

UNIVERSITE  
LUMIERE  
LYON 2



## Programmation Web

Master 1 IDSM-Kharkiv  
Année 2016-2017  
Jérôme Darmont

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

## Actualité du cours

-  [http://eric.univ-lyon2.fr/~jdarmont/?page\\_id=449](http://eric.univ-lyon2.fr/~jdarmont/?page_id=449)
-  <http://eric.univ-lyon2.fr/~jdarmont/?feed=rss2>
-  [https://twitter.com/darmont\\_lyon2](https://twitter.com/darmont_lyon2) hashtag #webidsm

2

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

## Plan du cours

- Langage HTML
- Éléments de design Web
- Feuilles de style en cascade (CSS)
- Langage de script PHP
- Connexion à une base de données (MySQL)

3

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

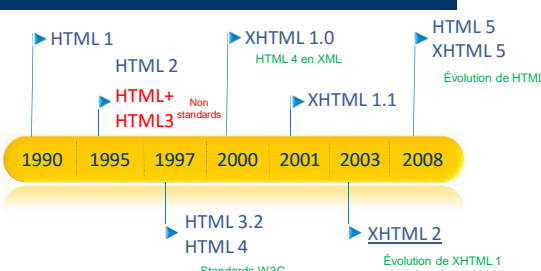
## Un peu d'histoire...

- **HTML** : *HyperText Markup Language*
  - Liens hypertextes
  - Langage à base de balises
- **Objectif** : publier sur la *World Wide Web* (ou WWW, W3, ou Web) des documents formatés
- Issu de **SGML** (*Standard Generalized Markup Language*), langage de définition de langages
- Développé par **Tim Berners-Lee** au CERN (Suisse) en 1990 et utilisé sur le Web depuis

4

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

## Versions successives de HTML



5

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

## 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

6

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

## 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.

7

Programmation Web

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

## Squelette d'un document HTML

```
<!DOCTYPE html> <!-- Identification du type du document
                    (note : ceci est un commentaire) -->
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

8

Programmation Web

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

## En-tête

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

9

Programmation Web

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

## Exemple d'en-tête

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

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

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

10

Programmation Web

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

## 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" />

11

Programmation Web

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

## 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
  - ...

12

Programmation Web

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



## 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>`

19

Programmation Web

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

## Images

- **Formats reconnus** : 
- **Balise** : ``
  - Ex. ``
  - Ex. ``
- **Lien sur une image**
  - Ex. `<a href="http://www.univ-lyon2.fr"></a>`

UNIVERSITÉ  
LUMIÈRE  
LYON 2

20

Programmation Web

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

## Multimédia

- **Sons** : balise `<audio> </audio>`
- **Vidéos** : balise `<video> </video>`
- **Attributs communs**
  - src : URI 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>`

21

Programmation Web

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

## Figures

- Élément de structuration (comme section, article...)
- **Balise** : `<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>
```

22

Programmation Web

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

## 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

21

Programmation Web

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

## 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>
```

24

Programmation Web

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

## 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
```

25

Programmation Web

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

## 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>
  <li>Elément non ordonné 2</li>
</ul>
```

26

Programmation Web

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

## 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>`

27

Programmation Web

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

## 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>
```

28

Programmation Web

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

## Apparence du tableau exemple

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

29

Programmation Web

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

## Tableaux à cellules recouvrantes

|     |    |     |    |
|-----|----|-----|----|
| C1  | C2 | L12 | L1 |
| C12 |    |     | L2 |

30

Programmation Web

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

## Tableaux à cellules recouvrantes

- "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">L1</td>`  
`<td>L1</td> </tr>`  
`<tr> <td>L2</td> </tr> </table>`

31

Programmation Web

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

## Plan du cours

- ✓ Langage HTML
- Éléments de design Web
- Feuilles de style en cascade (CSS)
- Langage de script PHP
- Connexion à une base de données (MySQL)

32

Programmation Web

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

## Dix erreurs de base à éviter

- Fonds sombres, images de fonds chargées, couleurs trop nombreuses
- Gadgets technologiques (animations, textes qui défilent, sons, Javascripts divers...)
- Surcharger la page de texte ou d'images : trop d'information tue l'information
- Absence d'aide à la navigation
- Liens de couleur non standard

33

Programmation Web

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

## Dix erreurs de base à éviter

- Ouvrir de nouvelles fenêtres, des *pop-ups*, etc.
- Pages lourdes à charger (contenant des images volumineuses, par exemple)
- Pages trop larges (*scrolling* latéral)
- Pages trop longues (*scrolling* vertical)
- Requérir de l'utilisateur une configuration particulière ("site optimisé pour tel navigateur en telle résolution")

34

Programmation Web

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

## Design « Web 2.0 » \*

- Qu'est-ce que le Web 2.0 ?
  - Développement de la Web économie
  - Interactivité accrue des sites Web (AJAX)
  - Web communautaire, réseaux sociaux
  - Nouvelle école de design Web

\* D'après [webdesignfromscratch.com](http://webdesignfromscratch.com)

35

Programmation Web

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

## Principe fondamental : la simplicité

- Chaque site / chaque page a un but précis.
- L'attention de l'utilisateur n'est pas infinie.
- Le concepteur **doit** aider l'utilisateur à trouver l'information qu'il cherche.
- Il ne faut pas noyer les informations importantes.
- ⇒ Minimiser le bruit = économie, **simplicité**

36

Programmation Web

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

## Comment « faire simple » ?

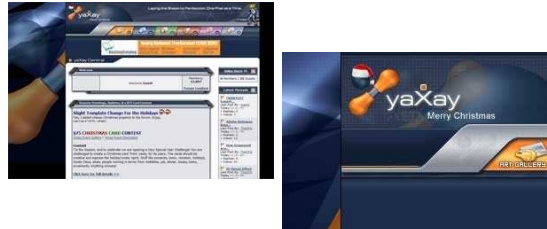
- Minimiser le ratio  $\frac{\text{pixels décoratifs}}{\text{pixels porteurs d'information}}$
- Supprimer tout les éléments non strictement nécessaires
- Essayer des solutions alternatives pour arriver au même résultat plus simplement
- Utiliser les éléments visuel pour **communiquer**, jamais pour décorer

37

Programmation Web

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

## Exemple de simplification



38

Programmation Web

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

## Éléments de présentation

- Centrer le site dans la fenêtre du navigateur
- Limiter le nombre de colonnes à 2 (3 maximum)



Contre-exemple !

39

Programmation Web

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

## Éléments de présentation

- Distinguer clairement une section d'entête (logo ou marque **bien visible**, zone de navigation)



40

Programmation Web

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

## Éléments de présentation

- Identifier clairement les sections de la page
  - Utiliser des couleurs (mais danger de distraction de l'utilisateur)
  - Espacer nettement les sections



41

Programmation Web

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

## Éléments de présentation

- **Texte en gros caractères**
  - Mise en valeur des informations importantes
  - Facilité de lecture (accessibilité)
- **Titres en gras**
  - Attire l'attention sur les zones importantes du site



42

Programmation Web

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

## Éléments de présentation

- Couleurs vives
- Dégradés
- Textures (reflets, transparence...)
- Icônes
- Délimitation de sections
- Mise en valeur d'éléments importants
- Mise en place d'une ambiance



Ne pas attirer l'attention ailleurs que sur le contenu

À utiliser avec parcimonie !

43

Programmation Web

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

## Navigation

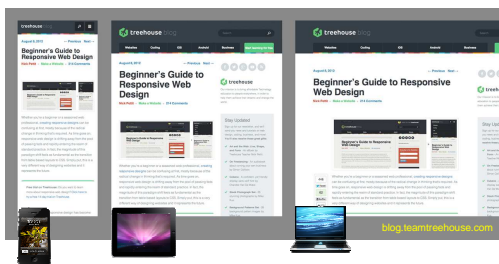
- **Objectifs** : l'utilisateur doit savoir facilement
  - Où il se trouve dans le site
  - Où il peut aller
  - Quelles sont options disponibles sur la page courante
- **Menu de navigation global**
  - Clairement séparé du contenu
  - Différencié en termes de couleur et de forme
  - Gros, clairement et toujours visible
  - Texte des liens explicite
- **Contenu** : Hyperliens bien différenciés du texte normal

44

Programmation Web

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

## Reponsive design



45

Programmation Web

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

## Qu'est-ce que le W3C ?

- **W3C** : *World Wide Web Consortium*
  - Organisation internationale indépendante des éditeurs de logiciels
  - <http://www.w3.org>
- **Objectif** : Développement de standards et de « bonnes pratiques » du Web
- **Directeur** : Tim Berners-Lee



46

46

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

## Pourquoi utiliser les standards du W3C ?

- Éviter la "balkanisation du Web" des années 90
- Interopérabilité et portabilité (PDA, 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...)

47

Programmation Web

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

## Pourquoi utiliser les standards du W3C ?

- 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

48

Programmation Web

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





## Accessibilité

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

49

Programmation Web

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

## Accessibilité

- 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 »)

50

Programmation Web

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

## Plan du cours

- ✓ Langage HTML
- ✓ Éléments de design Web
- Feuilles de style en cascade (CSS)
- Langage de script PHP
- Connexion à une base de données (MySQL)

51

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 balises
- Syntaxe différente de HTML
- Possibilité de validation automatique

52

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

53

Programmation Web

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

## Définitions de styles

- Trois localisations possibles (+local au +global)
- **Dans une balise HTML** : attribut style
  - Ex. `<p style="color: blue; text-align: center;">...</p>`
- **Dans une page HTML** : balise `<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" />`

54

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 une balise
  - ⇒ la définition locale de la balise prévaut
- **Imbrication des balises** : le style d'une balise s'applique aux balises imbriquées.
  - Ex. `<p>...<em>...</em>...</p>`
  - Si un style est appliqué à `<p>`, il s'applique à `<em>`.

55

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

56

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`

57

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;
}
```

58

Programmation Web

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

## Sélecteurs

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

(Commentaire)

59

Programmation Web

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

## Classe de style

- **Sélecteur défini par l'utilisateur** qui permet de dissocier un style de balises particulières
  - 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; }`

60

Programmation Web

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

## Classe de style

- 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>
```

61

## 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>
```

62

## 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;

63

## 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;

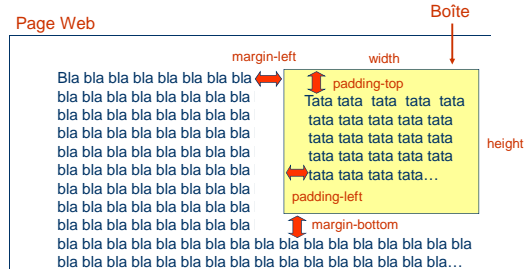
64

## 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%;

65

## Propriétés de boîte englobante



66

## Propriétés de boîte englobante

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

Boîte centrée

Coins ronds

67

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; }

68

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;

69

Programmation Web

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

## Propriétés de tableau

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

70

Programmation Web

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

## Exemple de style de tableau

Style 1

|    |    |
|----|----|
| C1 | C2 |
| C3 | C4 |

Style 2

|    |    |
|----|----|
| C1 | C2 |
| C3 | C4 |

71

Programmation Web

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

## Exemple de style de tableau

- **HTML**

```
<table>
  <tr> <td>C1</td> <td>C2</td> </tr>
  <tr> <td>C3</td> <td>C4</td> </tr>
</table>
```
- **Style 1 (CSS)**

```
table, td { border: 1px solid black; }
```
- **Style 2 (CSS)**

```
table, td { border: 1px solid black; }
table { border-collapse: collapse; }
```

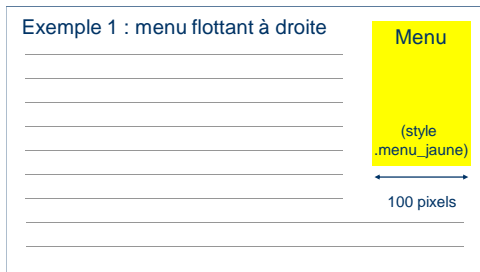
72

Programmation Web

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

## Positionnement flottant

Exemple 1 : menu flottant à droite



73

Programmation Web

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

## Positionnement flottant

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>
```

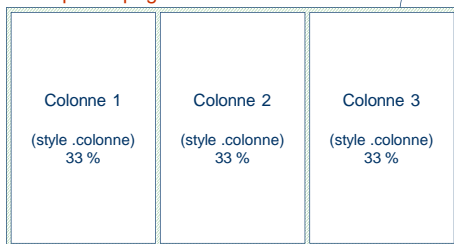
74

Programmation Web

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

## Positionnement flottant

Exemple 2 : pagination en trois colonnes



75

Programmation Web

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

## Positionnement flottant

Exemple 2 :

- **CSS**

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

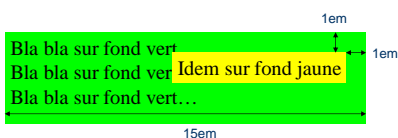
```
<div class="page"> <!-- Conteneur -->
  <div class="colonne"> </div>
  <div class="colonne"> </div>
  <div class="colonne"> </div>
</div>
```

76

Programmation Web

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

## Positionnement absolu



- **HTML**

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

77

Programmation Web

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

## Positionnement absolu

- **CSS**

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

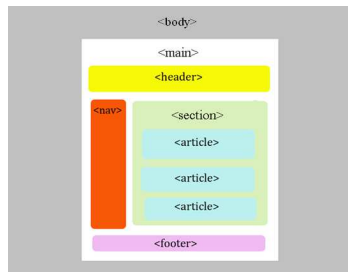
.boite_jaune { position: absolute;
top: 1em;
right: 1em;
background-color: #FFFF00; }
```

78

Programmation Web

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

## Exemple de mise en page : Objectif



79

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; IDSM
  </footer>
</main>
```

80

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;
  padding: 1em;
  border-radius: 15px;
}
article {
  background-color: aqua;
  margin: 1em;
}
footer {
  background-color: fuchsia;
  text-align: center;
  padding: 1em;
  border-radius: 15px;
}
```

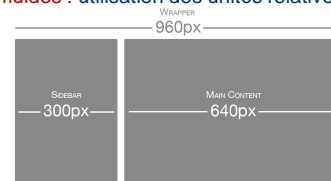
81

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

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

## Responsive design en pratique

- Grilles fluides : utilisation des unités relatives



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

82

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

## Responsive design en pratique

- 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 */ }
```

83

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

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

## Grilles

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

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

84

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

## Grilles

- **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 */
  }
}
```

## Grilles

- **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;
  }
}
```

## Plan du cours

- ✓ Langage HTML
- ✓ Éléments de design Web
- ✓ Feuilles de style en cascade (CSS)
- Langage de script PHP
- Connexion à une base de données (MySQL)

## Généralités

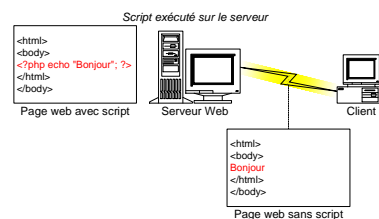
- **PHP** : *PHP Hypertext Preprocessor*
- Langage de script permettant d'insérer de la programmation dans des pages Web
- ⇒ Pages Web **dynamiques**
- **Versions** :
  - 1994 : Conception par Rasmus Lerdorf
  - 1995 : Première version publique
  - 1995 : PHP/FI
  - 1997 : PHP 3
  - 2000 : PHP 4
  - 2004 : PHP 5

## Généralités

- 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 et Windows

## Généralités

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



## Syntaxe de base

- **Passage du HTML à PHP** (échappement) : `<script language="php"> </script>` ou `<?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) `*/`

91

Programmation Web

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

## Variables et types

- **Variables** : préfixées par le caractère `$`
- PHP ne nécessite **pas** de déclaration explicite du type de variable (☹).
- **Types de données** :
  - Nombres entiers : `int`, `integer` Ex. d'affectation `$i = 1;`
  - Nombres réels : `real`, `double`, `float` `$pi=3.14;`
  - Chaînes de caractères : `string` `$ch="oui";`
- **Conversion de type** : "cast" comme en C
  - Ex. `$pi = (int) $pi;` // `$pi` est égale à 3

92

Programmation Web

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

## Variables et types

- **Tableaux** : scalaires ou associatifs, création par assignation des valeurs
  - Ex. `$ts[0] = "Chaîne 0";` // **Indiçage à partir de 0**  
`$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)`

93

Programmation Web

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

## Variables et types

- **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));`

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

94

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

95

Programmation Web

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

## Opérateurs

- **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)

96

Programmation Web

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



## Opérateurs

- Concaténation de chaînes de caractères : .

- Ex. 

```
$ch1 = $ch2 . $ch3;
$ch1 .= "<br />";
```

- Caractères spéciaux dans les chaînes (échappement)

- Antislash : 

```
\\
```
- Dollar : 

```
\$
```
- Guillemets : 

```
\"
```

97

Programmation Web

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

## Opérateurs

- Opérateurs logiques

- ET : and ou &&
- OU : or ou ||
- OU exclusif : xor
- NON : !

- Opérateurs de comparaison

- Égalité/Différence : 

```
== !=
```
- Inférieur/Supérieur : 

```
< >
```
- Inférieur ou égal/Supérieur ou égal : 

```
<= >=
```

98

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);
```

- **Formatage** : Utilisation des balises HTML

- Ex. 

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

- **Saut de ligne dans le code source**

- Ex. 

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

99

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);
```

100

Programmation Web

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

## Structures de contrôle

- **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;
}
```

101

Programmation Web

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

## Structures de contrôle

```
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";
```

102

Programmation Web

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

## Structures de contrôle

- **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\"";
        }
```

103

Programmation Web

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

## Structures de contrôle

- **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;
        }
```

104

Programmation Web

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

## Structures de contrôle

- **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 />";
        }
```

105

Programmation Web

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

## Structures de contrôle

- **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 />";
        }
```

106

Programmation Web

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

## Structures de contrôle

- **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 URI)

- `require("mes_fonctions.inc.php");`
- `include("une_page_web.html");`
- `include("http://serveur.fr/pg.html");`

107

Programmation Web

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

## Fonctions

- **Squelette de définition de fonction**

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

- **Exemple**

```
function mon_max ($n1, $n2) {
    return ($n1>$n2) ? $n1 : $n2;
}
```

- Une fonction doit être définie avant d'être appelée.

108

Programmation Web

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

## Fonctions

- **Passage de paramètres par référence** (paramètres 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
```

109

Programmation Web

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

## Visibilité des variables

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

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

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

110

Programmation Web

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

## Séparation de PHP et HTML

- Séparation des tâches de programmation et de design Web
- Maintenance facilitée
  - Lecture plus aisée des pages HTML + PHP
  - Modification facile du code sans affecter la mise en page et vice-versa
- Réutilisation de code
  - ⇒ Utilisation des fonctions `include()` et `require()`
    - Ex. Définition dans des fichiers séparés d'en-têtes et de pieds de page communs à plusieurs pages

111

Programmation Web

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

## Plan du cours

- ✓ Langage HTML
- ✓ Éléments de design Web
- ✓ Feuilles de style en cascade (CSS)
- ✓ Langage de script PHP
- Connexion à une base de données (MySQL)

112

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 : **MySQL**
  - Serveur de BD SQL multi-utilisateurs rapide
  - Développé depuis 1995 (version actuelle : MySQL 5)
  - Disponible sous licence GPL ou commerciale
- Utilisation de l'extension **PHP Data Objects (PDO)** : interface d'accès à une base de données (divers SGBD supportés dont MySQL ; nécessite PHP 5.1+)

113

Programmation Web

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

## Éléments de langage SQL

- **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))`

114

Programmation Web

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

## Éléments de langage SQL

- **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

115

Programmation Web

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

## Éléments de langage SQL

### Exemple de création de table

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

116

Programmation Web

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

## Éléments de langage SQL

- **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

117

Programmation Web

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

## Éléments de langage SQL

- **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

118

Programmation Web

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

## PDO : 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";  
}

119

Programmation Web

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

## PDO : Exécution d'une requête

- **Requête d'interrogation**  
\$idconn->query(*requête\_SQL*)
  - Ex. \$c->query("select nom, prenom from etudiant");
- **Accès au résultat de la requête** (ligne par ligne)
  - Ex. foreach(\$c->query("select nom, prenom from etudiant") as \$l)  
    echo \$l["nom"]. " " . \$l["prenom"]. "<br />\n";
- **Requête de mise à jour**  
\$idconn->exec(*requête\_SQL*)
  - Ex. \$c->exec("update etudiant set note = note + 1");
  - Retourne le nombre de n-uplets mis à jour

120

Programmation Web

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

## PDO : Exemple pratique

```
try {
    // Connexion
    $c = new PDO("mysql:host=localhost;dbname=darmont", "darmont", "x");
    // Requête d'interrogation
    $q = "select nom, prenom from etudiant";
    $res = $c->query($q);
    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";
    // Requête de mise à jour
    $res = $c->exec("update etudiant set note = note + 1");
    echo "<p>$res ligne(s) modifiée(s)</p>\n";
} catch (PDOException $erreur) { // Gestion des erreurs
    echo "<p>Erreur : " . $erreur->getMessage() . "</p>\n";
}
```

121

Programmation Web

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

## 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

122

Programmation Web

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

## Formulaires HTML

- **Formulaires** : permettent la saisie de données dans une page Web
- **Définition** : `<form>` `</form>`
  - Attribut **action** : URI 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)

123

Programmation Web

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

## Formulaires HTML

- **Différence entre les méthodes get et post**
  - **get** : apparition des valeurs saisies en paramètres de l'URI 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>`

124

Programmation Web

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

## Formulaires HTML

- **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" />
```

125

Programmation Web

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

## Formulaires HTML

- **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" />`

126

Programmation Web

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

## Formulaires HTML

- **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" />`

127

Programmation Web

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

## Formulaires HTML

- **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>`

128

Programmation Web

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

## Formulaires HTML

- **Liste déroulante** : `<select> </select>`
  - Attribut `name` : nom de la variable choix
  - Balise `<option> </option>` : élément 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>
```

129

Programmation Web

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

## Formulaires HTML

- Options d'accessibilité (navigation alternative)
- **Description de champ** : `<label> </label>`
  - Ex. `<label for="id_nom">`  
Nom : `<input type="text" id="id_nom" name="nom" />`  
`</label>`
- **Légende de zone de saisie** : `<legend> </legend>`
  - Ex. `<fieldset>`  
`<legend>État civil de l'étudiant</legend>`  
...  
`</fieldset>`

130

Programmation Web

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

## Formulaires HTML

```
<!-- 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>
```

131

Programmation Web

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

## Exploitation des données d'un formulaire avec PHP

- 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

132

Programmation Web

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

## Mise à jour d'une base de données avec PHP

```
<?php // Suite de l'exemple : document ajout_etu.php
$c = new PDO("mysql:host=localhost;dbname=darumont",
    "darumont", "x");

$nom = $_POST["nom"];
$age = $_POST["age"];
$annee = $_POST["annee"];

$requete = "insert into etudiant values ('$nom', $age, '$annee')";
$resultat = $c->exec($requete);
if ($resultat) echo "<p>Insertion effectuée</p>";
?>
```

133

Programmation Web

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

## Téléchargement de fichier

- **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

134

Programmation Web

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

## Téléchargement de fichier

- Variables disponibles dans la page cible
  - `$_FILES["monfichier"]["name"]` : nom original du fichier
  - `$_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>";
```

135

Programmation Web

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

## Transmission de variables d'une page PHP à une autre

- **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'URI de la page cible

136

Programmation Web

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

## Transmission de variables d'une page PHP à une autre

```
<!-- 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>
```

137

Programmation Web

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

## Transmission de variables d'une page PHP à une autre

```
<?php // suivante.php

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

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

?>
```

138

Programmation Web

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

## Sessions

- **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
  - ...

139

Programmation Web

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

## Sessions

```
<!-- Exemple -->
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<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>
```

140

Programmation Web

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

## Sessions

```
<?php
session_start();
// Enregistre la valeur de $nom dans la variable de session s_nom
$_SESSION["s_nom"] = $_POST["nom"];
?>
<!DOCTYPE html>
<html>
<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>
```

141

Programmation Web

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

## Sessions

```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<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">clôre la session</a> ?<p>
</body>
</html>
```

142

Programmation Web

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

## Sessions

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

143

Programmation Web

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

## Sessions

```
<?php
session_start();
session_destroy(); // Destruction de la session
?>
<!DOCTYPE html>
<html>
<head> <title>Session 5</title> </head>
<body>
<p>Session terminée.</p>
</body>
</html>
```

144

Programmation Web

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



## Plan du cours

- ✓ Langage HTML
- ✓ Éléments de design Web
- ✓ Feuilles de style en cascade (CSS)
- ✓ Langage de script PHP
- ✓ Connexion à une base de données (MySQL)

