

**UFR ANTHROPOLOGIE  
SOCIOLOGIE  
SCIENCE POLITIQUE**


**INSTITUT  
DE LA  
COMMUNICATION**

# Programmation Web


L3 MIASHS  
Année 2018-2019  
Jérôme Darmont

<http://eric.univ-lyon2.fr/~jdarmon/>

The Beginning

## Objectifs du cours (1/2)

- Création de **pages web dynamiques**
  - Générées à la demande
  - À l'aide d'un **langage de script**



Visiteur Client      1 : demande de la page      Serveur

2 : génération de la page

3 : envoi de la page

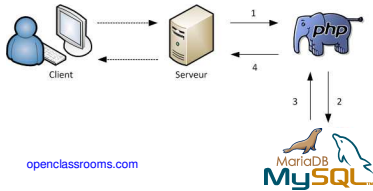
[openclassrooms.com](http://openclassrooms.com)

2      Programmation web      <http://eric.univ-lyon2.fr/~jdarmon/>

The Beginning

## Objectifs du cours (2/2)

- Interfaçage de pages web avec des **bases de données**



Client      Serveur      php      MySQL

1      2      3      4

[openclassrooms.com](http://openclassrooms.com)


3      Programmation web      <http://eric.univ-lyon2.fr/~jdarmon/>

The Beginning

## Actualité du cours

-  [http://eric.univ-lyon2.fr/~jdarmon/?page\\_id=445](http://eric.univ-lyon2.fr/~jdarmon/?page_id=445)
-  <http://eric.univ-lyon2.fr/~jdarmon/?feed=rss2>
-  [https://twitter.com/darmont\\_lyon2](https://twitter.com/darmont_lyon2) #webmiashs


4      Programmation web      <http://eric.univ-lyon2.fr/~jdarmon/>



## Plan du cours

- Langage de script PHP
  - **Bases de PHP**
  - PHP objet
  - Connexion à une base de données
  - Transmission de variables d'une page à une autre
  - Architecture Modèle-Vue-Contrôleur
  - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

5      Programmation web      <http://eric.univ-lyon2.fr/~jdarmon/>




Rasmus Lerdorf

## Généralités (1/3)

- **PHP** : *PHP Hypertext Preprocessor*
- Langage de script permettant d'insérer du code dans des pages web ⇒ pages Web dynamiques
- **Versions**
  - 1995 : PHP 1 – Rasmus Lerdorf (*Personal Home Page*)
  - 1997 : PHP/FI 2
  - 1998 : PHP 3 – Andi Gutmans & Zeev Suraski
  - 2000 : PHP 4 – Zend Engine
  - 2004 : PHP 5 – PHP devient orienté objet
  - 2015 : PHP 7 – Dmitry Stogov, Xinchun Hui & Nikita Popov


6      Programmation web      <http://eric.univ-lyon2.fr/~jdarmon/>



## Généralités (2/3)

- Syntaxe du langage PHP proche de celles de C, Perl ou Java
- Support d'un grand nombre de bases de données
- Support de services internet (IMAP, POP3, NNTP, HTTP)
- Gratuit, fonctionne sous Unix, Windows, Mac OS

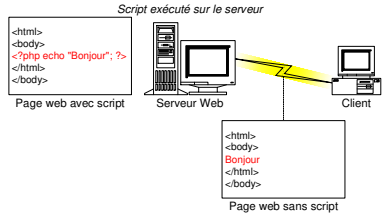
7 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Généralités (3/3)

- **PHP** : Langage de script "serveur" comme ASP (≠ Javascript, langage de script "client")

*Script exécuté sur le serveur*




8 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Syntaxe de base

- **Passage du HTML à PHP** (échappement) : `<?php ?>`
  - Ex. `<?php echo "Bonjour"; ?>`
- **Séparateurs d'instructions** : `;`
- **Commentaires**
  - Ex. `// Toute une ligne (façon C++)`
  - `# Toute une ligne (façon Shell Unix)`
  - `/* Plusieurs lignes (façon C) */`

9 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Variables et types (1/3)

- **Variables** : préfixées par le caractère `$`
- PHP ne nécessite **pas** de déclaration explicite du type de variable (☺).
- **Types de données** :
 

- Nombres entiers : int, integer	Ex. d'affectation
- Nombres réels : real, double, float	<code>\$i = 1;</code>
- Chaînes de caractères : string	<code>\$pi = 3.14;</code>
	<code>\$ch = "oui";</code>
- **Conversion de type** : "cast" comme en C
  - Ex. `$pi = (int) $pi; // $pi est égal à 3`


10 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Variables et types (2/3)

- **Tableaux** : scalaires ou associatifs, création par assignation des valeurs
  - Ex. `$ts[0] = "Chaîne 0"; // Indigage à partir de 0`
  - `$ts[1] = "Chaîne 1";`
  - `$ta["Dupont"] = 30;`
- **Fonctions associées** :
  - Initialisation :
    - Ex. `$notes_s = array(10, 12.5, 15, 8);`
    - `$notes_a = array("Valerii" => 16, "Vadim" => 12);`
  - Nombre d'éléments :
    - Ex. `$n = count($notes_s)`

11 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Variables et types (3/3)

- **Tableaux à plusieurs dimensions** : possibilité de mélanger indices scalaires et associatifs
  - Ex. `$matrice[0][0] = 2;`
  - `$msa["Dupont"][0] = 30;`

```
$matrice2 = array(
    array(1, 0, 0),
    array(0, 1, 0),
    array(0, 0, 1));
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

12 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Constantes

- **Variables d'environnement**
  - Ex. 

```
$ _SERVER["PHP_SELF"]
$_SERVER["SERVER_NAME"]
$_SERVER["HTTP_REFERER"]
$_SERVER["REMOTE_ADDR"]
```
- **Constantes définies par l'utilisateur**
  - Ex. 

```
define("MA_CHAINE", "Valeur de MA_CHAINE");
define("PI", 3.14159265);
echo "<p>MA_CHAINE = " . MA_CHAINE . "<br />";
echo "PI = " . PI . "</p>";
```

13 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Opérateurs (1/3)

- **Opérateurs d'affectation**

- Affectation simple :	\$a = 2;
- Affectation multiple :	\$a = \$b = 2;
- Affectation + opération :	\$a += 2;
- Pré/post incrémentation/décrémentation :	
++\$a;	--\$a;
\$a++;	\$a--;
- Affectation conditionnelle :	\$max = (\$a > \$b) ? \$a : \$b;
- **Opérateurs arithmétiques** : + - \* / % (modulo)

14 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Opérateurs (2/3)

- **Concaténation de chaînes de caractères** : .
  - Ex. 


```
$ch1 = $ch2 . $ch3;
$ch1 .= "<br />";
```
- **Caractères spéciaux dans les chaînes (échappement)**
  - Antislash : 

```
\\
```
  - Dollar : 

```
\$
```
  - Guillemets : 

```
\"
```

15 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Opérateurs (3/3)

- **Opérateurs logiques**
  - ET : and ou &&
  - OU : or ou ||
  - OU exclusif : xor
  - NON : !
- **Opérateurs de comparaison**

- Égalité/Différence :	==	!=
	=== pour les booléens	
- Inférieur/Supérieur :	<	>
- Inférieur ou égal/Supérieur ou égal :	<=	>=

16 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Affichage

- **Procédures prédéfinies**
  - Ex. 

```
echo "Bonjour !";
```
  - Ex. 


```
print("Bonjour !");
```
  - Ces deux procédures sont le **seul moyen** d'afficher quelque chose dans le document HTML final.
  - **Affichage de tableau** : 

```
print_r($my_array); // pour débogage
```
- **Formatage** : Utilisation des éléments HTML
  - Ex. 

```
echo "<h1>TITRE</h1>";
```
- **Saut de ligne dans le code source**
  - Ex. 

```
echo "<h1>TITRE</h1>\n";
```

17 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Affichage du contenu de variables

- **Utilisation de l'opérateur de concaténation**
  - Ex. 

```
echo "PI = " . $pi . "<br />";
```
- **Directement dans une chaîne**
  - Ex. 

```
echo "PI = $pi<br />";
```
- **Résultat de fonction** : concaténation obligatoire
  - Ex. 

```
echo "Carré de PI = " . carre($pi);
```

18 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Structures de contrôle (1/7)

- **Test :** `if (expression) instructions`  
`[elseif (expression) instructions]`  
`[else instructions]`

- Ex. `if ($a>$b) echo "A > B";`

`if ($a>$b) { // Plusieurs instructions`  
`echo "A > B";`  
`$b = $a;`  
`}`

19

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (2/7)

```
if ($a>$b) {
    echo "A > B";
} else {
    echo "A <= B";
}
```

```
if ($a>$b) echo "A > B";
elseif ($a<$b) echo "A < B";
else echo "A = B";
```

```
if (un_booleen === true) echo "Vrai !";
```

20

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (3/7)

- **Sélection :** `switch(variable) {cas}`

- Ex. `switch($i) {`  
`case 0: echo "i=0"; break;`  
`case 1: echo "i=1"; break;`  
`case 2: echo "i=2"; break;`  
`}`

`switch($ch) {`  
`case "a": echo "A"; $ch="A"; break;`  
`case "b": echo "B"; $ch="B"; break;`  
`case "c": echo "C"; $ch="C"; break;`  
`default: echo "Ni 'a' ni 'b' ni 'c'";`  
`}`

21

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (4/7)

- **Boucle "tant que" :** `while(expr) {instr}`  
 - Ex. `$i=1;`  
`while ($i<=10) { echo $i++;`  
`}`
- **Boucle "répéter tant que" :** `do {instr} while(expr)`  
 - Ex. `$i=1;`  
`do { echo $i++;`  
`} while ($i<=10);`
- **Boucle "pour" :** `for (expr1; expr2; expr3) {instr}`  
 - Ex. `for ($i=1; $i<=10; $i++) { echo $i;`  
`}`

22

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (5/7)

- **Boucle "pour tout élément" de tableau scalaire**  
`foreach (tableau as valeur) instructions`

- Ex. `$tab[0]="Rouge";`  
`$tab[1]="Vert";`  
`$tab[2]="Bleu";`  
`foreach ($tab as $val) {`  
`echo "Valeur courante : $val<br />";`  
`}`

23

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (6/7)


- **Boucle "pour tout élément" de tableau associatif**  
`foreach (tableau as clé => valeur) instructions`

- Ex. `$tab["Rouge"]="#FF0000";`  
`$tab["Vert"]="#00FF00";`  
`$tab["Bleu"]="#0000FF";`  
`foreach ($tab as $cle => $val) {`  
`echo "Code de $cle : $val<br />";`  
`}`

24

Programmation web


<http://eric.univ-lyon2.fr/~jdarmont/>



## Structures de contrôle (7/7)

- **Inclusion de fichiers dans une page**
  - Fonction `require()` : Provoque une erreur fatale si le fichier requis manque (interruption du script)
  - Fonction `include()` : Provoque seulement un avertissement (*warning*) si le fichier requis manque
  - Évaluation des fichiers inclus en mode HTML
- **Exemples** (paramètre des fonctions : une URL)
  - `require("mes_fonctions.inc.php");`
  - `include("une_page_web.html");`
  - `include("http://serveur.fr/pg.html");`

25 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>




## Fonctions (1/2)

- **Squelette de définition de fonction**

```
function nomFn($p1, $p2, ..., $pn) {
    // Code de la fonction
    return val_retour; // Optionnel
}
```
- **Exemple**

```
function monMax($n1, $n2) {
    return ($n1 > $n2) ? $n1 : $n2;
}
```
- Une fonction doit être définie avant d'être appelée.

26 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>




## Fonctions (2/2)

- **Passage de paramètres par référence** (les paramètres sont passés par valeur par défaut) : `&`
  - Ex.
 

```
function aLaLigne(&$chaine) {
    $chaine .= "<br />";
}
$ch="Coucou !";
aLaLigne($ch); // <br /> ajouté à la fin
```
- **Paramètres par défaut**
  - Ex.
 

```
function cafe($type="expresso") {
    return "Je fais un $type<br />";
}
echo cafe(); // Je fais un expresso
echo cafe("capucino"); // Je fais un capucino
```

27 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>




## Visibilité des variables

- **Variables normales** : visibles uniquement à l'intérieur de la fonction où elles sont définies
- **Variables globales**
  - Ex.
 

```
function exempleGlobal {
    global $vglob; // Visible hors de la fonction
}
```
- **Variables statiques**
  - Ex.
 

```
function exempleStatic {
    static $vstat; // Conserve sa valeur entre deux appels à la fonction
}
```


28 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>



## Plan du cours

- Langage de script PHP
  - ✓ Bases de PHP
  - **PHP objet**
  - Connexion à une base de données
  - Transmission de variables d'une page à une autre
  - Architecture Modèle-Vue-Contrôleur
  - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

29 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>

 Programmation Orientée Objet

## Classes, attributs, méthodes

```
<?php
class Exemple {
    // Attributs
    public $commentaire;
    private $nom;
    protected $note;

    // Méthodes
    public function afficherNom() {
        echo $this->nom;
    }
    public function changerNote($nouvelleNote) {
        $this->note = $nouvelleNote;
    }
}
```

30 Programmation web <http://eric.univ-lyon2.fr/~jdarmont/>

## Instanciation

```
<?php
// Création d'objet
$un_exemple = new Exemple();

// Appel de méthodes
$un_exemple->commentaire = "Pas mal pour un début";

$un_exemple->afficherNom();

$un_exemple->changerNote(20);
?>
```

31

## Constructeur

```
<?php
class Exemple { // suite

    // Constructeur
    function __construct($nom) {
        $this->commentaire = "No comment";
        $this->nom = $nom;
        $this->note = 0;
    }
}
?>
```

32

## Héritage et surcharge

```
<?php
class SousExemple extends Exemple {
    // Attribut spécifique
    private $appreciation;
    // Constructeur
    function __construct($nom, $appr) {
        parent::__construct($nom);
        $this->appreciation = $appr;
    }
    // Méthode surchargée
    public function changerNote($note, $appr) {
        parent::changerNote($note);
        $this->appreciation = $appr;
    }
}
?>
```

33



## PHP Standards Recommendations (PSR)

- Recommandations pour améliorer l'interopérabilité des applications PHP
- Tendent à devenir des standards
- Parmi celles qui sont validées :
  - PSR-1 : Basic Coding Standard
  - PSR-2 : Coding Style Guide

34



## L'essentiel de PSR-1

Balises PHP	<?php ?> ou <?= ?>
Encodage des caractères	UTF-8 without BOM
Nom de classes	En StudyCaps
Constantes	En majuscules
Noms de fonctions/méthodes	En camelCase()


35



## L'essentiel de PSR-2 (1/2)

Code PHP	En PSR-1
Sauts de ligne	Unix
Longueur des lignes de code	80 caractères maxi de préférence
Instructions	Une seule par ligne
Indentation	4 espaces (pas de tabulation)
Blocs de code	Retour à la ligne après { } seule sur une ligne


36



## L'essentiel de PSR-2 (2/2)

Visibilité des attributs/méthodes	Doit être déclarée
Instructions/constantes PHP	En minuscules
Paramètres de fonctions/méthodes	Pas d'espace avant la virgule Une espace après la virgule
Appel de fonctions/méthodes	Pas d'espace avant les ( )
Structures de contrôle	Une espace après l'instruction Entre les ( ), pas d'espace après ( ni avant ), mais une espace après )


37 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Plan du cours

- Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - Connexion à une base de données
  - Transmission de variables d'une page à une autre
  - Architecture Modèle-Vue-Contrôleur
  - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3


38 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Principe

- Imbrication de requêtes SQL dans du code PHP + formulaires HTML pour les mises à jour
- SGBD utilisé : **MariaDB**
  - SGBD relationnel multi-utilisateurs rapide et open-source
  - 1995 : MySQL AB (v1-v5.0)
  - 2008 : rachat de MySQL AB par Sun Microsystems (v5.1)
  - 2010 : rachat de Sun Microsystems par Oracle (v5.5-v8.0)
  - 2010 : MariaDB *fork* libre de MySQL (v5.2-v10.3)
  - Désormais utilisé dans de nombreuses distributions Linux

39 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Éléments de langage SQL (1/5)


- Base de données** : ensemble de tables
- Table** : ensemble d'attributs et leurs valeurs
  - Ex. ETUDIANT (numetu, nom, prenom, datenaiss, note)
- Interrogation simple**

```
SELECT liste_attributs FROM table
WHERE condition
```

  - Ex. SELECT nom, prenom FROM etudiant WHERE note >= 10
- Création de table**

```
CREATE TABLE nom_table (liste(champ, type))
```


40 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Éléments de langage SQL (2/5)

- Types de données principaux**
  - Nombre entier : INT
  - Nombre réel : FLOAT
  - Chaîne de caractères : VARCHAR(*taille*)
  - Date : DATE
- Clé primaire** : identifie de façon unique les n-uplets (lignes) de la table
  - Mot-clé PRIMARY KEY après le type

41 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Éléments de langage SQL (3/5)

Exemple de création de table

```
CREATE TABLE etudiant (
  numetu INT PRIMARY KEY,
  nom VARCHAR(50),
  prenom VARCHAR(100),
  datenaiss DATE,
  note FLOAT )
```

42 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Éléments de langage SQL (4/5)

- **Insertion d'un n-uplet**  
`INSERT INTO nom_table VALUES (liste_valeurs)`  
 - Ex. `INSERT INTO etudiant VALUES (123, 'Darmont', 'Jérôme', '15-01-1972', 15.5)`
- **Suppression d'un n-uplet**  
`DELETE FROM nom_table WHERE condition`  
 - Ex. `DELETE FROM etudiant WHERE numetu = 123`
- **Modification d'un n-uplet**  
`UPDATE nom_table SET attribut = valeur WHERE condition`  
 - Ex. `UPDATE etudiant SET note = 20 WHERE numetu = 123`

43

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Éléments de langage SQL (5/5)

- **Requête de jointure (multi-table)**  
`SELECT liste_attributs`  
`FROM table1, table2, ... tableN`  
`WHERE condition_jointure_t1t2`  
`...`  
`AND condition_jointure_tN-1tN`  
 - Ex. `SELECT *`  
`FROM etudiant, diplome`  
`WHERE etudiant.num_dipl = diplome.num_dipl`

44

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## PHP Data Objects (PDO)

- **Accès unifié à différents SGBD**  
 - Remplace les API précédentes (ex. API PHP-MySQL), **obsolètes et vouées à disparaître**
- **Préparation des requêtes**  
 - Performance lors d'exécutions multiples  
 - Sécurité (notamment contre les injections de code)
- **Gestion des exceptions (erreurs)**

45

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Connexion à une BD

- **Connexion**  
`$idconn = new PDO(id_serveur_bd, login_mysql, mot_de_passe);`  
 - `id_serveur = pilote:host=serveur;dbname=nom_bd`  
 - Ex. `$c = new PDO("mysql:host=localhost;dbname=darmont", "darmont", "x");`
- **Gestion des erreurs**  
`try { instructions } catch () { traitement_des_erreurs }`  
 - Ex. `try {`  
`// Connexion`  
`} catch (PDOException $erreur) {`  
`echo "<p>Erreur : " . $erreur->getMessage() . "</p>\n";`  
`}`

46

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Exécution d'une requête

- **Préparation de la requête**  
`$res = $idconn->prepare(requete_SQL)`  
 - Ex. `$res = $c->prepare("select nom, prenom from etudiant");`
- **Exécution de la requête**  
 - Ex. `$res->execute();`
- **Accès au résultat de la requête (ligne par ligne)**  
 - Ex. `foreach($res as $ligne)`  
`echo "<p>". $ligne["nom"] . " " . $ligne["prenom"] . "</p>\n";`

47

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Résultat intégral

- **Chargement complet du résultat de la requête en mémoire**  
 - Rapide, utile quand on utilise des gabarits (*templates*)  
 - À éviter lorsque le résultat est volumineux
- **Exemple**  
`$res = $c->prepare("select nom, prenom from etudiant");`  
`$res->execute();`  
`$data = $res->fetchAll(PDO::FETCH_ASSOC);`  
`foreach($data as $ligne)`  
`echo "<p>". $ligne["nom"] . " " . $ligne["prenom"] . "</p>\n";`

48

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



php  
MariaDB

## Requêtes paramétrées

```
// Exemple avec paramètres anonymes
$res = $c->prepare("select nom, prenom from etudiant
where datenaiss > ? and note >= ?");

$res->execute(["2018-01-01", 10]);
// Les paramètres doivent être passés dans l'ordre
```

49 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

php  
MariaDB

## Requêtes paramétrées multiples

```
// Exemple avec paramètres nommés
$res = $c->prepare("insert into ETUDIANT
values (:id, :nom, :prenom, NULL, NULL)");
// Liaison des paramètres
$res->bindParam(":id", $id);
$res->bindParam(":nom", $nom);
$res->bindParam(":prenom", $prenom);
// 1re insertion
$id = 800; $nom = "Bentayeb"; $prenom = "Fadila";
$res->execute();
// 2e insertion
$id = 810; $nom = "Harbi"; $prenom = "Nouria";
$res->execute();
```

50 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

php  
MariaDB

## Exemple complet

```
try {
    // Connexion
    $c = new PDO("mysql:host=localhost;dbname=darmont", "darmont", "x");
    $c->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    // Requête d'interrogation
    $res->prepare("select nom, prenom from etudiant");
    $res->execute();
    echo "<ul>\n";
    foreach($res as $l)
        echo "<li>". $l["nom"] . " " . $l["prenom"] . "</li>\n";
    echo "</ul>\n";
    echo "<p>". $res->rowCount() . " résultat(s)</p>\n";
    // NB : rowCount() fonctionne aussi avec les requêtes de mise à jour
} catch (PDOException $erreur) { // Gestion des erreurs
    echo "<p>Erreur : ". $erreur->getMessage() . "</p>\n";
}
```

51 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

php  
MariaDB

## Conversion de jeux de caractères

- Encodage de caractères Europe de l'Ouest : **iso-8859-1** (sur 2 octets)
- Encodage de caractères international : **utf-8** (sur 4 octets, plus récent)
- Si un texte contenu dans une BD est à un format et que le serveur Web utilise l'autre : problème (accents...)
  - Affichage de type **Ã** : caractères utf-8 affichés en iso-8859-1
    - ⇒ `$idconn->exec("set names latin1");`
  - Affichage de type **?** : caractères iso-8859-1 affichés en utf-8
    - ⇒ `$idconn->exec("set names utf8");`

Une seule fois, après la commande de connexion

52 Programmation Web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Formulaires HTML (1/9)

- **Formulaires** : permettent la saisie de données dans une page Web
- **Définition** : `<form> </form>`
  - Attribut **action** : URL de la page PHP à exécuter après validation du formulaire
  - Attribut **method** : méthode de transmission des données (valeurs possibles : **get** et **post**)
  - Attribut **enctype** : type d'encodage (par défaut **application/x-www-form-urlencoded** ou **multipart/form-data** pour envoyer des fichiers)

53 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Formulaires HTML (2/9)

- **Différence entre les méthodes get et post**
  - **get** : apparition des valeurs saisies en paramètres de l'URL de la page action
  - **post** : . valeurs saisies cachées  
. quantité de données possible plus importante
- **Exemple**  
`<form action="ajout_etu.php" method="post">...</form>`
- **Structure d'un formulaire** : ensemble de zones de saisie (groupes de champs)
- **Groupe de champs** : `<fieldset> </fieldset>`

54 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (3/9)

- **Saisie dans un formulaire** : `<input />`
  - Attribut obligatoire : `name`, nom de la variable
- **Champ texte** : `<input type="text" size="" />`
  - Ex. `<input type="text" name="nom" size="30" />`
- **Suggestion de valeurs** : `<datalist>...</datalist>`
  - Ex.
 

```
<datalist id="prop_noms">
  <option value="Dupond">
  <option value="Durand">
  <option value="Martin">
</datalist>
<input type="text" name="nom" list="prop_noms" />
```

55

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (4/9)

- **Vérification syntaxique** : expressions régulières
  - Ex. `<input type="email" pattern="[^\s@]*@[^\s@]*" />`
- **Champ mot de passe** : `<input type="password" />`
  - Ex. `<input type="password" name="passwd" size="8" />`
- **Champ caché** : `<input type="hidden" value="" />`
  - Ex. `<input type="hidden" name="numetu" value="10" />`
- **Fichier** : `<input type="file" />`
  - Ex. `<input type="file" name="Fichier_téléchargé" />`

56

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (5/9)

- **Bouton radio** : `<input type="radio" value="" />`
  - Ex.
 

```
Homme : <input type="radio" name="genre" value="H" />
Femme : <input type="radio" name="genre" value="F" />
```
- **Case à cocher** : `<input type="checkbox" />`
  - Ex.
 

```
choix 1 : <input type="checkbox" name="choix1" />
choix 2 : <input type="checkbox" name="choix2" />
```

57

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (6/9)

- **Boutons de commande**
  - Attribut `type = submit | reset` : validation ou réinitialisation du formulaire
  - Attribut `value` : légende du bouton
  - Ex. `<input type="submit" name="Valider" value="Valider" />`  
`<input type="reset" name="Annuler" value="Annuler" />`
- **Zone de texte long** : `<textarea> </textarea>`
  - Attribut `name` : nom de la zone de texte
  - Attributs `rows` et `cols` : nombre de lignes / colonnes
  - Ex. `<textarea name="texte" rows="10" cols="60">`  
`</textarea>`

58

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (7/9)

- **Liste déroulante** : `<select> </select>`
  - Attribut `name` : nom de la variable choix
  - Élément `<option> </option>` : objet de la liste
  - Attribut `selected` de `<option>` : choix par défaut
  - Ex.
 

```
<select name="annee">
  <option>Licence</option>
  <option selected="selected">M1</option>
  <option>M2 professionnel</option>
  <option>M2 recherche</option>
</select>
```

59

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (8/9)

- Options d'accessibilité (navigation alternative)
- **Description de champ** : `<label> </label>`
  - Ex.
 

```
<label for="id_nom">Nom</label>
<input type="text" id="id_nom" name="nom" />
```
- **Légende de zone de saisie** : `<legend> </legend>`
  - Ex. `<fieldset>`  
`<legend>État civil de l'étudiant</legend>`  
`...`  
`</fieldset>`

60

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Formulaires HTML (9/9)

```
<!-- Exemple complet -->
<form action="ajout_etu.php" method="post"> <fieldset>
<p> Nom : <input type="text" name="nom" size="50" /> </p>
<p> Âge : <input type="text" name="age" size="2" /> </p>
<p> Année : <select name="annee">
  <option>Licence</option>
  <option selected="selected">M1</option>
  <option>M2</option>
</select> </p>
<input type="hidden" name="action" value="ajout" />
<p> <input type="reset" value="Annuler" />
  <input type="submit" value="Valider" /> </p>
</fieldset> </form>
```

61

Programmation web

http://eric.univ-lyon2.fr/~jdamont/

## Exploitation des données d'un formulaire

- Dans la page cible (Ex. ajout\_etu.php)
  - Tableaux associatifs `$_GET[]` et `$_POST[]`
  - Permettent d'accéder aux valeurs transmises par les méthodes `get` et `post` des formulaires, respectivement

- Exemple

```
<?php
echo "<p>L'étudiant " . $_POST["nom"];
echo " (" . $_POST["age"] . " ans)";
echo " est en " . $_POST["annee"] . ". "</p>";
?>
```

- Cas particuliers : case à cocher (valeur "on" si cochée) et fichiers

62

Programmation web

http://eric.univ-lyon2.fr/~jdamont/

## Mise à jour d'une base de données

```
<?php // Suite de l'exemple : document ajout_etu.php
$c = new PDO("mysql:host=localhost;dbname=darmont",
  "darmont", "x");
$resultat = $c->prepare("insert into etudiant values (?, ?, ?)");
$resultat->execute([ $_POST["nom"],
  $_POST["age"],
  $_POST["annee"]  ]);
if ($resultat->rowCount() > 0)
  echo "<p>Etudiant.e ajouté.e</p>\n";
?>
```

63

Programmation web

http://eric.univ-lyon2.fr/~jdamont/

## Téléchargement de fichier (1/2)

- Étape 1 : formulaire dans une page HTML

```
- Ex.
<form action="telechargement.php" method="post"
  enctype="multipart/form-data" />
<fieldset>
  <input type="hidden" name="MAX_FILE_SIZE" value="50000" />
  Fichier : <input name="monfichier" type="file" />
  <input type="submit" value="Télécharger" />
</fieldset>
</form>
```

- Étape 2 : traitement à l'aide de PHP

64

Programmation web

http://eric.univ-lyon2.fr/~jdamont/

## Téléchargement de fichier (2/2)

- Variables disponibles dans la page cible
  - `$_FILES["monfichier"]["name"]` : nom original du fichier
  - `$_FILES["monfichier"]["type"]` : type du fichier
  - `$_FILES["monfichier"]["size"]` : taille du fichier
  - `$_FILES["monfichier"]["tmp_name"]` : nom temporaire du fichier sur la machine serveur
- Exemple de code dans la page telechargement.php
 

```
$destination = "/home/jd/public_html/" . $_FILES["monfichier"]["name"];
$res = move_uploaded_file($_FILES["monfichier"]["tmp_name"],
  $destination);
if ($res) echo "<p>Fichier téléchargé avec succès</p>";
else echo "<p>Erreur : " . $_FILES["monfichier"]["error"] . "</p>";
```

65

Programmation web

http://eric.univ-lyon2.fr/~jdamont/




## Plan du cours

- Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - Transmission de variables d'une page à une autre
  - Architecture Modèle-Vue-Contrôleur
  - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

66

Programmation web


http://eric.univ-lyon2.fr/~jdamont/



## Méthodes POST et GET (1/3)

- **Exemple** : On dispose du nom et de l'âge du visiteur de la page `courante.php` dans les variables `$nom` et `$age`. On souhaite les transmettre à la page `suivante.php`.
- **Méthode 1** : Utiliser les champs cachés d'un formulaire
- **Méthode 2** : Ajouter des paramètres à l'URL de la page cible

67 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Méthodes POST et GET (2/3)

```
<!-- courante.php -->

<!-- Méthode 1 -->
<form action="suivante.php" method="post"> <fieldset>
<input type="hidden" name="nom" value="<?php echo $nom; ?>" />
<input type="hidden" name="age" value="<?php echo $age; ?>" />
<input type="submit" value="Envoyer" />
</fieldset> </form>

<!-- Méthode 2 -->
<a href="suivante.php?nom=<?php echo $nom; ?>&age=
<?php echo $age; ?>">Envoyer</a>
```

68 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Méthodes POST et GET (3/3)


```
<?php // suivante.php

// Méthode 1
$nom = $_POST["nom"];
$age = $_POST["age"];

// Méthode 2
$nom = $_GET["nom"];
$age = $_GET["age"];

?>
```

69 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Sessions (1/6)

- **Objectif** : Stockage de variables lors de la navigation sur plusieurs pages Web successives
- **Utilisations courantes** :
  - Identification des visiteurs d'un site par login et mot de passe stockés dans une base de données
  - Gestion du profil des utilisateurs
  - ...

70 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Sessions (2/6)

```
<!-- Exemple -->
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="fr">
<head> <title>Session 1</title> </head>
<body>
<form action="session2.php" method="post">
<fieldset>
Votre nom : <input type="text" name="nom" />
<input type="submit" value="Valider" />
</fieldset>
</form>
</body>
</html>
```

71 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Sessions (3/6)

```
<?php
session_start();
// Enregistre la valeur du nom dans la variable de session s_nom
$_SESSION["s_nom"] = $_POST["nom"];
?>
<!DOCTYPE html>
<html lang="fr">
<head> <title>Session 2</title> </head>
<body>
<p>Identifiant de session : <?php echo session_id(); ?><br />
Nom de session : <?php echo session_name(); ?></p>
<p>Bienvenue sur ma page Web, <?php echo $_SESSION["s_nom"]; ?>.<br />
<a href="session3.php">Page suivante</a>.</p>
</body>
</html>
```

72 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Sessions (4/6)

```

<?php session_start(); ?>
<!DOCTYPE html>
<html lang="fr">
<head> <title>Session 3</title> </head>
<body>
<p>Es-tu toujours là, <?php echo $_SESSION["s_nom"]; ?> ?<br />
<a href="session4.php">Détruire la variable de session</a> ou
<a href="session5.php">close la session</a> ?</p>
</body>
</html>

```

73 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Sessions (5/6)

```

<?php
session_start();
unset($_SESSION["s_nom"]); // Suppression de la variable de session
?>
<!DOCTYPE html>
<html lang="fr">
<head> <title>Session 4</title> </head>
<body>
<p>Es-tu toujours là ?
<?php
if ($_SESSION["s_nom"]) echo "Oui.";
else echo "Non.";
?>
<br /><a href="session5.php">Close la session</a>.</p>
</body>
</html>

```

74 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>




## Sessions (6/6)

```

<?php
session_destroy(); // Destruction de la session
?>
<!DOCTYPE html>
<html lang="fr">
<head> <title>Session 5</title> </head>
<body>
<p>Session terminée.</p>
<p>Es-tu toujours là ?
<?php
if ($_SESSION["s_nom"]) echo "Oui.";
else echo "Non.";
?>
</p>
</body>
</html>

```


75 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Plan du cours

- Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - ✓ Transmission de variables d'une page à une autre
    - Architecture **Modèle-Vue-Contrôleur**
    - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

76 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Principe de séparation du code

- Tâches différentes / langages différents
  - **HTML** : Structuration et contenu statique
  - **CSS** : Présentation
  - **PHP** : Contenu dynamique
  - **SQL** : Interrogation base de données
- Pourquoi séparer le code ?
  - Maintenance facilitée
    - Code plus lisible
    - La modification d'une tâche n'affecte pas les autres
  - Réutilisation de code

77 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Comment séparer le code

- Styles CSS dans des fichiers séparés
- Fonctions/méthodes PHP
  - `include()/require()` : par ex., définition dans des fichiers séparés d'en-têtes et de pieds de page HTML communs à plusieurs pages
  - PDO `prepare()` : isolation des requêtes SQL
- Gabarits (*templates*)
  - Moteur logiciel (*template processor/engine*) pour associer gabarits et données
- Architecture **Modèle-Vue-Contrôleur**
  - Trygve Reenskaug, 1978 

78 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Principe de MVC

The diagram illustrates the MVC principle. A user (represented by a person icon) sends a request: "Demande la page depuis son navigateur". This request goes to the Controller (Contrôleur), which performs calculations ("Calcul"). The Controller then interacts with the Model (Modèle), which is connected to a database ("Accès à la BD"). The Controller also sends data to the View (Vue), which displays the page ("Affichage") using HTML ("

79 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Modèle

- **Gestion des données**
  - Base de données ou fichiers
- Ensemble de fonctions ou de classes
  - Mises à jour (ajout/modification/suppression)
  - Interrogation
- **Langages**
  - PHP
  - SQL

80 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Vue

- **Présentation des résultats**
- Interaction avec l'utilisateur-trice
- **Langages**
  - HTML uniquement si utilisation de gabarits
  - Instructions PHP **simples** sinon

81 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Contrôleur

- **Analyse des requêtes de l'utilisateur-trice**
- Interrogation ou mise à jour du modèle
- Modification de la vue
- **Langage**
  - PHP

82 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## MVC ≠ 3-tiers

- **Architecture 3-tiers**
  - Couches
- **Architecture MVC**
  - La vue peut consulter le modèle sans passer par le contrôleur (lecture uniquement)

83 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Cas pratique (1/2)

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>Exemple d'application non-MVC</title>
  <link rel="stylesheet" href="tableau.css" />
</head>
<body>
  <table>
    <caption>Liste des matières</caption>
    <tr>
      <th>codemat</th>
      <th>libelle</th>
      <th>coef</th>
    </tr>
```

84 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Cas pratique (2/2)

```
<?php
require("connect.inc.php");
$c = new PDO("mysql:host=$host;dbname=$dbname", $login, $password);
$res = $c->prepare("SELECT * FROM matiere");
$res->execute();
foreach ($res as $l)
    echo "<tr> <td>" . $l["codemat"] . "</td> <td>" . $l["libelle"]
        . "</td> <td>" . $l["coef"] . "</td> </tr>\n";
?>
```

</table>  
</body>  
</html>

85

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Découpage : modèle

```
<?php // modele-o.inc.php
class Modele {
    private $idc;
    private function connexion() {
        require("connect.inc.php");
        $this->idc = new PDO("mysql:host=$host;
            dbname=$dbname", $login, $password);
    }
    public function lireMatiere() {
        $this->connexion();
        $res = $this->idc->prepare("SELECT * FROM matiere");
        $res->execute();
        return $res;
    }
}
```

86

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Découpage : Vue (1/2)

```
<!DOCTYPE html>
<!-- vue.inc.php -->
<html lang="fr">
<head>
    <meta charset="utf-8" />
    <title>Exemple d'application MVC</title>
    <link rel="stylesheet" href="tableau.css" />
</head>
```

87

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Découpage : Vue (2/2)

```
<body>
    <table>
        <caption>Liste des matières</caption>
        <tr> <th>codemat</th> <th>libelle</th> <th>coef</th> </tr>
        <?php foreach ($res as $l) { ?>
            <tr>
                <td><?php echo $l["codemat"] ?></td>
                <td><?php echo $l["libelle"] ?></td>
                <td><?php echo $l["coef"] ?></td>
            </tr>
        <?php } ?>
    </table>
</body>
</html>
```

88

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Découpage : contrôleur

```
<?php // controleur-o.php
require("modele-o.inc.php");
$mod = new Modele();

$res = $mod->lireMatiere();

require("vue.inc.php");
?>
```

89

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Plan du cours

- Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - ✓ Transmission de variables d'une page à une autre
  - ✓ Architecture Modèle-Vue-Contrôleur
    - Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

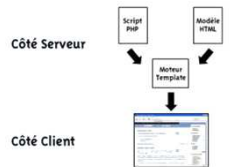
90

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Principe

- Objectifs :
  - Encore séparation du code PHP et HTML
  - Partage de gabarits entre plusieurs applications web



<https://phpcodeur.net/articles/php/templates>

91

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Exemple de gabarit

```
<table> <!-- gab1.tpl.html -->
<caption>Variables simples</caption>
<tr>
  <th>Nom</th>
  <th>Prénom</th>
</tr>
<tr>
  <td>{nom}</td>
  <td>{prenom}</td>
</tr>
</table>
```

92

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Affectation de variables à un gabarit

```
// Bibliothèque template simple et libre du phpBB Group
require("template.class.php");

// Lecture des données
$res = $c->prepare("select nom, prenom from etudiant where numetu = ?");
$res->execute([110]);
$ligne = $res->fetch();

// Mise en œuvre du gabarit
$gab1 = new Template("."); // Paramètre : chemin d'accès aux gabarits
$gab1->set_filenames(array("body1" => "gab1.tpl.html")); // Nom du fichier gabarit
// Assignation des variables
$gab1->assign_vars(array("nom" => $ligne["nom"],
                        "prenom" => $ligne["prenom"]));
$gab1->pparse("body1"); // Affichage des données
```

93

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Gabarit avec bloc de répétition

```
<table> <!-- gab2.tpl.html -->
<caption>Bloc de répétition</caption>
<tr>
  <th>Nom</th>
  <th>Prénom</th>
</tr>
<!-- BEGIN etu -->
<tr>
  <td>{etu.nom}</td>
  <td>{etu.prenom}</td>
</tr>
<!-- END etu -->
</table>
```

94

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Mise en œuvre d'un bloc de répétition

```
require("template.class.php");

// Lecture des données
$res = $c->prepare("select nom, prenom from etudiant");
$res->execute();

// Mise en œuvre du gabarit
$gab2 = new Template(".");
$gab2->set_filenames(array("body2" => "gab2.tpl.html"));
foreach($res as $ligne)
  $gab2->assign_block_vars("etu",
    array("nom" => $ligne["nom"],
          "prenom" => $ligne["prenom"]));

$gab2->pparse("body2");
```

95

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Blocs imbriqués

```
<table> <!-- gab3.tpl.html -->
<caption>Blocs imbriqués</caption>
<!-- BEGIN etu -->
<tr>
  <!-- BEGIN attribut -->
  <td>{etu.attribut.valeur}</td>
  <!-- END attribut -->
</tr>
<!-- END etu -->
</table>
```

96

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Mise en œuvre de blocs imbriqués

```
require("template.class.php");
// Lecture des données
$res = $c->prepare("select * from etudiant");
$res->execute();
$data = $res->fetchAll(PDO::FETCH_ASSOC);
// Mise en œuvre du gabarit
$gab3 = new Template("./");
$gab3->set_filenames(array("body3" => "gab3.tpl.html"));
foreach($data as $ligne) {
    $gab3->assign_block_vars("etu", array("rien" => "")); // Boucle sur les <tr>
    foreach ($ligne as $val)
        $gab3->assign_block_vars("etu.attribut", array("valeur" => $val));
}
$gab3->pparse("body3");
```

97

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Gabarit dans l'application MVC exemple (1/4)

```
<?php // modele-og.inc.php
class Modele {
    private $idc;
    private function connexion() {
        // Comme dans mode-o.inc.php
    }
    public function lireMateries() {
        // Comme dans mode-o.inc.php
    }
    public function afficherMateries($data) {
        $gab = new Template("./");
        $gab->set_filenames(array("body" => "vue.tpl.html"));
        foreach($data as $ligne)
            $gab->assign_block_vars("matiere",
                array(
                    "codemat" => $ligne["codemat"],
                    "libelle" => $ligne["libelle"],
                    "coef" => $ligne["coef"]
                ));
        $gab->pparse("body");
    }
}
?>
```

98

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Gabarit dans l'application MVC exemple (2/4)

```
<!DOCTYPE html>
<html lang="fr" > <!-- vue.tpl.html (vue et gabarit) -->
<head>
    <meta charset="utf-8" />
    <title>Exemple d'application MVC</title>
    <link rel="stylesheet" href="tableau.css" />
</head>
<body>
    <table>
        <caption>Liste des matières</caption>
        <tr> <th>codemat</th> <th>libelle</th> <th>coef</th> </tr>
```

99

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Gabarit dans l'application MVC exemple (3/4)

```
<!-- BEGIN matiere -->
<tr>
    <td>{matiere.codemat}</td>
    <td>{matiere.libelle}</td>
    <td>{matiere.coef}</td>
</tr>
<!-- END matiere -->
</table>
</body>
</html>
```

100

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Gabarit dans l'application MVC exemple (4/4)

```
<?php // controleur-og.php
require("template.inc.php");
require("modele-og.inc.php");

$mod = new Modele();
$res = $mod->lireMateries();
$mod->afficherMateries($res);
?>
```

101

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

## Plan du cours

- ✓ Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - ✓ Transmission de variables d'une page à une autre
  - ✓ Architecture Modèle-Vue-Contrôleur
  - ✓ Gabarits
- Rappels
  - Langage HTML5
  - Feuilles de style CSS3

102


Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>

HTML

## Un peu d'histoire...

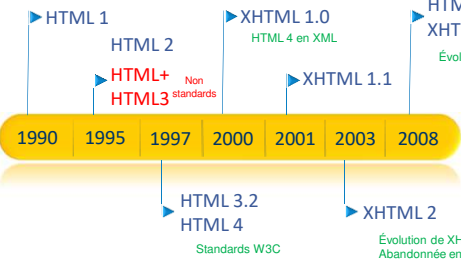
- **HTML** : *HyperText Markup Language*
  - Liens hypertextes
  - Langage à base de balises
- **Objectif** : publier sur le Web des documents formatés
- Développé par **Tim Berners-Lee** au CERN (Suisse)
- Standardisé par le **W3C**  
*World Wide Web Consortium*  
[www.w3.org](http://www.w3.org)



103 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Versions successives de HTML



104 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

W3C®

## Pourquoi des standards ? (1/2)

- Éviter la "balkanisation du Web" des années 1990
- Interopérabilité et portabilité (tablettes, téléphones mobiles...)
- Accessibilité aux personnes handicapées
- Réduction des coûts de développement
- Exploitation de la technologie XML (vers le **Web sémantique** / Web 3.0)
  - Exemple : moteurs de recherche (personnalisation, suggestions, résumé d'informations...)

105 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

W3C®

## Pourquoi des standards ? (1/2)

- Contrôle qualité (validation des pages Web)
- Réduction considérable du volume des documents
- Référencement efficace dans les moteurs de recherche
- Pérennité des documents
- Rétrocompatibilité avec les anciens navigateurs

106 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Éléments, balises et attributs

- **Élément** : nom, notion abstraite
  - Ex. Document HTML : html
- **Balise** : forme concrète d'un élément
  - Ex. de balise **ouvrante** : <html>
  - Ex. de balise **fermante** : </html>
- **Attribut** : propriété d'un élément (nom, valeur)
  - Ex. <a href="http://www.univ-lyon2.fr">

Nom                      Valeur

107 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Règles d'écriture XHTML

- Les noms des balises doivent être écrits en minuscules.
- Toute balise ouverte doit être fermée.
- Les chevauchements entre balise sont interdits (documents bien formés).
- Les noms des attributs doivent être écrits en minuscules.
- Les valeurs des attributs doivent être entre guillemets doubles.

108 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Squelette d'un document HTML

```

<!DOCTYPE html> <!-- Identification du type du document
                    (note : ceci est un commentaire) -->
<html lang="fr">
  <head> <!-- En-tête -->
  </head>
  <body> <!-- Corps du document -->
  </body>
</html>

```

109 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## En-tête

- Informations qui ne sont **pas** affichées mais qui sont utilisées à des fins diverses
- **Titre** : <title>...</title>
- **Métadonnées** : <meta />
  - Ex. Jeu de caractères, auteur, mots-clés, description de la page...
- **Note** : Une balise vide peut se noter <balise></balise> ou <balise />, au choix.

110 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Exemple d'en-tête

```

<head>
<!-- Jeu de caractères universel -->
<meta charset="utf-8" />

<!-- Autres métadonnées -->
<meta name="Author" content="Jérôme Darmont" />
<meta name="Keywords" content="Enseignement, Informatique" />
<meta name="Description" content="Exemple de page HTML" />

<!-- Titre -->
<title>Exemple de page HTML</title>
</head>

```

111 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Optimisation des performances

- **Préchargement DNS**
  - Ex. <link rel="dns-prefetch" href="//platform.twitter.com">
- **Préchargement de lien**
  - Ex. <link rel="prefetch" href="http://darmont.me/hello.html" />
  - Ex. <link rel="prefetch" href="http://darmont.me/picture.png" />
- **Prérendu de page**
  - Ex. <link rel="prerender" href="http://darmont.me/hello.html" />

112 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Corps du document

- **Éléments de structuration du document**
  - Invisibles
  - Permettent la mise en page ultérieure du document
  - Permettent la génération automatique, par ex., d'une table des matières, des figures, etc.
- **Éléments qui apparaissent explicitement à l'affichage**
  - Texte
  - Images
  - ...

113 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Structuration du document

- **Contenu principal** : <main> </main>
- **Section** : <section> </section>
- **Article** : <article> </article>
- **Encadré** : <aside> </aside>
- **Entête** : <header> </header>
- **Pied de page** : <footer> </footer>
- **Menu de navigation** : <nav> </nav>
- **Boîte de dialogue** : <dialog> </dialog>

114 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Titres, paragraphes, séparateurs

- **Titres (headings)** : Six niveaux notés h1 à h6
  - Ex. `<h1>Titre de niveau 1</h1>`
- **Paragraphes** (séparés entre eux par un espace)
  - Ex. `<p>Ceci est un paragraphe</p>`
- **Séparateurs** :
  - Retour à la ligne (sans espace) : `<br />`
  - Ligne horizontale : `<hr />`

115 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Autres blocs de texte prédéfinis

- **Adresse**
  - Ex. `<address>5 av. P. Mendès-France </address>`
- **Citation**
  - Ex. `<blockquote>Mignonne allons voir</blockquote>`
- **Texte préformaté**
  - Ex. `<pre>Le formatage sera conservé à l'écran</pre>`

116 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Formatage de texte

- Texte « appuyé » : `<strong>...</strong>`
- Emphase : `<em>...</em>`
- Surlignage : `<mark>...</mark>`
- Police « machine à écrire » : `<code>...</code>`
- En indice : `<sub>...</sub>`
- En exposant : `<sup>...</sup>`
- Caractères spéciaux :
  - < : `&lt;`                      & : `&amp;`
  - > : `&gt;`                      " : `&quot;`
  - espace insécable : `&nbsp;`

117 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Imbrication des balises

`<!-- Exemple -->`


```
<header><h1>Présentation du master IDSM</h1></header>
<article>
<p>L'<strong>Université Lumière Lyon 2</strong> et
l'<strong>Université Nationale d'Économie de Kharkiv</strong>
proposent une formation d'excellence (<em><strong>double diplôme
de master</strong></em>) en <em>informatique et
statistique</em>. <br /> Ces domaines se développent rapidement,
tant au niveau de la recherche que dans l'industrie, et offrent de
nombreux emplois.</p>
</article>
```

118 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Liens hypertextes

- **Forme générale** : `<a href="URL">libellé</a>`
- **URL : Uniform Resource Locator**
  - **Absolute** (adresse complète)
    - Ex. `http://www.hdeu.edu.ua`  
`ftp://ftp.ciril.fr`  
`mailto:jerome.darmont@univ-lyon2.fr`
  - **Relative** (à partir du répertoire courant)
    - Ex. `page_suivante.html`  
`rep/page_dans_repertoire.html`
  - **Exemple** : `<a href="http://www.univ-lyon2.fr">Université Lyon 2</a>`



119 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML




## Ancres

- Permettent un lien vers un endroit précis dans une page Web
- **Définition dans une page** : attribut id
  - Ex. `<h1 id="menu">MENU</h1>`
- **Référence depuis la même page**
  - Ex. `<a href="#menu">Aller au menu</a>`
- **Référence depuis une autre page**
  - Ex. `<a href="page.html#menu">Retour menu</a>`

120 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Images

- **Formats reconnus :**   
- **Balise :** ``
  - Ex. ``
  - Ex. ``
- **Lien sur une image**
  - Ex.
 

```
<a href="http://www.univ-lyon2.fr">
  
</a>
```

UNIVERSITÉ  
LUMIÈRE  
LYON 2

121 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Multimédia

- **Sons :** élément `<audio>` `</audio>`
- **Vidéos :** élément `<video>` `</video>`
- **Attributs communs**
  - src : URL du contenu
  - controls : affiche les contrôles du lecteur multimédia
  - autoplay : démarrage dès que possible
  - loop : en boucle
  - muted : sans son
- **Exemple**
  - `<audio src="test.mp3" controls="controls" loop="loop">`  
Le format MP3 n'est pas supporté par votre navigateur.  
`</audio>`

122 Programmation Web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Figures

- Élément de structuration (comme section, article...)
- **Balises :** `<figure>` `</figure>`
- **Légende :** `<figcaption>` `</figcaption>`
- **Exemple :**

```
<figure>
  
  <video src="test.ogg" controls="controls" muted="muted">
    Le format .ogg n'est pas supporté.
  </video>
  <audio src="chord.wav" controls="controls">
    Le format .wav n'est pas supporté.
  </audio>
  <figcaption>Tout le multimédia de HTML5</figcaption>
</figure>
```


123 Programmation Web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Jauges

- **Barre de progression**

```
<progress max="100" value="75"></progress>
```

Apparence : 
- **Barre de mesure**

```
<meter min="0" max="100" low="10" high="90"
  optimum="50" value="75">75 %</meter>
```

Apparence : variable selon les navigateurs

25 Programmation Web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Listes ordonnées et non-ordonnées

- **Listes à puces :** `<ul>` `</ul>`
- **Listes numérotées :** `<ol>` `</ol>`
- **Exemple**

```
<ul>
  <li>1er élément</li>
  <li>2ème élément</li>
  <li>3ème élément</li>
</ul>
```

125 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

HTML

## Listes de définitions

- **Exemple**

```
<dl>
  <dt>1er terme</dt>
  <dd>Définition du 1er terme</dd>
  <dt>2ème terme</dt>
  <dd>Définition du 2ème terme</dd>
</dl>
```
- **Résultat à l'affichage**

```
1er terme
  Définition du 1er terme
2ème terme
  Définition du 2ème terme
```

126 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Imbrication de listes

- Exemple

```
<ul>
  <li>Elément non ordonné 1
    <ol>
      <li>Sous-élément ordonné 1.1</li>
      <li>Sous-élément ordonné 1.2</li>
    </ol>
  </li>
</ul>
<li>Elément non ordonné 2</li>
</ul>
```

127

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Tableaux simples

- Tableau : Ensemble de lignes elles-mêmes constituées de cellules
- Définition d'un tableau : `<table> </table>`
- Légende : `<caption> </caption>`
- Définition d'une ligne : `<tr> </tr>`
- Définition d'une cellule d'en-tête : `<th> </th>`
- Définition d'une cellule normale : `<td> </td>`

128

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Exemple de tableau

```
<table>
  <caption>Mes totaux</caption>
  <tr>
    <th>Jour de la semaine</th> <th>Montant</th> <th>Total</th>
  </tr>
  <tr>
    <td>Lundi</td> <td>456 euros</td> <td>456 euros</td>
  </tr>
  <tr>
    <td>Mardi</td> <td>200 euros</td> <td>656 euros</td>
  </tr>
</table>
```

129

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Apparence du tableau exemple

Mes totaux

Jour de la semaine	Montant	Total
Lundi	456 euros	456 euros
Mardi	200 euros	656 euros

130

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Tableaux à cellules recouvrantes (1/2)

C1	C2	L12	L1
C12			L2

131

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Tableaux à cellules recouvrantes (2/2)

- "Fusion" de cellules (`<td>` ou `<th>`)
- Cellule sur plusieurs colonnes : attribut `colspan`
  - Ex. `<table> <tr> <td>C1</td> <td>C2</td> </tr> <tr> <td colspan="2">C12</td> </tr> </table>`
- Cellule sur plusieurs lignes : attribut `rowspan`
  - Ex. `<table> <tr> <td rowspan="2">L12</td> <td>L1</td> </tr> <tr> <td>L2</td> </tr> </table>`

132

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Accessibilité (1/2)

26,4 % de la population française souffre d'une entrave dans une activité quotidienne et 10 % est affectée d'un handicap. (SVM, novembre 2008)

### Préceptes d'accessibilité :

- Proposer un texte alternatif aux images et aux contenus multimédias en général (description, transcription...)
- Utiliser des feuilles de style

133

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Accessibilité (2/2)

- Prévoir des en-têtes de ligne et de colonnes dans les tableaux
- Lier chaque champ de formulaire à une légende
- Définir des raccourcis clavier
- Prévoir une alternative HTML aux technologies propriétaires (ex. Flash)
- Éviter l'usage des cadres (*frames*)
- Prévenir des changements de langue dans le texte
- Expliciter les liens (« page d'accueil » >> « cliquer ici »)

134

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Plan du cours

- ✓ Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - ✓ Transmission de variables d'une page à une autre
  - ✓ Architecture Modèle-Vue-Contrôleur
  - ✓ Gabarits
- Rappels
  - ✓ Langage HTML5
  - Feuilles de style CSS3

135

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Définition et caractéristiques

- *Cascading Style Sheets (CSS)* : Feuilles de style en cascade
- Première spécification par le W3C en 1996
- Séparation du contenu et de la présentation
- Définition de styles génériques pour les éléments
- Syntaxe différente de HTML
- Possibilité de validation automatique

136

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Avantages

- Gestion simplifiée et globale de la présentation d'un site
- Coûts de développement et de maintenance allégés (code plus simple)
- Code source allégé
- Possibilité de présentations différentes selon le profil utilisateur, la résolution écran...
- Meilleure accessibilité

137

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>



## Définitions de styles

- Trois localisations possibles (+local au +global)
- Dans un élément HTML : attribut style
  - Ex. `<p style="color: blue; text-align: center;">...</p>`
- Dans une page HTML : élément `<style>`
  - Dans l'en-tête de la page Web (head) : `<style type="text/css">...</style>`
- Dans un fichier séparé
  - Appel dans l'en-tête de la page Web (head) : `<link rel="stylesheet" type="text/css" href="feuille_style.css" />`

138

Programmation web

<http://eric.univ-lyon2.fr/~jdamont/>

## Notion de cascade

- **Imbrication des définitions** : la plus locale l'emporte.
  - Ex. Définition de style dans un fichier externe vs. définition dans un élément
  - ⇒ la définition locale de l'élément prévaut
- **Imbrication des éléments** : le style d'un élément s'applique aux éléments imbriqués.
  - Ex. `<p>...<em>...</em>...</p>`
  - Si un style est appliqué à `<p>`, il s'applique à `<em>`.

139 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Unités de taille

- **Unités absolues**
  - px : pixel
  - pt : point
- **Unités relatives** (à privilégier, cf. *Responsive design*)
  - em : hauteur de ligne
  - % : proportion par rapport à la taille de la fenêtre

140 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Spécification de couleurs

- **Nom prédéfini** : aqua, black, blue, fuchsia, green, gray, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
- **Spécification RGB** (Rouge Vert Bleu - RVB)
  - Ex. Jaune : `rgb(255, 255, 0)`
- **Code RGB hexadécimal**
  - Ex. Jaune : `#FFFF00`

141 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Écriture d'une feuille de style

- **Feuille de style** : ensemble de règles
  - Un **sélecteur**
  - Des **propriétés**
  - Une **valeur** pour chaque propriété
- **Exemple**

```
h1 { color: yellow;
      font-weight: bold;
    }
```

142 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Sélecteurs

- **Sélecteur universel** : tous les éléments HTML
  - Ex. `* { color: blue; }`
- **Style d'une catégorie d'éléments**
  - Ex. `h1 { color: #FF00FF; }`
- **Style de plusieurs catégories d'éléments**
  - Ex. `h1, h2, h3, p { color: green; }`
- **Style pour éléments imbriqués**
  - Ex. `p em { color: red; }` /\* Italiques dans un paragraphe \*/

(Commentaire)

143 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Classe de style (1/2)

- **Sélecteur défini par l'utilisateur** qui permet de dissocier un style d'éléments particuliers
  - Ex.
 

```
.style_rouge { color: red; }
.style_bleu { color: blue; }
p.parag_vert { color: green; }
```
- **Sélecteur associé à un identifiant particulier**
  - Ex. `#titre { text-align:center; color: navy; }`

144 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Classe de style (2/2)

- **Utilisation dans une page Web**
  - Ex. 

```
<h1 class="style_rouge">Titre rouge</h1>
<p class="style_rouge">Texte rouge</p>
<p class="style_bleu">Texte bleu</p>
<p class="parag_vert">Texte vert</p>
```
- **Application à un ensemble d'éléments : <div> </div>**
  - Ex. 

```
<div class="style_bleu">
  <h1>...</h1>
  <p>...</p>
  <p>...</p>
</div>
```

145 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Application des styles

- **À une partie d'un paragraphe : <span> </span>**
  - Ex. 

```
<p class="style_rouge">Ceci est rouge sauf
  <span class="style_bleu">ce qui est bleu</span>
</p>
```
- **À un élément identifié :**
  - Ex. 

```
<div id="titre"> ... </div>
<p id="titre"> ... </p>
```

146 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de couleur

- **color** color: black; color: #000000;
- **background-color** background-color: white; background-color: #FFFFFF;
- **background-image** background-image: url("URL");
- **background-repeat** background-repeat: no-repeat;
- **background-position** background-position: center;

147 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de typographie

- **font-family** font-family: times; font-family: arial, verdana;
- **font-size** font-size: 1.5em;
- **font-style** font-style: italic;
- **font-weight** font-weight: bold; font-weight: bolder;
- **text-decoration** text-decoration: underline; text-decoration: line-through;

148 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de texte

- **text-align** text-align: left; text-align: right; text-align: center; text-align: justify;
- **text-indent** text-indent: 10pt; text-indent: 15px; text-indent: 5%;

149 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de boîte englobante (1/2)

150 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de boîte englobante (2/2)

- width** width: 80pt;
- height** height: 25%;
- padding** padding: 5px;  
-left, -right, -top, -bottom padding-bottom: 10px;
- margin** margin: 10px;  
-left, -right, -top, -bottom margin-left: 10%; margin-right: auto; margin-left: auto;
- border** border: 2px solid #FF00FF;  
-width, -style, -color border-style: dotted; border-radius: 15px;
- overflow** overflow: auto;

Boîte centrée

Coins ronds

151 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés d'hyperlien

- Hyperliens**
  - Ex. a { color: blue; }
- Hyperliens visités**
  - Ex. a:visited { color: purple; }
- Hyperliens survolés (avec le pointeur de souris)**
  - Ex. a:hover { color: white; background-color: blue; }
- Hyperliens activés**
  - Ex. a:active { color: purple; }

152 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de liste

- Listes non-ordonnées**
  - list-style list-style: disc; list-style: circle; list-style: square;
  - list-style-image list-style-image: url("URL");
- Listes ordonnées**
  - list-style list-style: decimal; list-style: upper-roman; list-style: lower-alpha;

153 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Propriétés de tableau

- Élément table :** border, width, border-collapse: collapse...
- Élément caption :** caption-side: top; caption-side: bottom;
- Élément tr :** height
- Éléments td et th :** border, width... vertical-align: top; vertical-align: middle; vertical-align: bottom;

154 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement flottant (1/4)

Exemple 1 : menu flottant à droite

Menu

(style .menu\_jaune)

100 pixels

155 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement flottant (2/4)

- Exemple 1 :
- CSS**

```
.menu_jaune { float: right; background-color: yellow; width: 100px; }
```
- HTML**

```
<div class="menu_jaune">
  <p>Menu</p>
</div>
</div>
<!-- Contenu de la page -->
</div>
```

156 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement flottant (3/4)

Conteneur (style .page) 100 %

Exemple 2 : pagination en trois colonnes

Colonne 1 (style .colonne) 33 %	Colonne 2 (style .colonne) 33 %	Colonne 3 (style .colonne) 33 %
---------------------------------------	---------------------------------------	---------------------------------------

157 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement flottant (4/4)

- Exemple 2 :
- CSS**

```
.page { float: left; width: 100%; }
.colonne { float: left; width: 33%; }
```
- HTML**

```
<div class="page"> <!-- Conteneur -->
  <div class="colonne"> </div>
  <div class="colonne"> </div>
  <div class="colonne"> </div>
</div>
```

158 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement absolu (1/2)

- HTML**

```
<div class="boite_verte">
  <p>Bla bla sur fond vert...</p>
  <div class="boite_jaune">
    <p>Idem sur fond jaune</p>
  </div>
</div>
```

159 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Positionnement absolu (2/2)

- CSS**

```
.boite_verte { position: relative;
  background-color: #00FF00;
  width: 15em; }

.boite_jaune { position: absolute;
  top: 1em;
  right: 1em;
  background-color: #FFFF00; }
```

160 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Exemple de mise en page : Objectif

```

- <body>
  - <main>
    - <header>
    - <nav>
    - <section>
      - <article>
      - <article>
      - <article>
    - <footer>
  
```

161 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Exemple de mise en page : HTML

```

<main>
  <header>
    <h1>Exemple de mise en page</h1>
  </header>
  <div id="conteneur">
    <nav>
      <a href="http://www.univ-lyon2.fr">Université Lyon 2</a>
    </nav>
    <section>
      <article>Article 1</article>
      <article>Article 2</article>
      <article>Article 3</article>
    </section>
  </div>
  <footer>
    &copy; IDS
  </footer>
</main>

```

162 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Exemple de mise en page : CSS

```

* {
  font-family: arial;
}
body {
  background-color: silver;
}
main {
  width: 1000px;
  margin-left: auto;
  margin-right: auto;
  background-color: white;
  padding: 1em;
}
header {
  background-color: yellow;
  text-align: center;
  padding: 1em;
  border-radius: 15px;
}
#conteneur {
  width: 100%;
  margin-top: 1em;
}
nav {
  float: left;
  width: 200px;
  background-color: red;
  padding: 1em;
  border-radius: 15px;
}
section {
  float: left;
  width: 750px;
  background-color: lime;
  margin-left: 1em;
  border-radius: 15px;
}
article {
  background-color: aqua;
  margin: 1em;
}
footer {
  background-color: fuchsia;
  text-align: center;
  padding: 1em;
  border-radius: 15px;
}

```

163 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Responsive design

164 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Responsive design en pratique (1/2)

- **Grilles fluides** : utilisation des unités relatives

<http://blog.teamtreehouse.com/modern-field-guide-responsive-web-design>

- **Sidebar** :  $300px / 960px = 31.25\%$
- **Main Content** :  $640px / 960px = 66.67\%$
- **Margin** :  $20px / 960px = 2.08\%$

165 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Responsive design en pratique (2/2)

- **Images fluides**
  - Taille variable en fonction de la grille fluide
  - **CSS**  
`img { max-width: 100%; }`
- **Requêtes de média**
  - Application de style conditionnelle
  - **CSS**  
`@media screen and (min-width: 300px) { /* styles pour mobiles */ }`  
`@media screen and (min-width: 600px) { /* styles pour tablettes */ }`  
`@media screen and (min-width: 900px) { /* styles pour ordis */ }`

166 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Grilles (1/3)

<https://blog.goetter.fr/2016/01/02/grid-layout-vers-la-grille-parfaite/>

- Remplacent les mises en forme classiques
- **Définition d'une grille**

```

.conteneur {
  display: grid; /* conteneur grille */
  grid-template-columns: repeat(4, 1fr); /* 4 cols, même taille */
  grid-gap: 1rem; /* gouttière verticale et horizontale */
}

```

167 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>

## Grilles (2/3)

- **Éléments à taille multiple**

```

@media (min-width: 481px) {
  .col-span {
    grid-column: span 2; /* double taille en largeur */
  }
  .row-span {
    grid-row: span 2 /* double taille en hauteur */
  }
}

```

168 Programmation web <http://eric.univ-lyon2.fr/~jdamont/>



## Grilles (3/3)

- Grille adaptative

```
/* Tablette : 2 colonnes */
@media (max-width: 640px) {
  .container {
    grid-template-columns: repeat(2, 1fr);
  }
}
/* Téléphone mobile : 1 colonne */
@media (max-width: 480px) {
  .container {
    grid-template-columns: 1fr;
  }
}
```

169

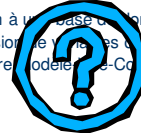
Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>



## Plan du cours

- ✓ Langage de script PHP
  - ✓ Bases de PHP
  - ✓ PHP objet
  - ✓ Connexion à une base de données
  - ✓ Transmission de données d'une page à une autre
  - ✓ Architecture Modèle-Vue-Contrôleur
  - ✓ Gabarits
- ✓ Rappels
  - ✓ Langage HTML5
  - ✓ Feuilles de style CSS3



170

Programmation web

<http://eric.univ-lyon2.fr/~jdarmont/>