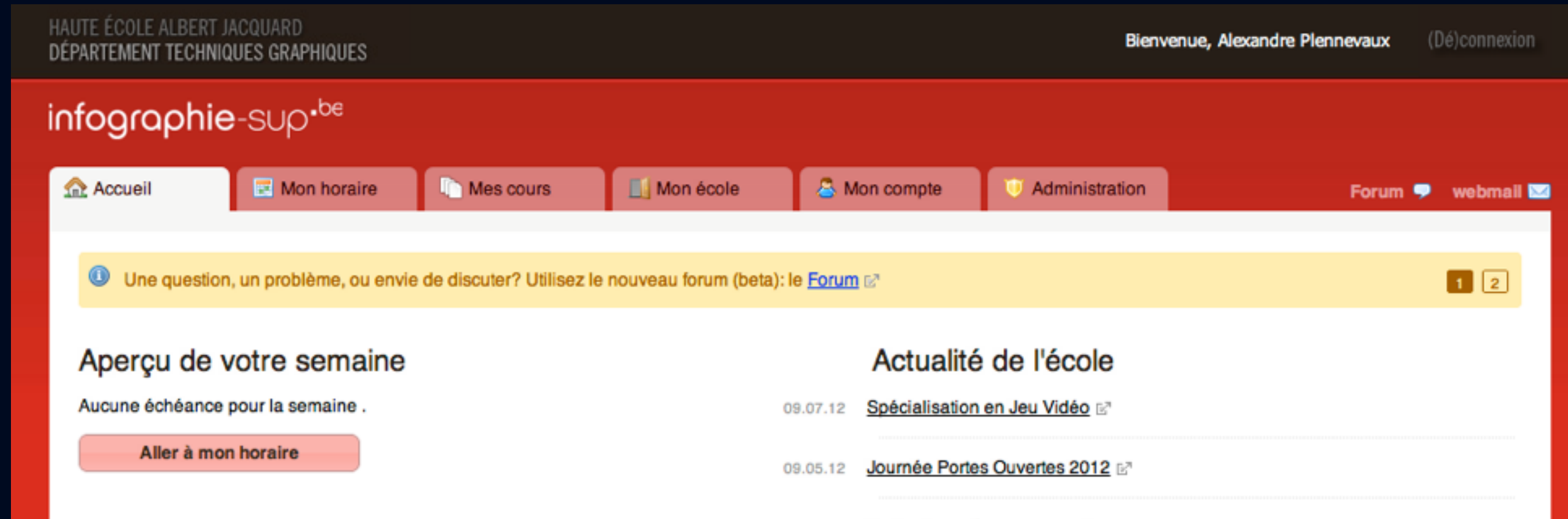
**= contenu «dynamique»**  
(opposé au contenu dit «statique»)



# EXEMPLE D'APPLICATION

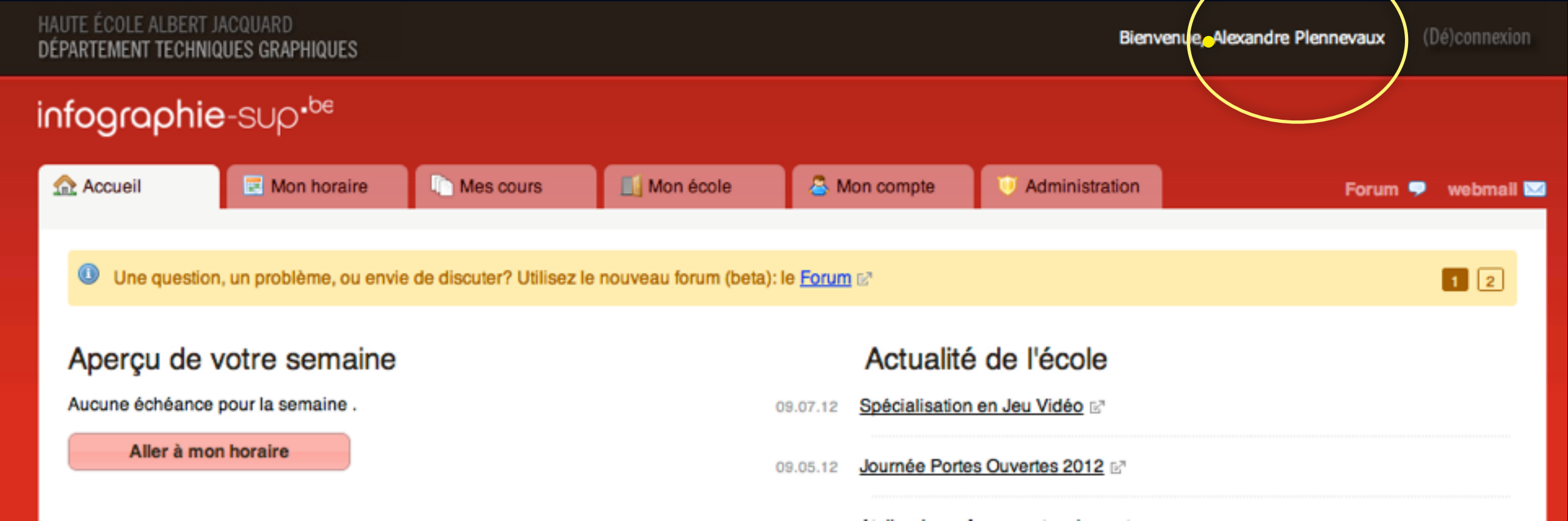
# EXEMPLE D'APPLICATION

variable

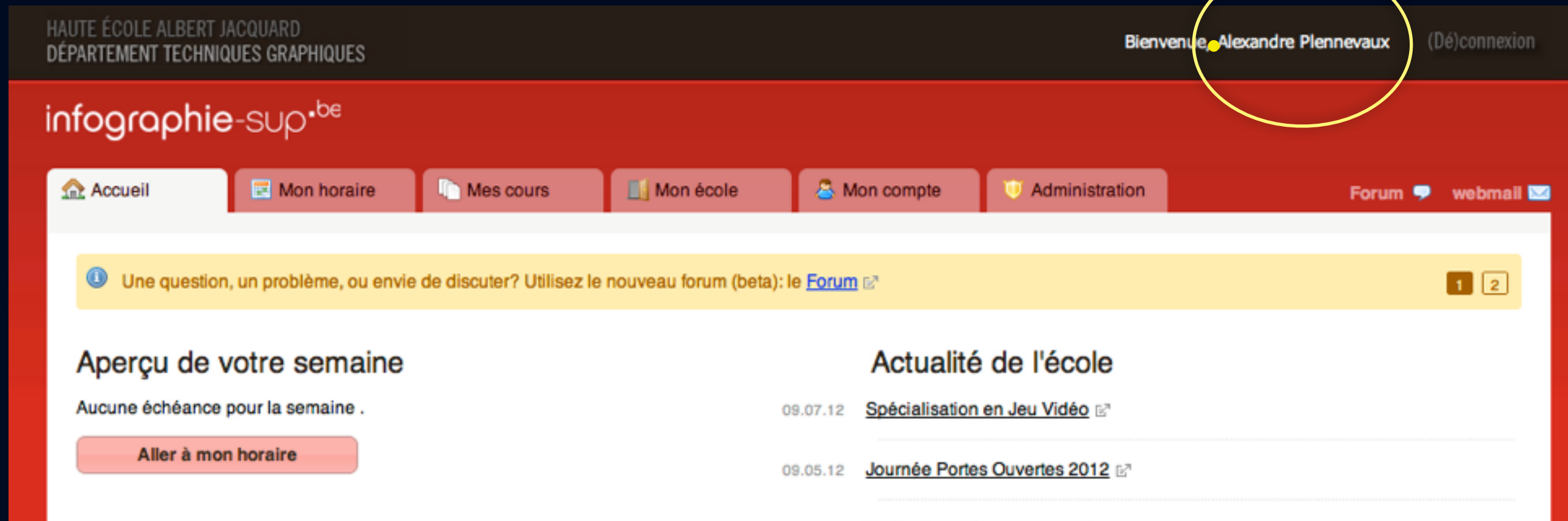


# EXEMPLE D'APPLICATION

variable



# EXEMPLE D'APPLICATION



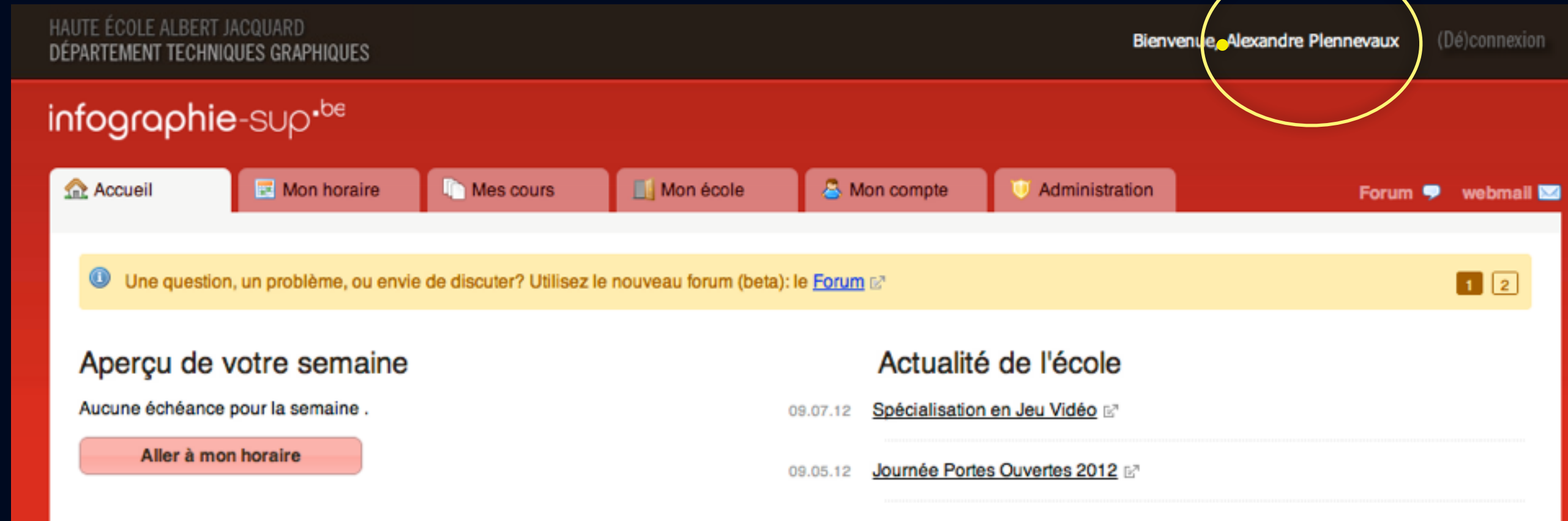
```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if(LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>logout/" id="logout"><b>Déconnexion</b></a>
      <span id="nomprenom">Bienvenue, <?php echo $firstname, ' ', $lastname; ?></span>
      <?php
      }
      ?>

    </div>
    <div id="logo-holder" class="clearfix">
      <a id="logo1" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>Extranet - HEAJ</small></a>
      <a id="logo2" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>infographie-sup.be</small></a>
    </div>
  </div>
```

# EXEMPLE D'APPLICATION

variable



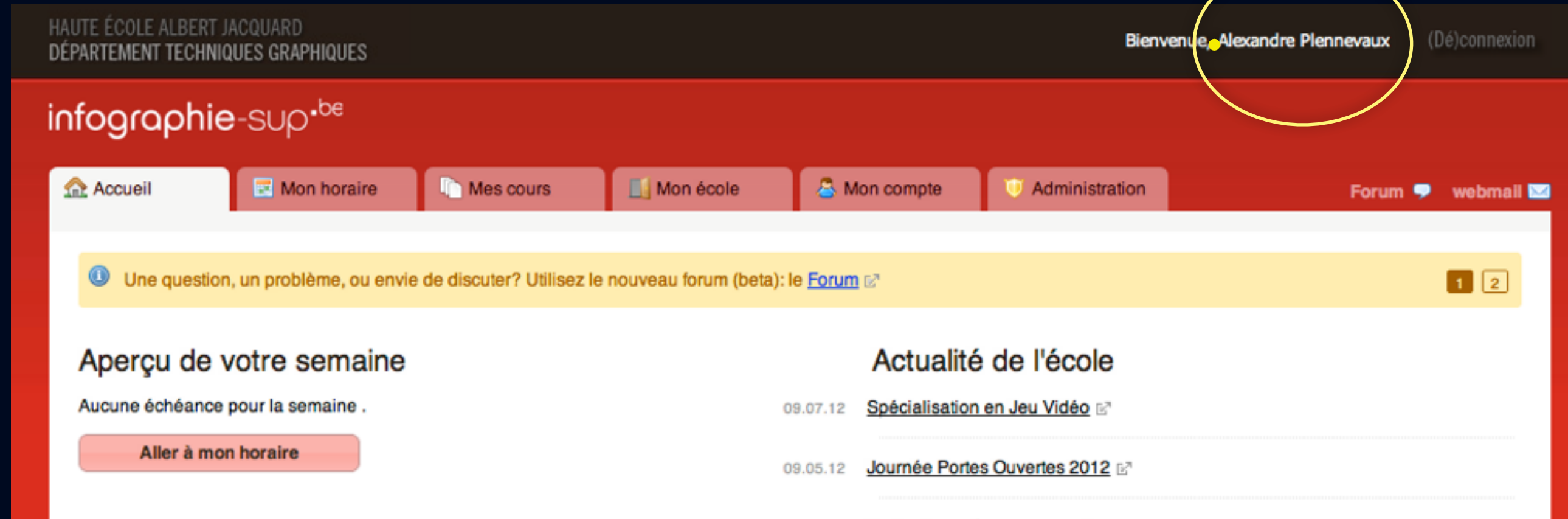
```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if(LOGGED_IN) { ?>
        <a href="<?php echo WWWROOT ?>logout/" id="logout"><b>Déconnexion</b></a>
        <span id="nomprenom">Bienvenue, <?php echo $firstname, ' ', $lastname; ?></span>
      <?php
    }
  <?>
</div>
<div id="logo-holder" class="clearfix">
  <a id="logo1" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>Extranet - HEAJ</small></a>
  <a id="logo2" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>infographie-sup.be</small></a>
</div>
```



# EXEMPLE D'APPLICATION

variable



```
<div id="page">
  <header class="clearfix">
    <div id="infotop_right">
      <?php if(!LOGGED_IN) { ?>
        <a href="<?php echo WWWROOT ?>logout/" id="logout"><b>Déconnexion</b></a>
        <span id="nomprenom">Bienvenue, <?php echo $firstname, ' ', $lastname; ?></span>
        <?php
        }
      ?>
    </div>
    <div id="logo-holder" class="clearfix">
      <a id="logo1" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>Extranet - HEAJ</small></a>
      <a id="logo2" title="Retour à l'accueil" href="<?php echo WWWROOT ?>" class="clearfix"><small>infographie-sup.be</small></a>
    </div>
  </div>
```

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if(LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
    </div>
```

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if (LOGGED_IN) { ?>
        <a href="<?php echo WWWROOT ?>"
        <span id="nomprenom">Bienvenue
          <?php
        }
      ?>
    </div>
  </div>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.



# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if (LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
    </div>
  </header>
</div>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.
- Exemples:

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if (LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
    </div>
  </header>
</div>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.
- Exemples:
  - état connecté de l'utilisateur

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if (LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
    </div>
  </header>
</div>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.
- Exemples:
  - état connecté de l'utilisateur
  - nom de la base de données

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if (LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
    </div>
  </header>
</div>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.
- Exemples:
  - état connecté de l'utilisateur
  - nom de la base de données
- Par convention, on les indique en MAJUSCULE et sans le dollar. On les crée via la fonction **define('NOM', 'valeur');**

# CONSTANTE

```
<div id="page">
  <header class="clearfix">
    <div id="info_top_right">

      <?php if(LOGGED_IN) { ?>
      <a href="<?php echo WWWROOT ?>"
      <span id="nomprenom">Bienvenue
      <?php
      }
      ?>
```

- on utilise les « constantes » pour stocker une valeur qui ne doit pas changer durant le traitement du script.
- Exemples:
  - état connecté de l'utilisateur
  - nom de la base de données
- Par convention, on les indique en MAJUSCULE et sans le dollar. On les crée via la fonction **define('NOM', 'valeur');**
- Ex:

```
<?php
```

```
define('DIRECTEUR_HEAJ', 'Joel Jacobs');
```

```
echo DIRECTEUR_HEAJ;
```

```
?>
```

JEAN-PIERRE ♡...

# JEAN-PIERRE ♥...

```
<?php
    $prenom = 'Jean-Pierre';

?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<p>Il paraît que tu adooooores <?php echo $aime; ?>. C'est
vrai?</p>
```

## INSTRUCTIONS

- modifie le code en ajoutant la variable \$aime.
- Utilise-la pour désigner ce que JP ... aime.

# JEAN-PIERRE ♥...

```
<?php
    $prenom = 'Jean-Pierre';

?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<p>Il paraît que tu adooooores <?php echo $aime; ?>. C'est
vrai?</p>
```

## INSTRUCTIONS

- modifie le code en ajoutant la variable \$aime.
- Utilise-la pour désigner ce que JP ... aime.

## OBSERVATIONS

- lorsque j'ai plus qu'une ligne de code, je dois séparer chaque ligne par un point-virgule ( ; )



# JEAN-PIERRE ♥...

```
<?php
    $prenom = 'Jean-Pierre';
    $aime = 'les chicons au gratin';

?>

<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>
<p>Il paraît que tu adooooores <?php echo $aime; ?>. C'est
vrai?</p>
```

## INSTRUCTIONS

- modifie le code en ajoutant la variable \$aime.
- Utilise-la pour désigner ce que JP ... aime.

## OBSERVATIONS

- lorsque j'ai plus qu'une ligne de code, je dois séparer chaque ligne par un point-virgule ( ; )

# QUI ON VEUT

```
<?php
$prenom = $_GET['prenom'];
$aime = $_GET['aime'];
?>
<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>

<p>Il parait que tu adooooores <?php echo $aime; ?>.
C'est vrai?</p>
```

# QUI ON VEUT

```
<?php
$prenom = $_GET['prenom'];
$aime = $_GET['aime'];
?>
<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>

<p>Il paraît que tu adooooores <?php echo $aime; ?>.
C'est vrai?</p>
```

## INSTRUCTIONS

- va à l'url  
<http://localhost/test.php?prenom=Jacques>
- puis, va à l'url  
<http://localhost/test.php?prenom=Jean>
- Enfin, va à l'url  
<http://localhost/test.php?prenom=Jean&aime=ton+école>

# QUI ON VEUT

```
<?php
$prenom = $_GET['prenom'];
$aime = $_GET['aime'];
?>
<h1>Salut <?php echo $prenom; ?> ! Cha va?</h1>

<p>Il paraît que tu adooooores <?php echo $aime; ?>.
C'est vrai?</p>
```

## INSTRUCTIONS

- va à l'url  
<http://localhost/test.php?prenom=Jacques>
- puis, va à l'url  
<http://localhost/test.php?prenom=Jean>
- Enfin, va à l'url  
<http://localhost/test.php?prenom=Jean&aime=ton+école>

## OBSERVATIONS

- Je peux communiquer la valeur d'une variable à PHP via l' URL.
- Je peux communiquer la valeur de plusieurs variables à PHP via l' URL, en séparant chacune par un &
- PHP les stocke sous forme de Tableau dans une variable superglobale \$\_GET
- on en reparle plus tard...

**exercice: crée un formulaire. Utilise la méthode GET (<form method="GET" . . . >)**

input:

TON NOM....

CE QUE TU AIMES...

ENVOYER

output:

**<h1>Jean-Charles aime les cornichons au chocolat</h1>**

**Exercice: crée un formulaire. Cette fois-ci, utilise la méthode POST (au lieu de GET). Qu'est-ce qui change ?**

input:

TON NOM....

CE QUE TU AIMES...

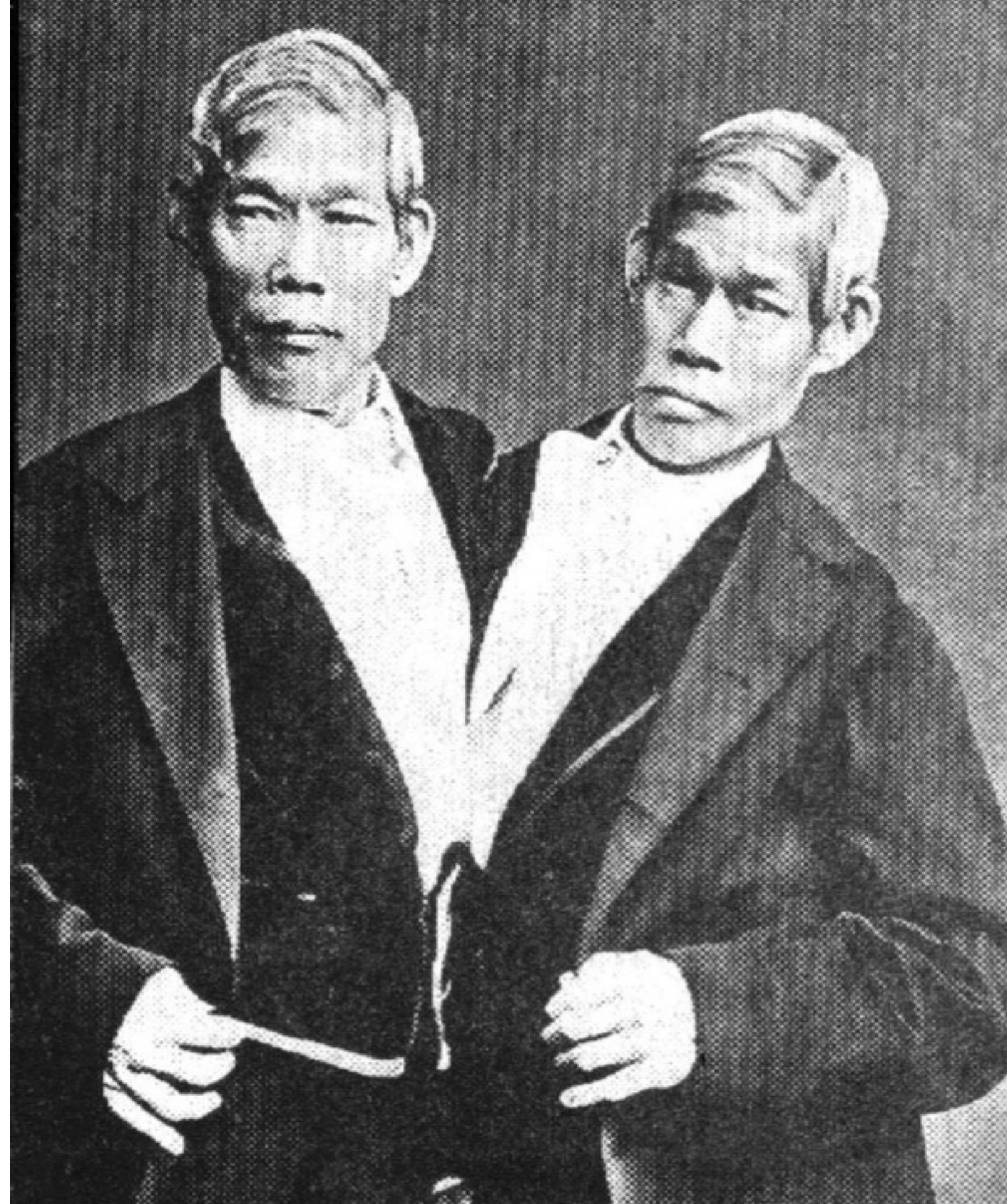
ENVOYER

output:

**<h1>Jean-Charles aime les cornichons au chocolat</h1>**



Quand Jean aperçu Claude,  
il ne pensa plus qu'à se  
**concaténer** avec.  
Ainsi naquit "JeanClaude"



# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaaat ?



# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaaat ?

## INSTRUCTIONS

- concaténer = former un seul texte à partir de deux (ou plus) variables de type "texte".

# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaaat ?

## INSTRUCTIONS

- concaténer = former un seul texte à partir de deux (ou plus) variables de type "texte".

## OBSERVATIONS

- le signe «+» est un opérateur d'addition, pas de concaténation.
- la concaténation se fait via le point (.)

# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaat ?

## SOLUTION

```
<h1>Salut <?php echo $nom1 . $nom2; ?> !
Cha va?</h1>
```

Résultat:

```
<h1>Salut Jeanclaude ! Cha va?</h1>
```

## INSTRUCTIONS

- concaténer = former un seul texte à partir de deux (ou plus) variables de type "texte".

## OBSERVATIONS

- le signe «+» est un opérateur d'addition, pas de concaténation.
- la concaténation se fait via le point (.)

# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaat ?

## EXERCICE:

**Mets un trait d'union entre les  
2 prénoms en utilisant la  
concaténation.**

## SOLUTION

```
<h1>Salut <?php echo $nom1 . $nom2; ?> !
Cha va?</h1>
```

Résultat:

```
<h1>Salut Jeanclaude ! Cha va?</h1>
```

## INSTRUCTIONS

- concaténer = former un seul texte à partir de deux (ou plus) variables de type "texte".

## NOTATIONS

«+» est un opérateur d'addition,  
concaténation.

- la concaténation se fait via le point (.)

# CONCATÉNATION

```
<?php
// Je veux former un bout de texte à partir de 2
variables "texte"

$nom1 = 'Jean';
$nom2 = "Claude";
?>
<h1>Salut <?php echo $nom1 + $nom2; ?> ! Cha va?</h1>
```

Résultat: 0 Whaaaaat ?

## EXERCICE:

```
echo $nom1 . '-' . $nom2 ;
```

concaténation.

## INSTRUCTIONS

- concaténer = former un seul texte à partir de deux (ou plus) variables de type "texte".

## INSTRUCTIONS

«+» est un opérateur d'addition, concaténation.

- la concaténation se fait via le point (.)

```
<h1>Salut <?php echo $nom1 . $nom2; ?> !
Cha va?</h1>
```

Résultat:

```
<h1>Salut Jeanclaude ! Cha va?</h1>
```

## SOLUTION

Tu en sais suffisamment  
pour programmer un  
cadavre exquis  
rudimentaire.

Cadavre Exquis

**Casimir** chopait **Vladimir** depuis  
des lustres.

**Un certain regard** promène l'anus d'une chèvre  
aux Champs-Élysée.

# EXERCICE: EXQUISE CONCATÉNATION

## INSTRUCTIONS

- 10 minutes
- Prépare un formulaire avec 3 entrées: sujet, verbe & complément.
- Passe les paramètres via l'url (method="GET")
- Lorsqu'on soumet le formulaire, affiche le résultat de la concaténation de sujet, verbe et complément.
- Ex: `http://localhost/cadavre.php?sujet=Popol&verbe=mange&complement=un+violon`
- Rappel: pour récupérer les valeurs passées par l'url dans la variable superglobale GET `$_GET['nom-de-la-variable']`

# EXERCICE: EXQUISE CONCATÉNATION

```
$sujet = '';  
$verbe = '';  
$complement = '';
```

```
echo $sujet.' '.$verbe.' '.$complement;
```

## Résultat

Popol mange un violon.

## INSTRUCTIONS

- 10 minutes
- Prépare un formulaire avec 3 entrées: sujet, verbe & complément.
- Passe les paramètres via l'url (method="GET")
- Lorsqu'on soumet le formulaire, affiche le résultat de la concaténation de sujet, verbe et complément.
- Ex: <http://localhost/cadavre.php?sujet=Popol&verbe=mange&complement=un+violon>
- Rappel: pour récupérer les valeurs passées par l'url dans la variable superglobale GET `$_GET['nom-de-la-variable']`



# EXERCICE: EXQUISE CONCATÉNATION

```
$sujet = '';  
$verbe = '';  
$complement = '';  
  
echo $sujet.' '.$verbe.' '.$complement;
```

## Résultat

Popol mange un violon.

```
<?php  
$message = $_GET['sujet'].' '.  
$_GET['verbe'].' '.$_GET['complement'];  
echo $message;  
?>
```

## INSTRUCTIONS

- 10 minutes
- Prépare un formulaire avec 3 entrées: sujet, verbe & complément.
- Passe les paramètres via l'url (method="GET")
- Lorsqu'on soumet le formulaire, affiche le résultat de la concaténation de sujet, verbe et complément.
- Ex: <http://localhost/cadavre.php?sujet=Popol&verbe=mange&complement=un+violon>
- Rappel: pour récupérer les valeurs passées par l'url dans la variable superglobale GET `$_GET['nom-de-la-variable']`

# VARIABLE BOOLÉENNE

```
var $ceCoursEstFacile = true;
```

# VARIABLE BOOLÉENNE

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)
- Ex: La terre est sphérique. Vrai ou Faux?



# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)
- Ex: La terre est sphérique. Vrai ou Faux?
- Ex: La baleine a des dents. Vrai ou Faux?

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)
- Ex: La terre est sphérique. Vrai ou Faux?
- Ex: La baleine a des dents. Vrai ou Faux?

# VARIABLE BOOLÉENNE

- une variable booléenne ne peut contenir que 2 valeurs:
  - soit TRUE (vraie)
  - soit FALSE (fausse)
- Ex: La terre est sphérique. Vrai ou Faux?
- Ex: La baleine a des dents. Vrai ou Faux?
- Il n'y a pas d'autre réponse possible que ... VRAI (true) ou FAUX (false)

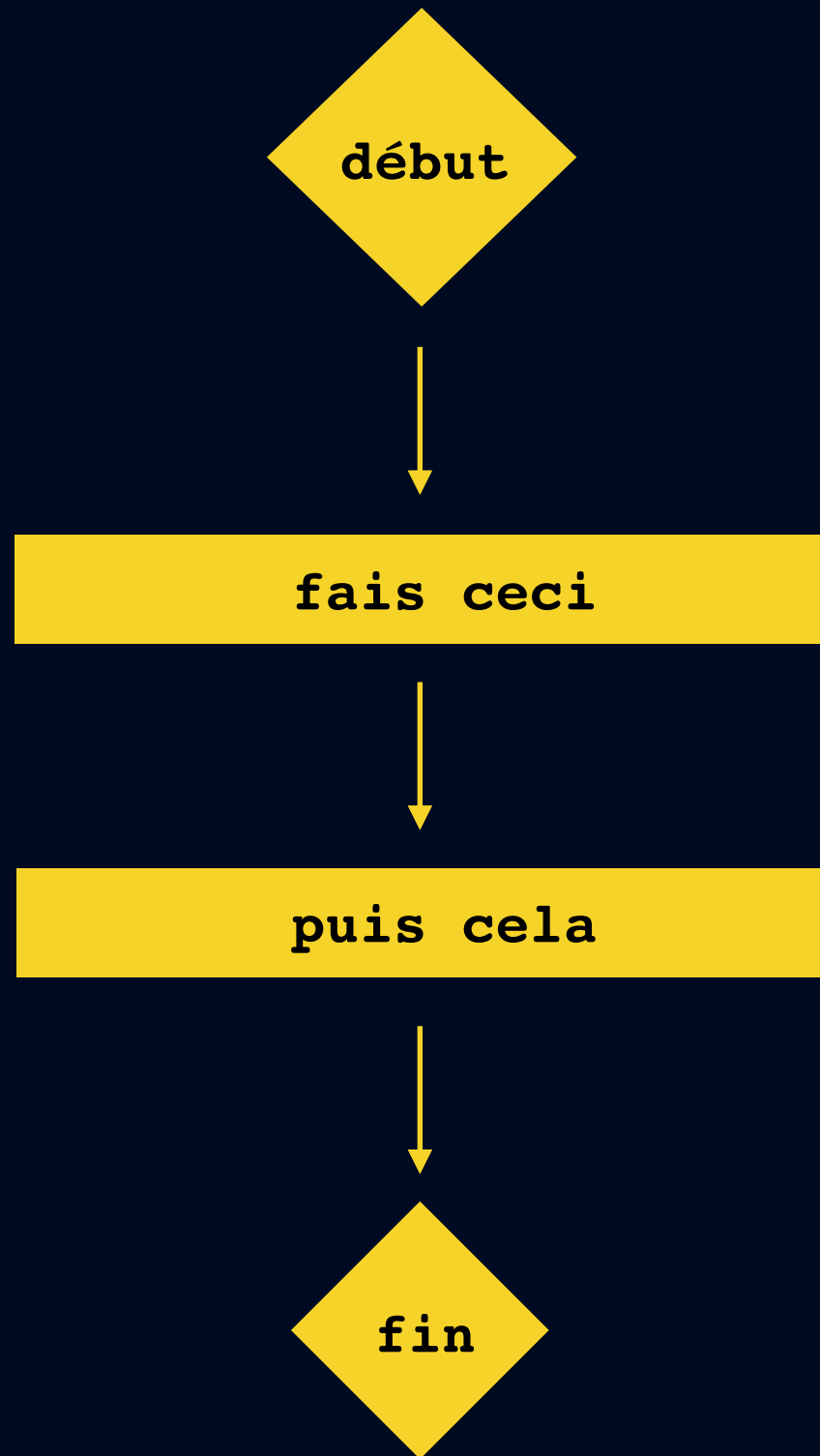
```
var $jeSuisCharlie = ?;
```

```
var $booléen_est_manichéen = true;
```

Les booléens sont très pratiques pour  
vérifier des **conditions**.

Et les conditions, c'est ce qui fait  
l'intérêt de la programmation.

## SANS CONDITION PAS D'OPTION



## AVEC DES CONDITIONS, ON PEUT FAIRE DES PROGRAMMES PLUS "INTELLIGENTS"

