

# W31

## Programmation web côté serveur

---

Pierre Kraemer

[kraemer@unistra.fr](mailto:kraemer@unistra.fr)

# Client

navigateur web



# Client

navigateur web



Interprète et affiche :

- HTML (structure)
- CSS (style)
- JavaScript (dynamisme / événements)

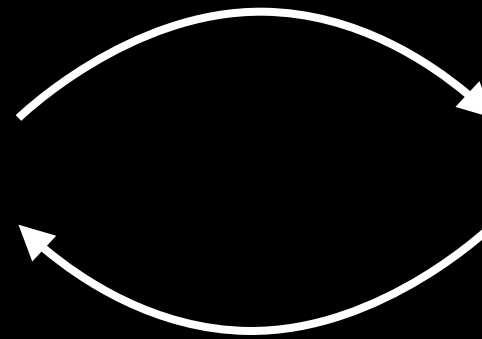
# Client

navigateur web



# Serveur

serveur HTTP + modules



Interprète et affiche :

- HTML (structure)
- CSS (style)
- JavaScript (dynamisme / événements)

Fournit des données au client :

- telles quelles

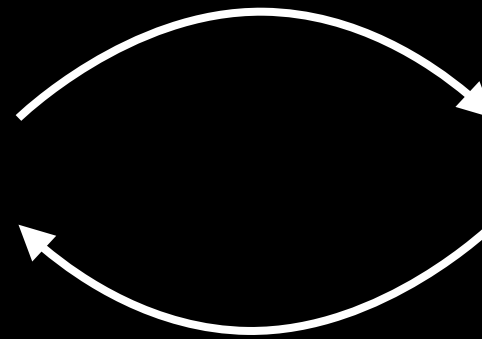
# Client

navigateur web



# Serveur

serveur HTTP + modules



Interprète et affiche :

- HTML (structure)
- CSS (style)
- JavaScript (dynamisme / événements)

Fournit des données au client :

- telles quelles
- générées dynamiquement
  - ➔ Java (JSP), Python (Django), Ruby (RoR), JS (Node.js), PHP ...

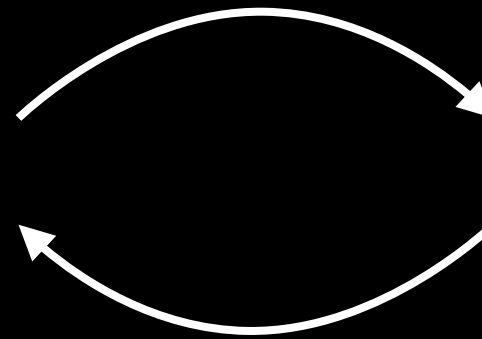
# Client

navigateur web



# Serveur

serveur HTTP + modules



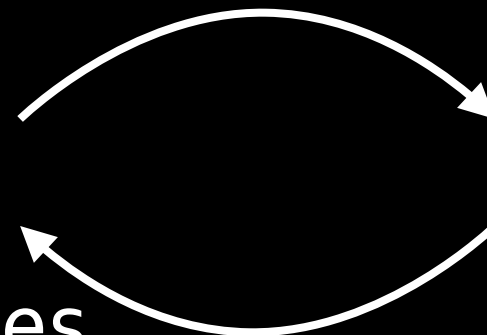
Fournit des données au client :

- telles quelles
- générées dynamiquement



Système de Gestion de  
Base de Données

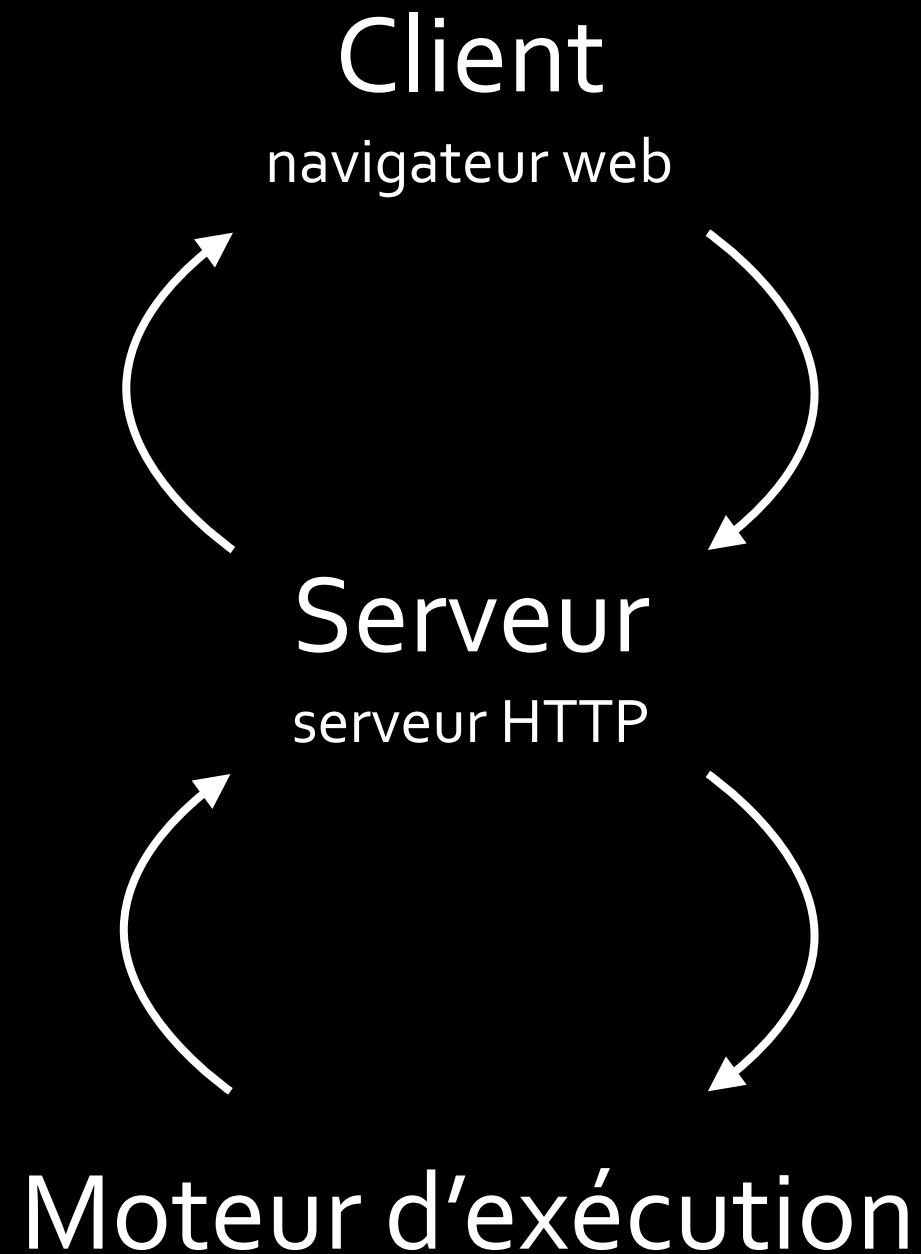
Persistance des données



→ Java (JSP), Python  
(Django), Ruby (RoR),  
JS (Node.js), PHP ...

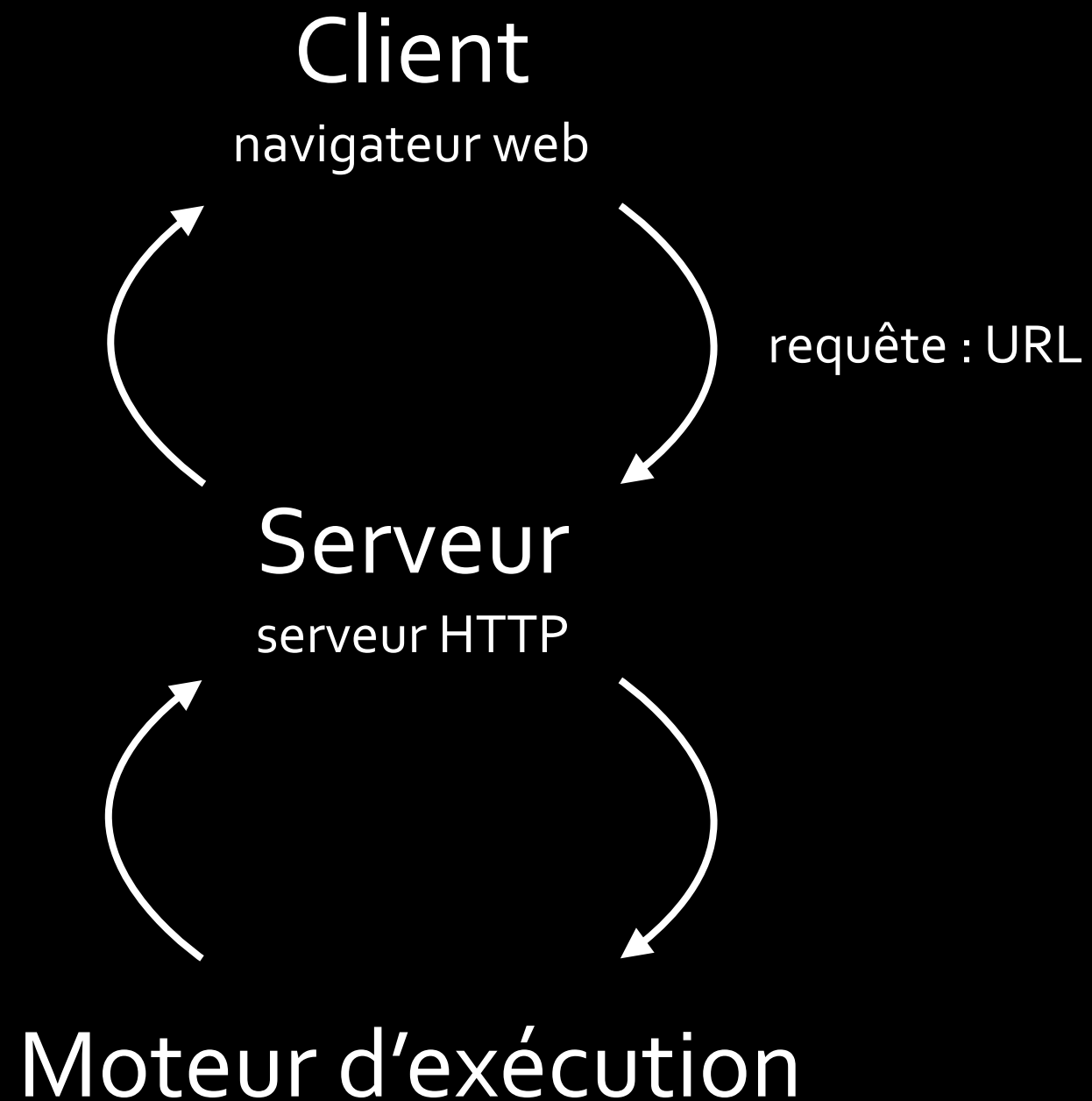
# Fonctionnement général

---



# Fonctionnement général

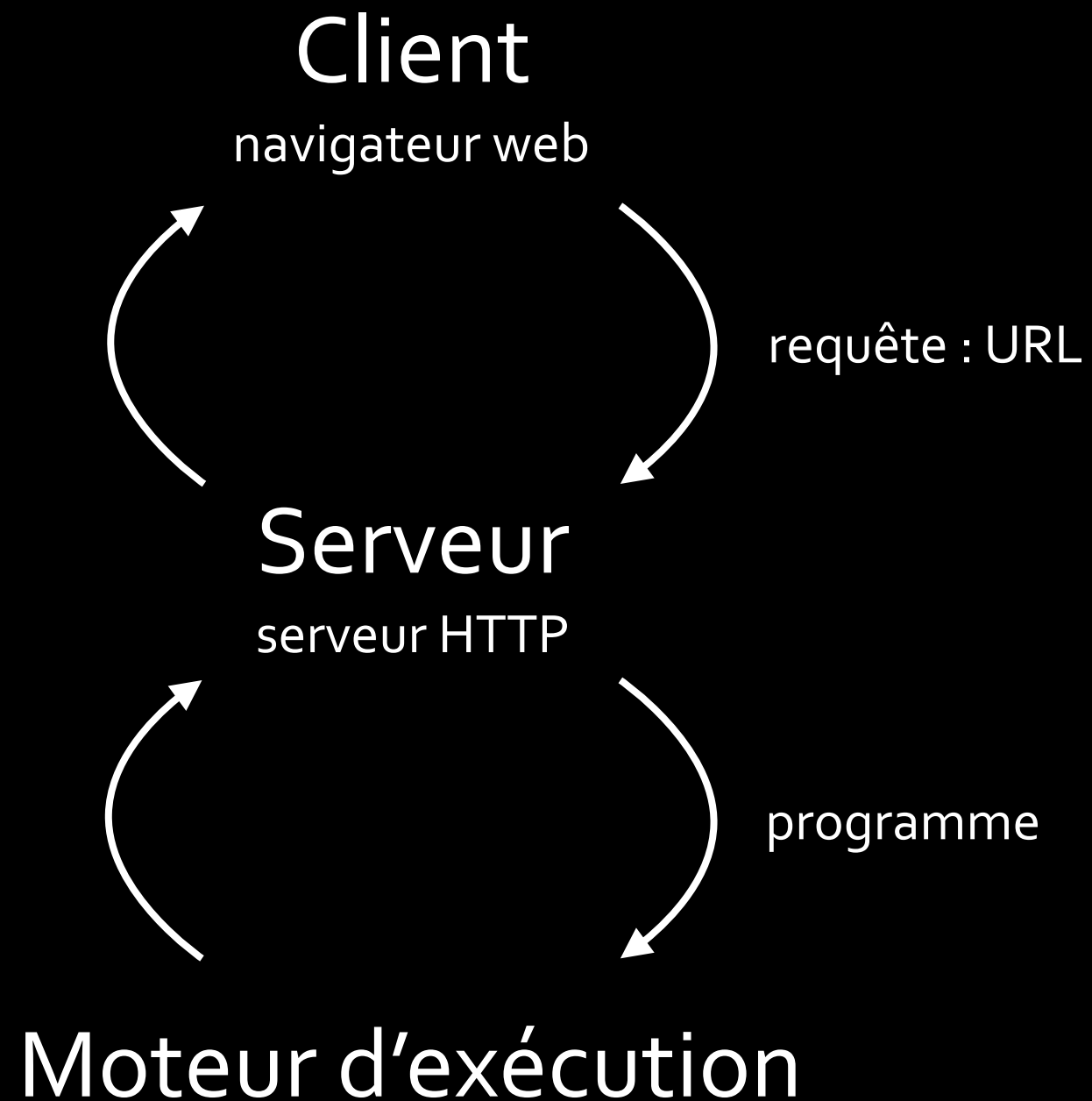
---





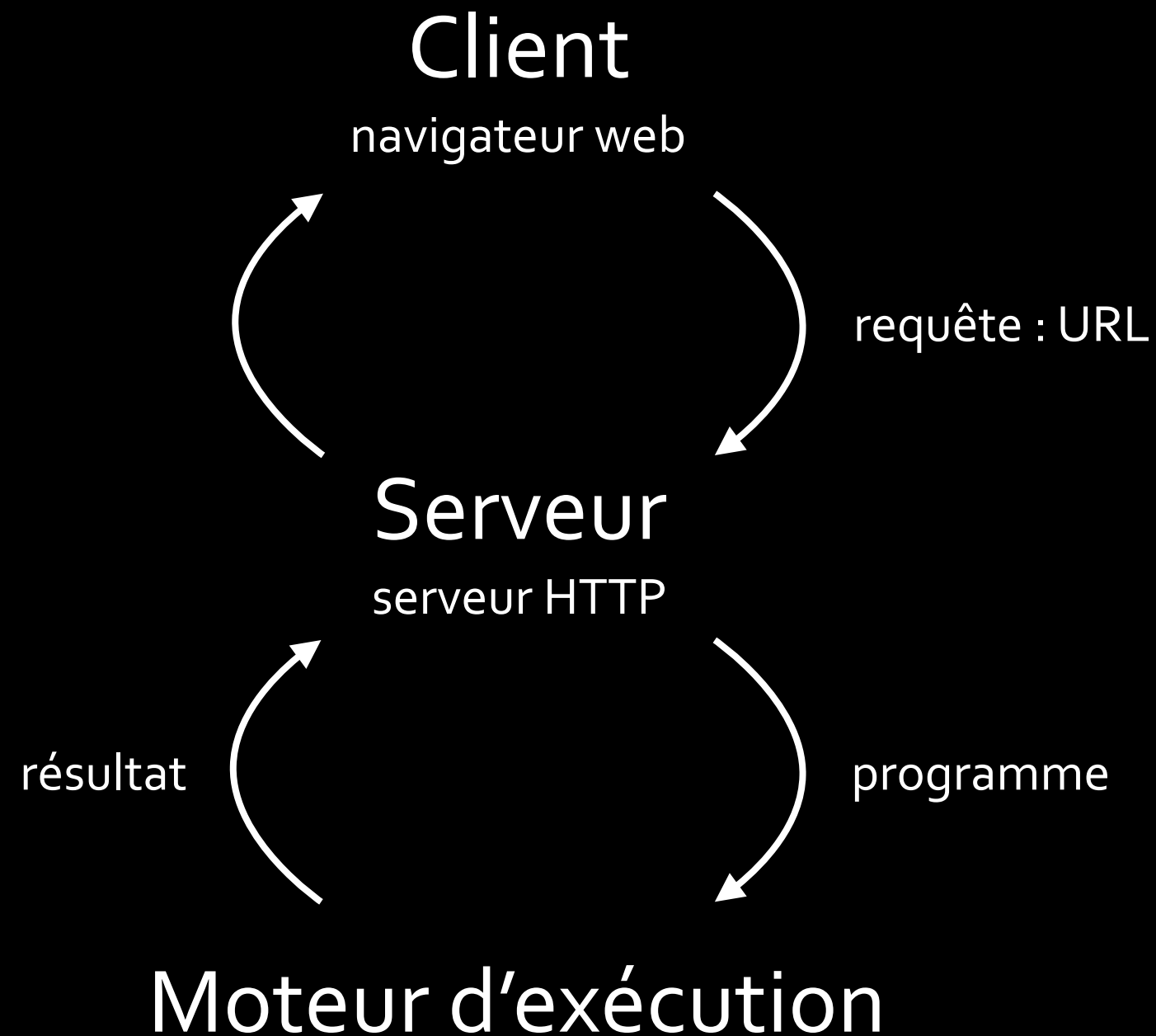
# Fonctionnement général

---



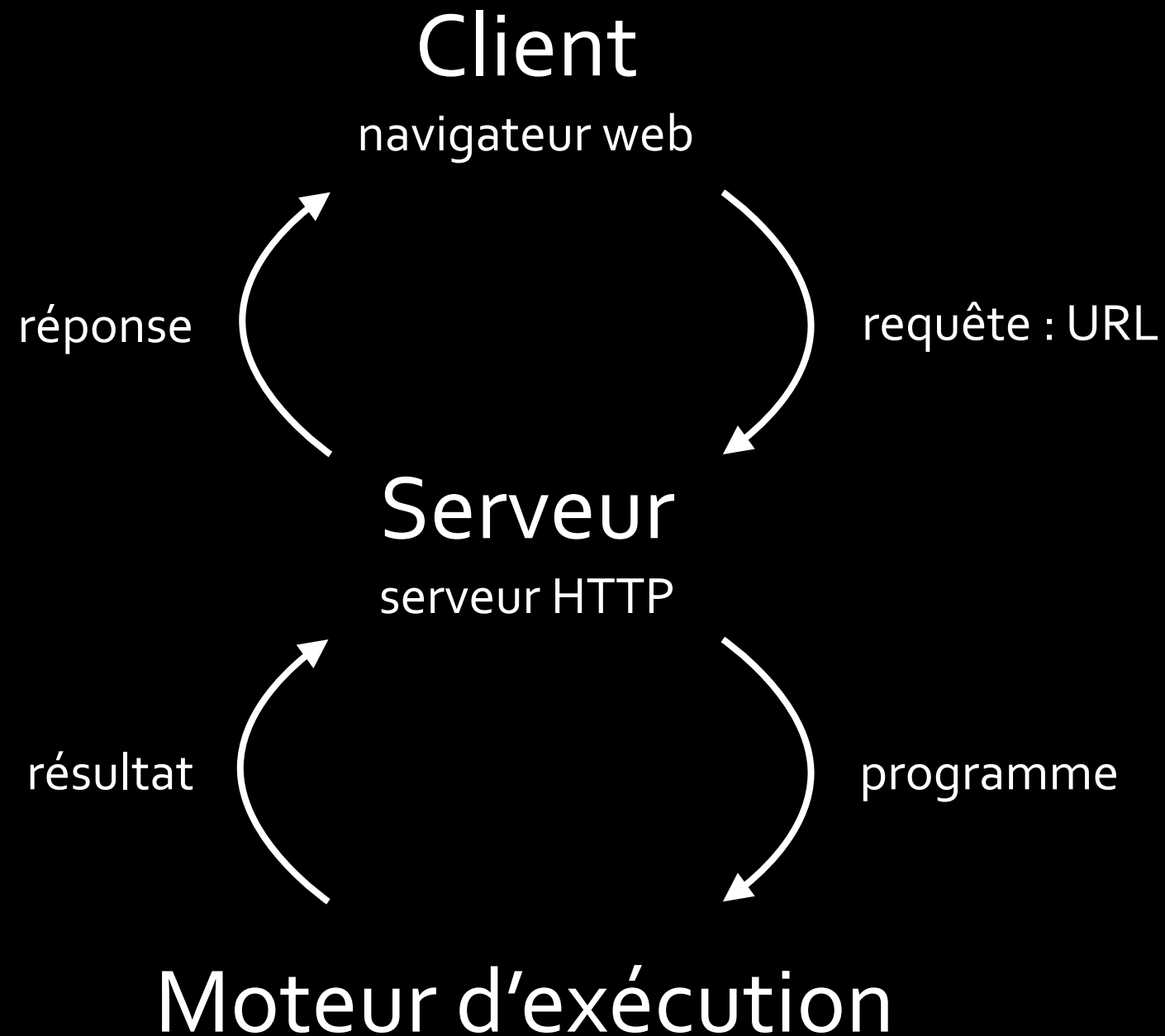
# Fonctionnement général

---



# Fonctionnement général

---



# PHP

---

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title> PHP </title>
```

```
</head>
```

```
<body>
```

```
    <p> <?php echo 'Youpi !' ?> </p>
```

```
</body>
```

```
</html>
```

# Bases du langage

---

```
<?php
```

```
// commentaire sur une ligne
```

```
/*
```

```
commentaire sur
```

```
plusieurs
```

```
lignes
```

```
*/
```

```
$a = 5; // déclaration et affectation
```

```
$b = 8; // de variables
```

```
$res = $a + $b; // opérateurs
```

```
$res *= $b - $a; // arithmétiques
```

```
$res += $b % $a; // classiques
```

```
// chaînes de caractères
```

```
$message = 'calcul sur les valeurs : ' . $a . ' et ' . $b;
```

```
// conditionnelle
```

# Bases du langage

---

```
$res = $a + $b; // opérateurs
$res *= $b - $a; // arithmétiques
$res += $b % $a; // classiques

// chaînes de caractères
$message = 'calcul sur les valeurs : ' . $a . ' et ' . $b;

// conditionnelle
if ($res == 42) {
    $reponse = 'succès';
}

// test d'existence d'une variable
if(isset($reponse)) {
    echo $message . ' -> ' . $reponse; // ajout d'une chaîne
                                        // au résultat
}

// type d'une variable ?
echo gettype($a); // 'integer'
echo gettype($message); // 'string'
echo gettype(3.14); // 'double'
```

# Bases du langage

---

```
// type d'une variable ?
echo gettype($a); // 'integer'
echo gettype($message); // 'string'
echo gettype(3.14); // 'double'

// test sur les types
if (is_int($a) && is_string($reponse)) {
    echo 'c\'est OK';
}

define('NOM_MODULE', 'P31'); // définition d'une constante

// quelques constantes magiques
echo __LINE__; // numéro de ligne courante
echo __FILE__; // nom de fichier courant
echo __DIR__; // nom du répertoire courant

// définition d'une fonction
function accueil($u) {
    echo 'Bonjour ' . $u;
    echo 'Bienvenue en ' . NOM_MODULE;
}
```

# Bases du langage

---

```
// définition d'une fonction
function accueil($u) {
    echo 'Bonjour ' . $u;
    echo 'Bienvenue en ' . NOM_MODULE;
}

accueil('Pierre'); // appel de la fonction

$tab = array(1,2,3); // déclaration d'un tableau

// parcours d'un tableau

for ($i = 0; $i < count($tab); ++$i) {
    echo 'case ' . $i . ' -> ' . $tab[$i];
}

foreach ($tab as $value) {
    echo $value;
}

foreach ($tab as $key => $value) {
    echo 'case ' . $key . ' -> ' . $value;
}
```



# Bases du langage

---

```
foreach ($tab as $key => $value) {  
    echo 'case ' . $key . ' -> ' . $value;  
}
```

```
$tab[1] = 4; // affectation dans la case d'indice 1  
$tab[] = 12; // ajout d'une valeur en fin de tableau
```

```
print_r($tab); // affiche le contenu du tableau ([1,4,3,12])
```

```
// tableaux associatifs
```

```
$tab = array(  
    'France' => 'Paris',  
    'Allemagne' => 'Berlin',  
    'Italie' => 'Milan',  
    'Espagne' => 'Madrid',  
    'Belgique' => 'Bruxelles'  
);
```

```
$tab['Italie'] = 'Rome';
```

```
if (!function_exists('startsWithVowel')) {  
    function startsWithVowel($a) {
```

# Bases du langage

---

```
        'Belgique' => 'Bruxelles'
    );

$tab['Italie'] = 'Rome';

if (!function_exists('startsWithVowel')) {
    function startsWithVowel($s) {
        $vowels = array('a','e','i','o','u','y');
        $firstLetter = strtolower(substr($s,0,1));
        return in_array($firstLetter, $vowels);
    }
}

foreach ($tab as $key => $value) {
    $res = $value . ' est la capitale de ';
    if(startsWithVowel($key))
        $res .= "l' " . $key;
    else
        $res .= 'la ' . $key;
}

?>
```

# Inclusions

---

header

navigation

content

footer

# Inclusions

---

header

navigation

content

footer

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
    <meta charset="UTF-8" />
```

```
    <title> PHP </title>
```

```
</head>
```

```
<body>
```

```
    <?php include 'header.php'; ?>
```

```
    <?php include 'navigation.php'; ?>
```

```
    <section>
```

```
        Mon contenu très intéressant..
```

```
    </section>
```

```
    <?php include 'footer.php'; ?>
```

```
</body>
```

```
</html>
```

# Transmettre des données

---

<http://www.website.fr/index.php>

# Transmettre des données

---

`http://www.website.fr/index.php?param1=val1&param2=val2`

# Transmettre des données

---

http://www.website.fr/index.php?param1=val1&param2=val2

```
<!DOCTYPE html>

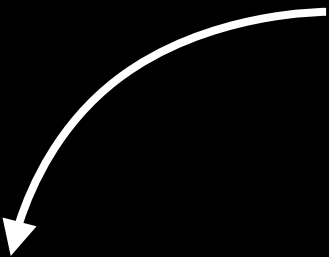
<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p> Premier paramètre : <?php echo $_GET['param1']; ?> </p>
    <p> Deuxième paramètre : <?php echo $_GET['param2']; ?> </p>
</body>

</html>
```

tableau  
global



# Transmettre des données

---

```
<!DOCTYPE html>

<html lang="fr">

<head>
  <meta charset="UTF-8">
  <title> PHP </title>
</head>

<body>
  <form method="post" action="target.php">
    <input type="text" name="nom">
    <select name="fruit">
      <option value="pomme"> Pomme </option>
      <option value="banane"> Banane </option>
      <option value="framboise"> Framboise </option>
    </select>
    <input type="number" name="quantite">
    <input type="submit">
  </form>
</body>

</html>
```



# Transmettre des données

---

```
        <option value="framboise"> Framboise </option>
    </select>
    <input type="number" name="quantite">
    <input type="submit">
</form>
</body>

</html>
```

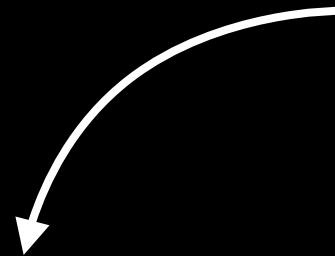
"target.php"

```
<?php
```

```
echo 'Bonjour ' . $_POST['nom'] . ' !';
$s = $_POST['quantite'] > 1 ? 's' : '';
echo 'Vous avez commandé ' . $_POST['quantite'] . ' ' . $_POST['fruit'] . $s;

?>
```

tableau  
global



# Transmettre des données

---



*Principe général :*  
ne **jamais** faire confiance aux données reçues

# Transmettre des données

---



*Principe général :*

ne **jamais** faire confiance aux données reçues

- Vérifier l'existence

```
if (isset($_GET[param1]) && isset($_GET[param2])) {  
    ...  
}
```

# Transmettre des données

---



*Principe général :*

ne **jamais** faire confiance aux données reçues

- Vérifier l'existence

```
if (isset($_GET[param1]) && isset($_GET[param2])) {  
    ...  
}
```

- Vérifier le type

```
$val = (int) $_GET[param1]; // si on attend un entier
```

# Transmettre des données

---



*Principe général :*

ne **jamais** faire confiance aux données reçues

- Vérifier l'existence

```
if (isset($_GET[param1]) && isset($_GET[param2])) {  
    ...  
}
```

- Vérifier le type

```
$val = (int) $_GET[param1]; // si on attend un entier
```

- Vérifier la valeur

```
if ($val < 10) { // dépend de ce que l'on fait  
    ...        // avec les données  
}
```

# Transmettre des données



```
<script> alert('héhé'); </script>
```

Pomme



Submit

```
<?php
```

```
echo 'Bonjour ' . $_POST['nom'] . ' !';
```

```
$s = $_POST['quantite'] > 0 ? 's' : '';
```

```
echo 'Vous avez commandé ' . $_POST['quantite'] . ' ' . $_POST['fruit'] . $s;
```

```
?>
```

# Transmettre des données



```
<script> alert('héhé'); </script>
```

Pomme



Submit

→ exécution du script  
chez le client, dans la  
page générée

Faible XSS  
(cross-site scripting)

```
<?php
```

```
echo 'Bonjour ' . $_POST['nom'] . ' !';
```

```
$s = $_POST['quantite'] > 0 ? 's' : '';
```

```
echo 'Vous avez commandé ' . $_POST['quantite'] . ' ' . $_POST['fruit'] . $s;
```

```
?>
```

# Transmettre des données



`<script> alert('héhé'); </script>`

Pomme

Submit

→ exécution du script  
chez le client, dans la  
page générée

Faible XSS  
(cross-site scripting)

- Echapper les balises HTML

```
<?php
```

```
$n = isset($_POST['nom']) ? htmlspecialchars($_POST['nom']) : '';  
$f = isset($_POST['fruit']) ? htmlspecialchars($_POST['fruit']) : '';  
$q = isset($_POST['quantite']) ? (int) $_POST['quantite'] : 0;
```

```
echo 'Bonjour ' . $n . ' !';
```

```
$s = $q > 0 ? 's' : '';
```

```
echo 'Vous avez commandé ' . $q . ' ' . $f . $s;
```

```
?>
```



# Transmettre des données



Pomme

▼

Submit

exécution du script  
chez le client, dans la  
page générée

Faible XSS  
(cross-site scripting)

- Echapper les balises HTML

```
<?php
```

```
$n = isset($_POST['nom']) ? htmlspecialchars($_POST['nom']) : '';  
$f = isset($_POST['fruit']) ? htmlspecialchars($_POST['fruit']) : '';  
$q = isset($_POST['quantite']) ? (int) $_POST['quantite'] : 0;
```

```
echo 'Bonjour ' . $n . ' !';
```

```
$s = $q > 0 ? 's' : '';
```

```
echo 'Vous avez commandé ' . $q . ' ' . $f . $s;
```

```
?>
```

```
&lt;script&gt; alert('héhé'); &lt;/script&gt;
```

# Conserver des données

---

page1.php

```
<?php
setcookie(
    'nb', '42',
    time() + 60*60*24*10,
    null, null, false, true);
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

# Conserver des données

page1.php

```
<?php
setcookie(
    'nb', '42',
    time() + 60*60*24*10,
    null, null, false, true);
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

page2.php

```
<!DOCTYPE html>

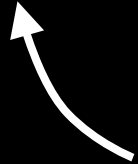
<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est :
        <?php
            $n = htmlspecialchars(
                $_COOKIE['nb']
            );
            echo $n;
        ?>
    </p>
</body>

</html>
```

tableau global



# Conserver des données

page1.php

```
<?php
setcookie(
    'nb', '42',
    time() + 60*60*24*10,
    null, null, false, true);
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

à appeler  
avant tout  
code HTML

page2.php

```
<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est :
        <?php
            $n = htmlspecialchars(
                $_COOKIE['nb']
            );
            echo $n;
        ?>
    </p>
</body>

</html>
```

tableau  
global

# Conserver des données

page1.php

```
<?php
setcookie(
    'nb', '42',
    time() + 60*60*24*10,
    null, null, false, true);
?>
```

```
<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

à appeler  
avant tout  
code HTML

htmlOnly :  
pas d'accès  
côté client  
(JS)

page2.php

```
<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est :
        <?php
            $n = htmlspecialchars(
                $_COOKIE['nb']
            );
            echo $n;
        ?>
    </p>
</body>

</html>
```

tableau  
global

# Conserver des données

page1.php

```
<?php
setcookie(
    'nb', '42',
    time() + 60*60*24*10,
    null, null, false, true);
?>
```

```
<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

à appeler  
avant tout  
code HTML

htmlOnly :  
pas d'accès  
côté client  
(JS)

page2.php

```
<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est :
        <?php
            $n = htmlspecialchars(
                $_COOKIE['nb']
            );
            echo $n;
        ?>
    </p>
</body>

</html>
```

provient du  
client :  
pas confiance

tableau  
global

# Conserver des données

---

Cookies :

- données stockées chez le client
- envoyées au serveur à chaque requête
- durée de vie spécifiée à la création
- destruction automatique à expiration  
(peut-être forcé par une mise-à-jour avec une date d'expiration antérieure à la date courante)

# Conserver des données

---

Cookies :

- données stockées chez le client
- envoyées au serveur à chaque requête
- durée de vie spécifiée à la création
- destruction automatique à expiration  
(peut-être forcé par une mise-à-jour avec une date d'expiration antérieure à la date courante)

à filtrer !





# Conserver des données

---

page1.php

```
<?php
session_start();

$_SESSION['nb'] = 42;
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

# Conserver des données

page1.php

```
<?php
session_start();

$_SESSION['nb'] = 42;
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

page2.php

```
<?php
session_start();
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est : 
        <?php echo $_SESSION['nb'] ?>
    </p>
</body>

</html>
```

tableau  
global



# Conserver des données

page1.php

```
<?php
session_start();

$_SESSION['nb'] = 42;
?>

<!DOCTYPE html>

<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <a href="page2.php">Go</a>
</body>

</html>
```

à appeler  
avant tout  
code HTML

page2.php

```
<?php
session_start();
?>

<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8" />
    <title> PHP </title>
</head>

<body>
    <p>
        Le nombre est : 
        <?php echo $_SESSION['nb'] ?>
    </p>
</body>

</html>
```

tableau  
global

# Conserver des données

---

Sessions :

- crée un cookie contenant un identifiant unique
- données stockées sur le serveur
- court terme :
  - détruit automatiquement après un certain temps
  - détruit explicitement par un appel à `session_destroy()`;

# En-têtes HTTP (headers)

---

HTTP/1.1 200 OK

**Date:** Wed, 01 Aug 2012 15:58:13 GMT

**Server:** Apache/2.2.22 (Debian)

**Last-Modified:** Thu, 05 Jul 2012 07:10:02 GMT

**ETag:** "22c73-b1-4c4ofd6220544"

**Accept-Ranges:** bytes

**Content-Length:** 177

**Vary:** Accept-Encoding

**Connection:** close

**Content-Type:** text/html

```
<html><body><h1>It works!</h1>
```

```
<p>This is the default web page for this server.</p>
```

```
<p>The web server software is running but no  
content has been added, yet.</p>
```

```
</body></html>
```

# En-têtes HTTP (headers)

---

```
HTTP/1.1 200 OK
Date: Wed, 01 Aug 2012 15:58:13 GMT
Server: Apache/2.2.22 (Debian)
Last-Modified: Thu, 05 Jul 2012 07:10:02 GMT
ETag: "22c73-b1-4c4ofd6220544"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

→ Headers

```
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no
content has been added, yet.</p>
</body></html>
```

# En-têtes HTTP (headers)

---

```
HTTP/1.1 200 OK
Date: Wed, 01 Aug 2012 15:58:13 GMT
Server: Apache/2.2.22 (Debian)
Last-Modified: Thu, 05 Jul 2012 07:10:02 GMT
ETag: "22c73-b1-4c4ofd6220544"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

→ Headers

```
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no
content has been added, yet.</p>
</body></html>
```

→ Content

# En-têtes HTTP (headers)

---

```
setcookie(...);  
session_start();
```



génèrent des headers



doivent avoir lieu avant l'envoi  
de tout contenu



# En-têtes HTTP (headers)

---

```
setcookie(...);  
session_start();
```



génèrent des headers



doivent avoir lieu avant l'envoi  
de tout contenu

Des headers par défaut sont générés par  
PHP et le serveur HTTP

# En-têtes HTTP (headers)

---

```
setcookie(...);  
session_start();
```



génèrent des headers



doivent avoir lieu avant l'envoi  
de tout contenu

Des headers par défaut sont générés par  
PHP et le serveur HTTP

`header();` → permet de définir ou redéfinir des headers

```
// redéfinit le code de réponse HTTP  
header('HTTP/1.1 403 Forbidden');
```

```
// redéfinit le type de contenu envoyé  
header('Content-Type: image/jpeg');  
header('Content-Type: application/pdf');
```

```
// demande de redirection (avec un code HTTP 302 Redirect)  
header('Location: http://www.google.fr');
```