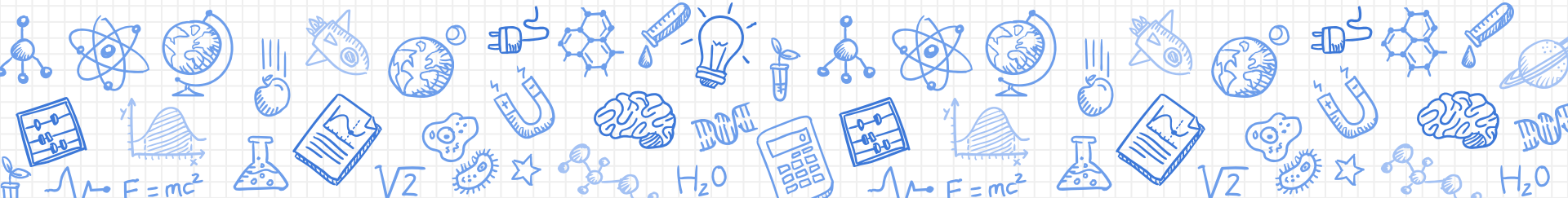
R301 : Dév. web
#1 Site dynamique

Comment fonctionne un échange web ?

C'est *peut-être* des révisions



Bonjour !

Je suis JC Dubacq

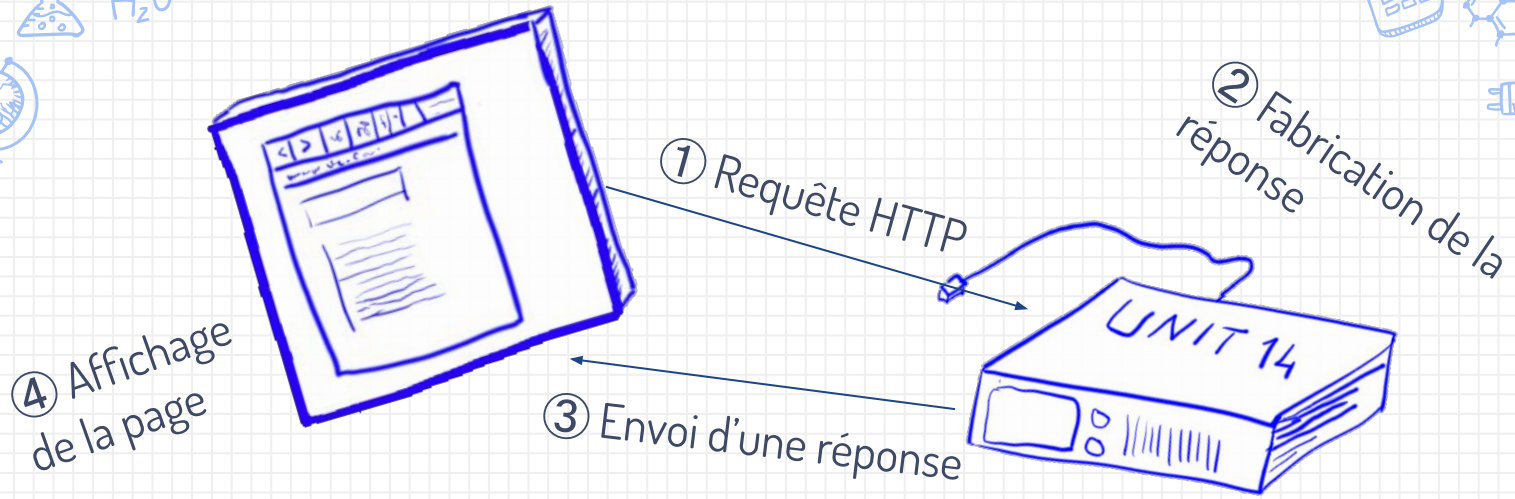
jean-christophe.dubacq@univ... (vous savez la suite)

Je suis le responsable de ce cours, et votre chargé de TD.

Le cours a été écrit essentiellement par un collègue, M. Lacroix.

Le fonctionnement du web

- ✘ Deux systèmes, un dialogue
- ✘ Système qui interroge : *client web*
- ✘ Système qui répond : *serveur web*
- ✘ Au milieu : transport des informations via internet (TCP/IP), parfois relais au niveau applicatif *(serveurs proxys)*
- ✘ Protocole principal : HTTP, variante sécurisée : HTTPS



Requête / Réponse

L'échange se déroule en 4 temps

Vocabulaire HTTP

Requête

GET /contact.php HTTP/1.1

Host: www.example.com

User-Agent: AutoBot 2.34

Accept: */*

Réponse

HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Content-Length: 1234[...]

<html><body><h1>Hello!</h1>[...]

Le protocole HTTP

- ✘ La requête comporte un chemin (+paramètres) et des entêtes.
- ✘ La réponse comporte un statut, des entêtes et un corps (de réponse).
- ✘ Le chemin est déduit à partir de l'URL

URL = protocole + serveur + chemin + paramètres + localisation dans la page

Comment fabriquer la réponse ?

Un logiciel serveur

Démon qui tourne sur le serveur ; **écoute** le port 80/443 ; **délègue** à un thread la réponse ; **détermine** le moyen d'obtenir le corps de la réponse

① Réponse statique

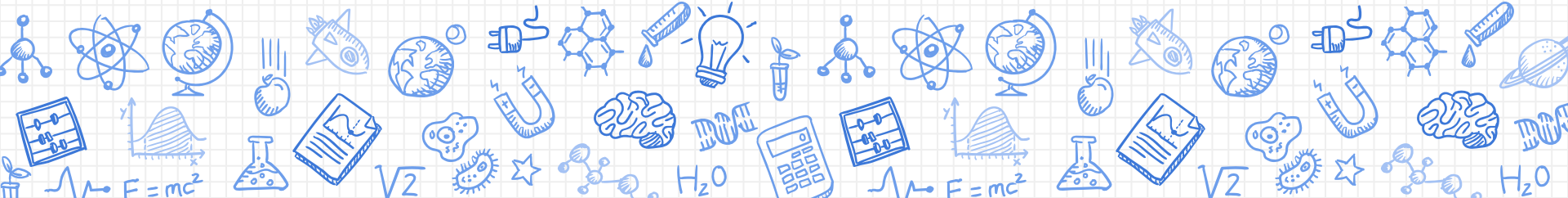
Transforme le chemin de la requête en un chemin d'un *fichier de données* du système de fichiers ; **Identifie** le type MIME du fichier (par extension souvent) ; si le fichier existe, le **renvoie** comme corps

② Réponse dynamique

Transforme le chemin de la requête en un chemin d'un *script* du système de fichiers ; **Exécute** l'interpréteur sur le script ; **récupère** la sortie standard du script ; **renvoie** la sortie comme corps

PHP : interpréteur & langage

On rentre dans le dur



PHP: un interpréteur qui par défaut n'interprète pas

```
<html><head><?php $a="Mon texte"; ?>
  <title><?= $a ?></title>
</head><body><h1><?= $a ?></h1>
<p>Je suis PHP, votre langage. <?php
$t = getdate(); if ($t['hours']<18) { echo "Bonjour"; }
else { echo "Bonsoir"; } ?>!</p>
</body></html>
```

Le contrôle du flux de l'interpréteur

- ✘ La paire <?php ... ?> contient du code interprété ;
  <?= $xxx ?> est un raccourci de <?php echo $xxx; ?>
- ✘ Ce qui n'est pas interprété est transmis sans modification.
- ✘ Les opérateurs de flux ont une syntaxe spécifique pour l'alternance interprétée / non-interprétée
- ✘ Fin du fichier, fin d'interprétation (avec ou sans ?>)
- ✘ require/include permet d'imbriquer les fichiers

Contrôle de flux : syntaxe alternante

```
<?php if ($a == 2): ?>
  <p>La variable vaut bien 2</p>
<?php else: ?>
  <p>La variable n'est pas 2</p>
<?php endif; ?>
```

```
<?php
if ($a == 2) {
  echo "<p>La variable vaut bien 2</p>";
} else {
  echo "<p>La variable n'est pas 2</p>";
} ?>
```

Require ou include ?

toto.php

```
<?php
require "tutu.php";
include "titi.php";
include "tata.php";
```

tututututiti

titi.php

```
<?php
require "tutu.php";
echo "titi";
```

tutu.php

```
<?php
$a = "veryimportant";
echo "tutu";
```

toto2.php

```
<?php
require_once "tutu.php";
include "titi2.php";
```

tututiti

titi2.php

```
<?php
require_once "tutu.php";
echo "titi";
```

toto3.php

```
<?php
require_once "tutuX.php";
include "titi.php";
```

ERROR 500

Comment le script s'arrête ?

Une erreur fatale

- ✘ require ou require_once d'un fichier inexistant
- ✘ Fonction ou classe inexistante

Arrêt immédiat, ajout d'un texte, puis plus rien (code 500 ?)

Une erreur surmontable

Ajout d'un texte de WARNING, et continue le script (exemple : 1/0, ou include d'un fichier non-existant).

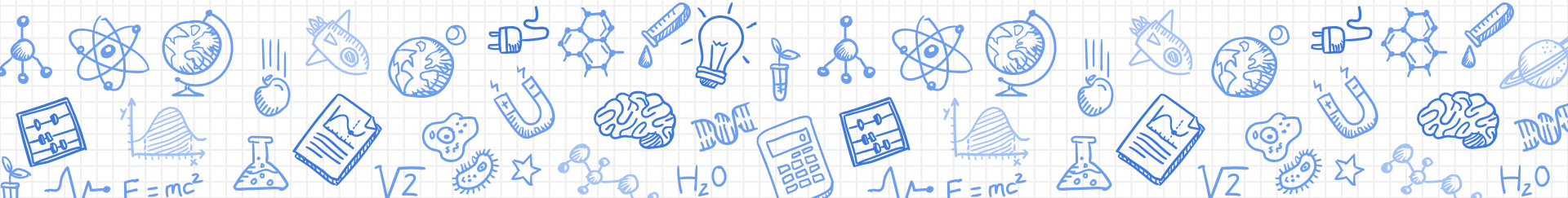
Die("...") ou exit()

Comme une erreur fatale, mais puisque c'est prévu, on peut l'anticiper pour envoyer un code signifiant (500, 401, etc.). Ajoute le texte (pour *die*) et plus rien.

⚠️ Dans tous les cas, une fois du texte écrit, le code ne change plus.

La syntaxe de PHP

Référez-vous au document de cours



Le cours est la référence

⚠️ Au fur et à mesure des exercices qui vous seront proposés, vous devrez lire le document de référence. Les quelques points qui sont ci-après sont des rappels de ce que vous devez avoir révisé à la fin de ce chapitre pour bien aborder les chapitres suivants, avec des exemples qui représentent des tournures parfois complexes.

Les points à connaître

Le contrôle de flux :

- ✘ if (...) {...} elseif (...) {...} else {...}
- ✘ while (...) {...}
- ✘ for (initialisation;continuation;incrémentation) {...}
- ✘ Exemple : for ($i=0;$i<20;$i+=3) {echo " $i ";}

Le contrôle de chaîne :

```
<?php $prenom = "Kevin"; $nom="Dupont";
$message = "Ciao $prenom $nom"; // ou "Ciao ".$prenom." ".$nom
echo $message; ?>
```

Les structures de données : tableaux & tableaux associatifs

- ✘ Tableaux à valeurs hétérogènes
- ✘ Clés : entiers ou chaînes (⚠️ on peut mélanger mais on ne le fait pas : c'est très dangereux)
- ✘ Clés entières, on dit tableau (liste de python)
- ✘ Clés chaînes, on dit tableau associatif (dict de python)
- ✘ Tableau (int) avec pas de clé : clé automatique
- ✘ $a=array(0=>"a", 3=>"b"); $a[]="c" ⇒ $a[4]="c"
- ✘ $b=array("toto"=>"3""titi"=>2) ⇒$b["titi"] est 2
- ✘ foreach ($a as $k) { echo $k // Valeurs successives a, b, c }
- ✘ foreach ($b as $k => $v) { echo $k ; echo $v } // toto 3 titi 2

Les fonctions et les objets

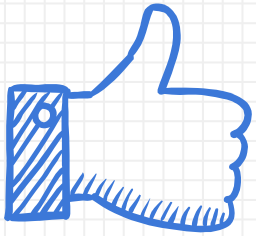
function toto($param1,$param2) { ...return “ok”; }

Variables superglobales ($_GET , $_POST , $_SESSION et $_COOKIE)

```
class Humain {
    public $nom;
    public function __construct(string $nom) {$this->nom = $nom;}
    public function bonjour() { echo "Hello " . $this->nom; }
}
$h1 = new Humain("Alice");
$h1->bonjour(); // Affiche : Hello Alice
```

🛑 Quelques tournures complexes (à comprendre à la fin)

```
<?php $nom = 'Tom ''O\'Malley'; $val=18; $what='4$val'
$message = "Pour '$nom', le prix de '$what' est $val\$US.";
echo $message; ?> ⇒ Pour 'Tom O'Malley', le prix de 4$val est 18$US.
<?php $data = ['client' => ['nom'=> 'Kevin', 'age'=>45],'item'=>'Marteau']
echo $data['client']['nom']; ?>
<?php class Humain {
    public static $salutation = "Hello"; public $nom;
    public function __construct(string $nom) {$this->nom = $nom;}
    public function bonjour() { echo self::$salutation . " " . $this->nom; }
}
```



C'est tout!

Des questions?

- ✘ Dans le Discord
- ✘ Par mail