

Cours 1 : intro à la prog web et mobile



Web — Applications web et mobile

Christophe Gonzales

Choix des technologies

Contexte historique du web



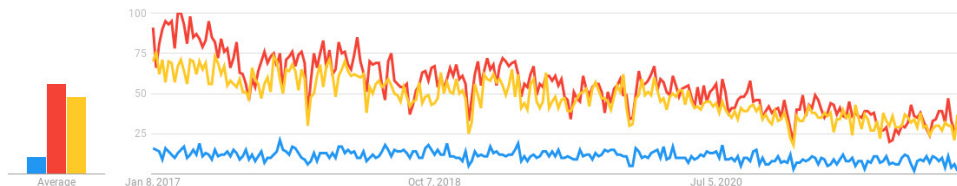
Tendances (1/2)

- Tendances de requêtes google sur Angular/PHP/Javascript sur les 5 dernières années (source : Google trends) :

Interest over time ?

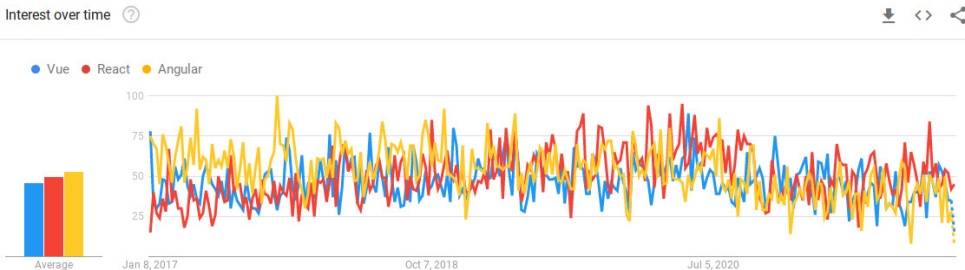


Angular php javascript



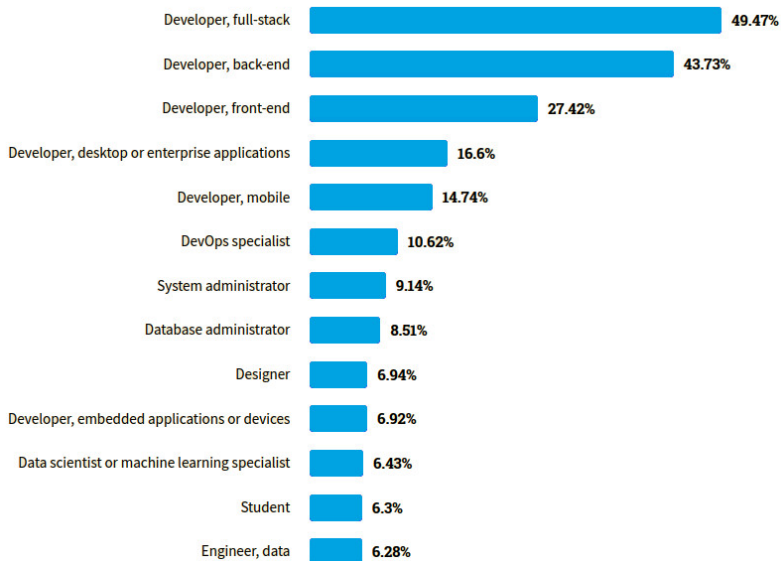
Tendances (2/2)

- Tendances de requêtes google sur Vue/React/Angular sur les 5 dernières années (source : Google trends) :



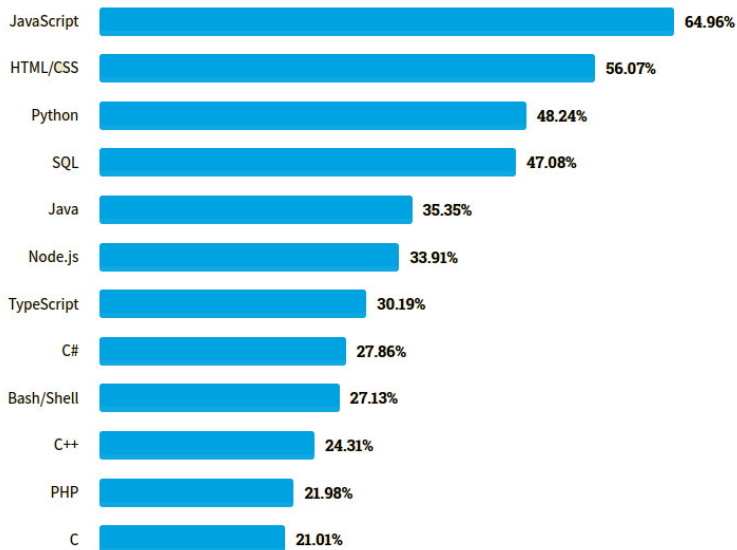
Types de développeurs

► Étude de StackOverflow de 2021 :



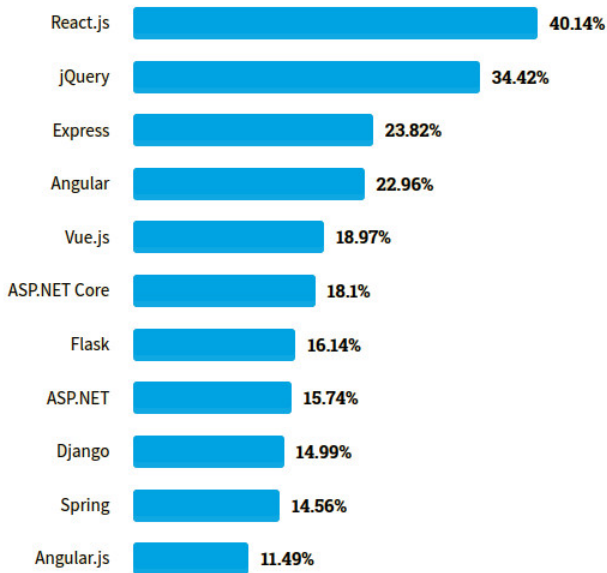
Langages de programmation utilisés

► Étude de StackOverflow de 2021 :




Technologies du web

► Étude de StackOverflow de 2021 :



Quelle technologie connaître/choisir ?

 : WordPress, Drupal, Joomla
80% des sites web dans le monde

 : Facebook, Twitter, Netflix, Instagram, Airbnb, Yahoo Mail

 : Paypal, GMail, Netflix, Lego, Weather, Delta

 : Nintendo, Gitlab, Alibaba, Xiaomi



stable

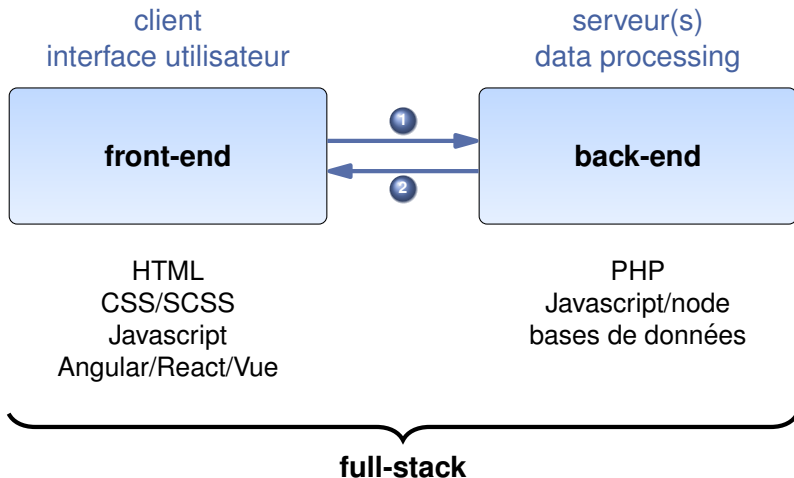


update majeure
tous les 6 mois



API 90% similaire
entre la V1 et la V2

Un peu d'architecture...



Objectif : développement full-stack / mobile

Plan du module :

- 1 HTML/CSS/PHP/Javascript
- 2 Angular/TypeScript
- 3 Node/Express
- 4 Ionic

Objectif des TP

Forum Polytech

Tous les cours / Programmation Objet

Nouveau sujet

Sujet	#posts	Dernier message
Est-ce que le C++ contient le langage C ?	1	le 14/01/2021 à 15:28:02
héritage multiple	4	le 06/01/2021 à 19:03:57
Est-ce que le C++ est un langage objet ?	2	le 06/01/2021 à 18:52:13
qu'est-ce que le polymorphisme ?	0	--

Items per page: 5 1 - 4 of 4 |< < > >|

Galaxy S9/S... 360 x 740 DPR: 4 No Throttling

← Forum: topics

NOUVEAU SUJET

Est-ce que le C++ contient le langage C ?
le 14/01/2021 à 15:28:02 1 post

héritage multiple
le 06/01/2021 à 19:03:57 4 posts

Est-ce que le C++ est un langage objet ?
le 06/01/2021 à 18:52:13 2 posts

qu'est-ce que le polymorphisme ?
--

► <https://christophe-gonzales.pedaweb.univ-amu.fr/forum>

► <https://christophe-gonzales.pedaweb.univ-amu.fr/forum-ionic>

Technologies : PHP, Angular, node/express, ionic

Rappels HTML/CSS

Rappels élémentaires de HTML

```
<!doctype html>
<html class="no-js" lang="">

<head>
  <meta charset="utf-8">
  <title></title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/main.css">
</head>

<body>
  <p>Hello world</p>

  <script src="js/vendor/modernizr-3.8.0.min.js"></script>
  <script src="https://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMG"
  <script>window.jQuery || document.write('<script src="js/vendor/jquery-3.4.1.min.js"></script>')</script>
  <script src="js/plugins.js"></script>
  <script src="js/main.js"></script>

  <!-- Google Analytics: change UA-XXXXX-Y to be your site's ID. -->
  <script>
    window.ga = function () { ga.q.push(arguments) }; ga.q = []; ga.l = +new Date;
    ga('create', 'UA-XXXXX-Y', 'auto'); ga('set','transport','beacon'); ga('send', 'pageview')
  </script>
  <script src="https://www.google-analytics.com/analytics.js" async></script>
</body>
</html>
```

HTML 5

header

balises

corps

Balises

► En général :

`<balise>...</balise>` ou

`<balise attribut1="val1" attribut2="val2">...</balise>`

► Quelques exceptions :

`<balise/>` ou

`<balise attribut1="val1" attribut2="val2" />`

► Exemples :

Balise	Signification
<code><p></p></code>	paragraphe
<code><div></div></code>	conteneur de « division »
<code></code>	conteneur « inline »
<code>, , </code>	listes
<code></code>	images
<code>
</code>	retour à la ligne

Attributs

- ▶ **class** : classe \Rightarrow permet de manipuler l'élément via CSS ou Javascript (sélecteur)
- ▶ **id** : identifiant unique \Rightarrow permet de manipuler l'élément via CSS ou Javascript (sélecteur)

▶ Exemple :

HTML

```
<h1>Section 1 :</h1>
<div class="mydiv">
Hello
<span>promo</span>
<span id="myspan">3A</span>
</div>
```

CSS

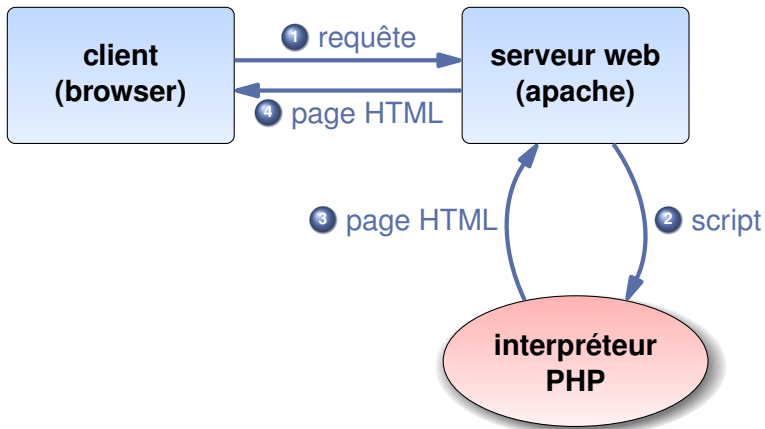
```
div.mydiv { text-align: right; }
div span { color: red; }
#myspan {
  color: blue;
  font-weight: bold;
}
```

Section 1 :

Hello promo 3A

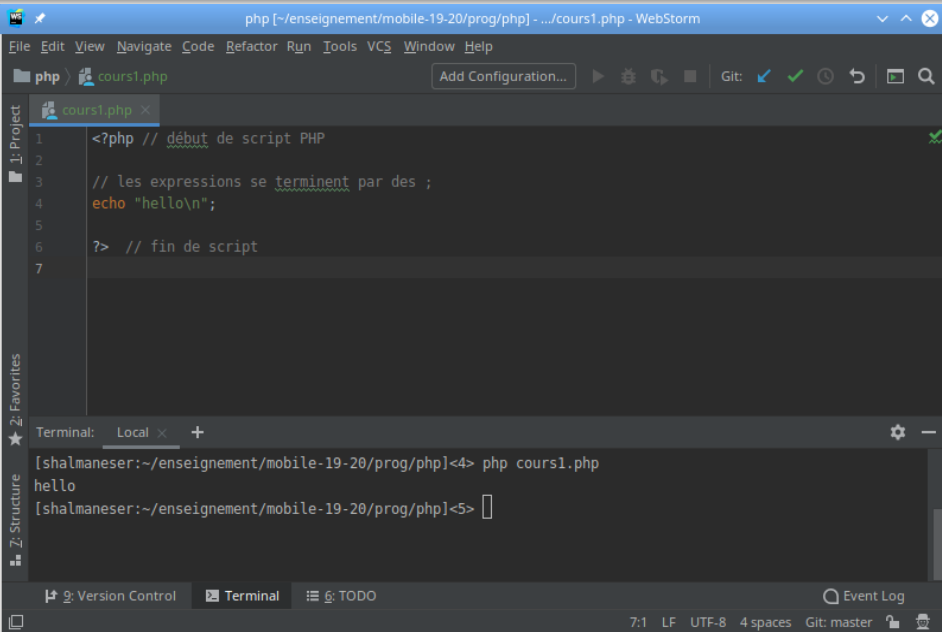
Introduction à PHP

Architecture autour de PHP

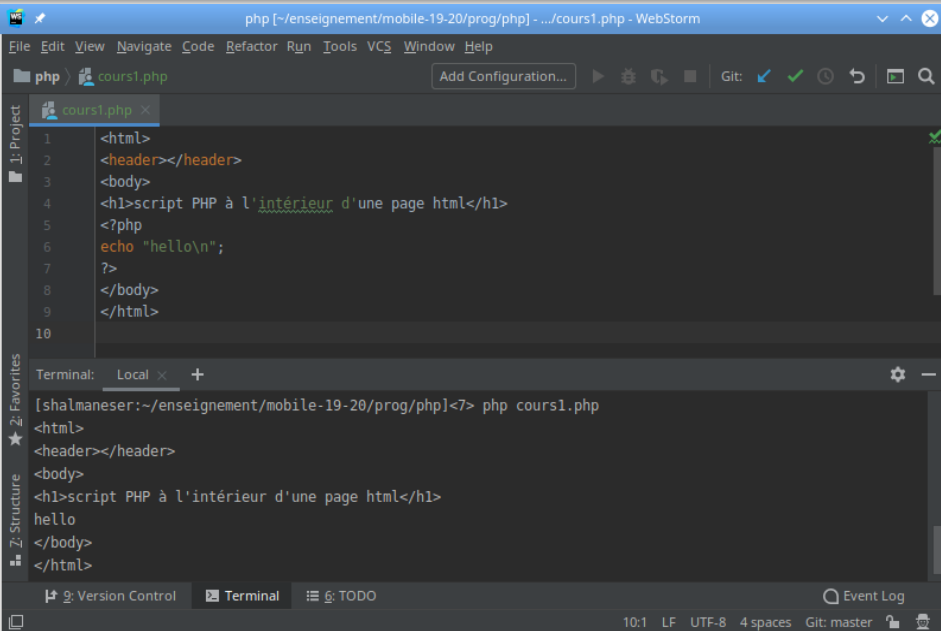


⇒ But : les scripts PHP produisent du texte HTML !
ils sont aussi utiles pour produire des données JSON

Scripts PHP



PHP à l'intérieur de balises HTML



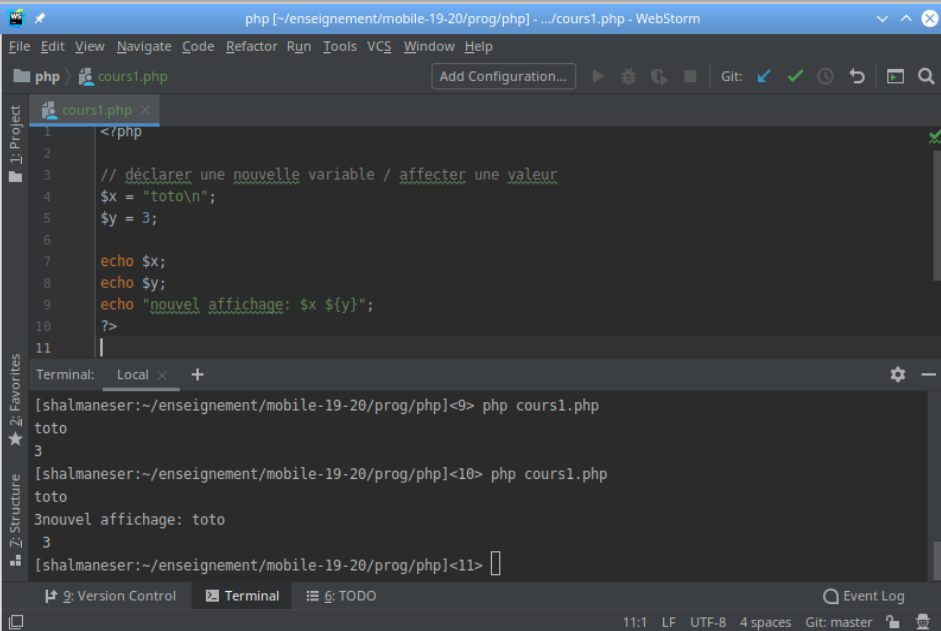
The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project explorer on the left shows the file structure. The main editor displays the content of `cours1.php`, which contains an HTML document with a PHP script. The terminal window at the bottom shows the command `php cours1.php` being executed, resulting in the output `hello`.

```
1 <html>
2 <header></header>
3 <body>
4 <h1>script PHP à l'intérieur d'une page html</h1>
5 <?php
6 echo "hello\n";
7 ?>
8 </body>
9 </html>
10
```

Terminal: Local x +

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<7> php cours1.php
<html>
<header></header>
<body>
<h1>script PHP à l'intérieur d'une page html</h1>
hello
</body>
</html>
```

Déclarer des variables/affecter des valeurs



The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the toolbar, the project structure on the left shows '1: Project' and '2: Favorites'. The main editor displays the file 'cours1.php' with the following PHP code:

```
1 </php
2
3 // déclarer une nouvelle variable / affecter une valeur
4 $x = "toto\n";
5 $y = 3;
6
7 echo $x;
8 echo $y;
9 echo "nouvel affichage: $x ${y}";
10 ?>
11 |
```

Below the editor is a terminal window titled 'Terminal: Local'. It shows the execution of the PHP file:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<9> php cours1.php
toto
3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<10> php cours1.php
toto
3nouvel affichage: toto
3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<11> |
```

The bottom status bar shows '11:1 LF UTF-8 4 spaces Git: master' and an 'Event Log' icon.

Les chaînes de caractères

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: / > home > gonzaes > enseignement > m. The active file is string.php. The code editor displays the following PHP code:

```
1 <?php
2 $x = 3;
3
4 // chaînes de caractères interprétées (entre guillemets) :
5 $y = "toto $x\n";
6 // chaînes non interprétées (entre quotes) :
7 $z = 'toto $x\n';
8
9 echo $y;
10 echo $z;
11 echo $x . $x; // '.' => concaténation de chaînes de caractères
12 ?>
13
```

Below the code editor is a terminal window with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/angular/cours3]<1> cd ../../php
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<2> php string.php
toto 3
toto $x\n33
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<3>
```

The bottom status bar shows various settings: 6 TODO, 9 Version Control, TypeScript 3.5.3, Terminal, 2 Event Log, and a message: "External file changes sync may be slow: The current Inotify(7) watch limit is ... (3 minutes ago) 10:9 LF UTF-8 4 spaces Git: master".

Les nombres

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the toolbar is a breadcrumb path: php > cours1.php. The main editor area displays the contents of 'cours1.php' with line numbers 1 through 17. The code is as follows:

```
1 <?php
2 $x = 3;    // ceci est un entier
3 $y = 4.65; // ceci est un flottant
4
5 echo is_int($x);
6 echo "=====1";
7 echo is_int($y);
8 echo "#####1";
9 echo is_float($y);
10 ?>
11
12
```

Below the editor is a terminal window titled 'Terminal: Local'. It shows the command 'php cours1.php' being executed, resulting in the output '1=====1#####1'.

The bottom status bar displays '11:1 LF UTF-8 4 spaces Git: master' and an 'Event Log' icon.

Les tableaux

The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows 'php > cours1.php'. The editor window displays the following PHP code:

```
1 <?php
2 // création de tableaux :
3 $tab1 = array ('x', 'y'); $tab2 = array ('key1' => 'val1', 'key2' => 'val2');
4 $tab3 = ['x', 'y']; $tab4 = [3 => 'val1', 7 => 'val2']; // syntaxe depuis PHP 5.4
5
6 // ajout/ modification d'éléments :
7 $tab1[4] = 'z'; $tab3[] = 'z';
8
9 // accès aux éléments
10 echo $tab1[0] . ' ' . $tab4[3] . "\n";
11 print_r ($tab2);
12 ?>
```

Below the editor is a terminal window titled 'Terminal: Local x +'. It shows the command execution and its output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<62> php cours1.php
x val1
Array
(
    [key1] => val1
    [key2] => val2
)
```

The bottom status bar indicates 'All files are up-to-date (4 minutes ago)', '12:3', 'LF', 'UTF-8', '4 spaces', 'Git: master', and an 'Event Log' icon.

Les constantes

The screenshot shows the WebStorm IDE interface. The title bar indicates the file path: `php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm`. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows `php > cours1.php`. The editor displays the following PHP code:

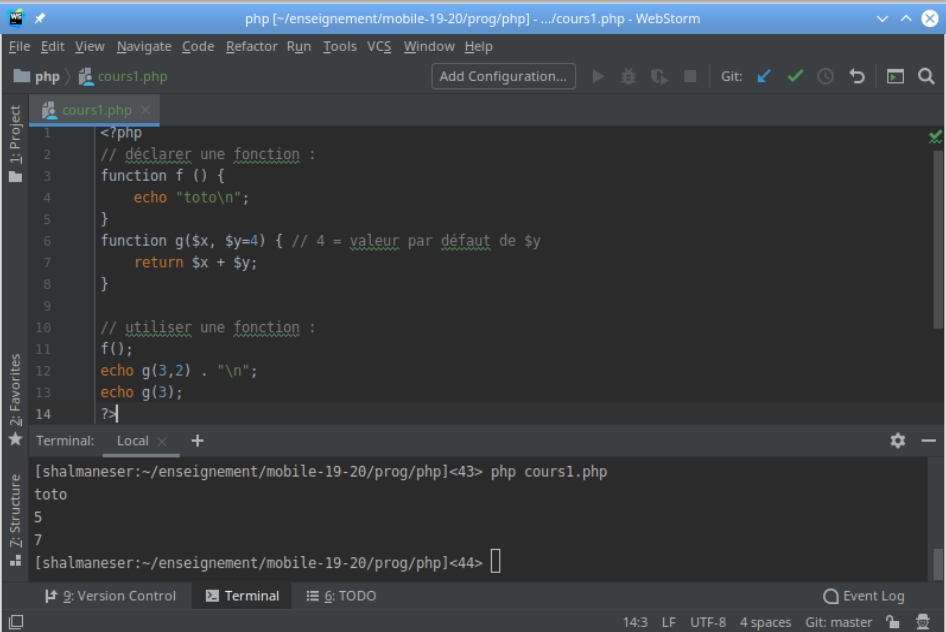
```
1 <?php
2 // déclarer une constante :
3 define ("MYCONST", "toto\n");
4 define ("CST2", 3.4);
5
6 // utiliser une constante :
7 echo MYCONST;
8 echo CST2;
9
10 ?>
11
```

The left sidebar shows the Project view with '1: Project' and '2: Favorites' sections. The bottom status bar shows '9: Version Control', 'Terminal', and 'TODO' tabs. The terminal output is as follows:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<27> php cours1.php
toto
3.4
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<28> 
```

The bottom status bar also displays '10:1 LF UTF-8 4 spaces Git: master' and an 'Event Log' icon.

Les fonctions



The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project pane on the left shows the file structure. The main editor displays the content of 'cours1.php'.

```
<?php
// déclarer une fonction :
function f () {
    echo "toto\n";
}
function g($x, $y=4) { // 4 = valeur par défaut de $y
    return $x + $y;
}

// utiliser une fonction :
f();
echo g(3,2) . "\n";
echo g(3);
?>
```

Below the editor is a terminal window titled 'Terminal: Local'. It shows the execution of the PHP script:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<43> php cours1.php
toto
5
7
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<44> 
```

The bottom status bar indicates the current file is 'cours1.php', the encoding is 'UTF-8', and the git status is 'master'.

La portée des variables (1/4)

The screenshot shows a web development IDE with a blue title bar. The main editor displays a PHP file named `cours1.php` with the following code:

```
1 <?php
2 // scope des variables : global, local, static
3
4 // ici, scope = global :
5 $x = 4;
6
7 function f($y) {
8     echo ($x + $y) . "\n";
9 }
10 f(2);
11 ?>
12
13 |
```

Below the editor is a terminal window titled "Terminal: Local x +". It shows the command `php cours1.php` being executed, resulting in two notices:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<53> php cours1.php
PHP Notice: Undefined variable: x in /home/gonzales/enseignement/mobile-19-20/prog/php/cours1.php on line 8
Notice: Undefined variable: x in /home/gonzales/enseignement/mobile-19-20/prog/php/cours1.php on line 8
```

The IDE interface includes a sidebar on the left with "1: Project", "2: Favorites", and "3: Structure" views. The bottom status bar shows "13:1 LF UTF-8 4 spaces Git: master" and an "Event Log" icon.

La portée des variables (2/4)

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb shows the file path: php > scope3.php. The editor displays the following PHP code:

```
1 <?php
2 // ici, scope = global :
3 $x = 4;
4 $z = 6;
5
6 function f($y) {
7     // pour accéder à x, qui n'est pas défini dans la fonction :
8     // soit utiliser $GLOBALS['x']
9     // soit déclarer global $z;
10    global $z;
11
12    echo ($GLOBALS['x'] + $y) . "\n"; // $GLOBALS : tableau des variables globales
13    echo ($z + $y) . "\n";
14 }
15 f(2);
16 ?>
```

The bottom panel shows the Terminal with the command `php scope3.php` executed. The output shows the values of the variables. The status bar at the bottom indicates the current file is scope3.php, the encoding is UTF-8, and the Git status is master.

La portée des variables (3/4)

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project explorer on the left shows the 'cours1.php' file. The main editor displays the following PHP code:

```
1 <?php
2 // scope des variables : global, local, static
3 $x = 4;
4 function f() {
5     $x = 5; // variable locale à la fonction, différente de la variable globale
6     {
7         $x = 6; // même variable qu'au dessus
8     }
9     echo $x . "\n";
10 }
11 f();
12 echo $x;
13 ?>
14
```

The terminal at the bottom shows the command `php cours1.php` being executed, with the output `6` and `4` on separate lines. The status bar at the bottom indicates the file is encoded in UTF-8 with 4 spaces, and the current branch is 'Git: master'.

La portée des variables (4/4)

The screenshot shows the WebStorm IDE interface. The top bar indicates the file path: `php [~/enseignement/mobile-19-20/prog/php] - .../cours1.php - WebStorm`. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The toolbar shows various icons for file operations and Git integration. The left sidebar displays the Project and Favorites views. The main editor area shows the code for `cours1.php`:

```
1 <?php
2 // scope des variables : global, local, static
3 $x = 4;
4 function f() {
5     static $x = 5; // variable statique
6     $x ++;
7     echo $x . "\n";
8 }
9
10 f();
11 f();
12 ?>
13 |
```

The bottom panel shows the Terminal view with the command `php cours1.php` executed, resulting in the output:

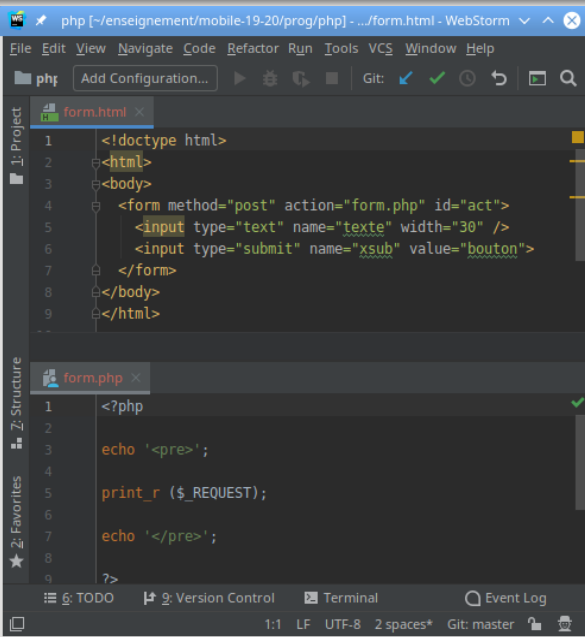
```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<61> php cours1.php
6
7
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<62> |
```

The status bar at the bottom indicates the current file is `9: Version Control`, the editor is `Terminal`, and the file encoding is `UTF-8` with `4 spaces` indentation. The Git status is `Git: master`.

Les variables superglobales

- ▶ **Superglobal** : accessible dans tous les scopes
- ▶ Ce sont des tableaux (`$GLOBALS [' x ']`)
- ▶ Variables contenant les infos transmises par l'utilisateur au serveur web :
 - ▶ `$_POST`
 - ▶ `$_GET`
 - ▶ `$_REQUEST = $_POST + $_GET`
- ▶ Variables contenant les infos sur le serveur web :
 - ▶ `$_SERVER`
- ▶ Variables contenant les infos sur les fichiers uploadés :
 - ▶ `$_FILES`
- ▶ Variables de session et cookies :
 - ▶ `$_SESSION`
 - ▶ `$_COOKIE`

Forms HTML et variables superglobales PHP



sss bouton

```
Array
(
    [texte] => sss
    [xsub] => bouton
)
```


Backend PHP et transfert d'informations en JSON

The image shows a Postman interface for a REST client and a corresponding PHP script. The Postman window displays a POST request to `127.0.0.1/mobile/cours/json.php`. The response body is shown in JSON format:

```
{
  "cle1": "xxx",
  "cle2": "yyy"
}
```

The PHP script, titled `php - json.php`, is shown in a code editor. It sets the content type to `application/json; charset=utf8` and echoes the JSON-encoded data:

```
<?php
header( header: 'Content-type:application/json; charset=utf8' );

echo json_encode([
    'cle1' => 'xxx',
    'cle2' => 'yyy'
]);
```

Les boucles

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project sidebar on the left shows '1: Project' and '2: Favorites'. The main editor displays the file 'cours1.php' with the following PHP code:

```
1 <?php
2 $tab = array ('x', 'y', 'z');
3
4 // boucle pour itérer sur les éléments de $tab :
5 foreach ($tab as $val) { echo "$val "; }
6 echo "\n";
7 foreach ($tab as $key => $val) { echo "$key = $val "; }
8 echo "\n";
9
10 // boucle à la C :
11 for ($i = 0; $i < count($tab); $i++) { echo "${tab[$i]} "; }
12 ?>
13
```

Below the editor is a terminal window titled 'Terminal: Local'. It shows the command 'php cours1.php' being executed, with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<66> php cours1.php
x y z
0 = x 1 = y 2 = z
x y z
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<67> 
```

The bottom status bar indicates 'All files are up-to-date (14 minutes ago)', '13:1', 'LF', 'UTF-8', '4 spaces', 'Git: master', and an 'Event Log' icon.

Les alternatives

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the toolbar, the file explorer shows the project structure with 'cours1.php' selected. The editor displays the following PHP code:

```
1 <?php
2 $x = 3;
3
4 // if then else :
5 if ( $x == 4 )    echo "x vaut 4\n";
6 elseif ($x == 3) echo "x vaut 3\n";
7 else echo "x est différent\n";
8
9 switch ($x) { // switch : similaire au langage C
10     case 3:
11         echo "x = 3"; break;
12     default:
13         echo "x != 3";
14 }
15 ?>
```

The terminal at the bottom shows the command `php cours1.php` being executed, resulting in the output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<69> php cours1.php
x vaut 3
x = 3
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<70> 
```

The status bar at the bottom indicates that all files are up-to-date (19 minutes ago), the current file is 15:3, the encoding is UTF-8, the indentation is 4 spaces, and the VCS status is Git: master.

Les importations

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: php > cours1.php. The editor displays two files: `cours1.php` and `myscript.php`.

`cours1.php` content:

```
1 <?php
2 // importation d'un script php dans un autre :
3 // require_once : importe exactement 1 fois
4 require_once 'myscript.php';
5 echo "fin\n";
6 require_once 'myscript.php';
7 ?>
```

`myscript.php` content:

```
1 <?php
2 echo "je suis dans myscript\n";
3 ?>
```

The terminal at the bottom shows the execution of `php cours1.php`, resulting in the output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/php]<3> php cours1.php
je suis dans myscript
fin
```

Below the terminal output, the command prompt shows `[shalmaneser:~/enseignement/mobile-19-20/prog/php]<4>` with a cursor.

The bottom status bar indicates: WebStorm 2019.3.2 available: // Update... (today 8:49 AM), 9:1 LF UTF-8 4 spaces, Git: master, and an Event Log icon.

Requêtes MySQL \implies utiliser un « *database abstraction layer* »

► ici : utilisation de PDO (PHP Data Objects)

► **Avantages :**

- sécurité (« *prepared statements* »)
- facilité d'utilisation (« *helpers* »)
- réutilisabilité (« *API unifiée de bases de données* »)
- applicabilité (« *compatible avec de nombreuses bases : MySQL, SQLite, Firebird, Oracle, etc.* »)

Petit aperçu de PDO : création d'une instance

```
<?php
// creation de l'instance PDO et connexion à la BD
$dsn = "mysql:host=$mysqlHost;" .
       "dbname=$mysqlDatabase;" .
       "charset=$charset";

// les options
$opt = array (
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false );

$PDO = new PDO($dsn, $mysqlLogin, $mysqlPassword, $opt);
?>
```

Petit aperçu de PDO : utilisation de l'instance

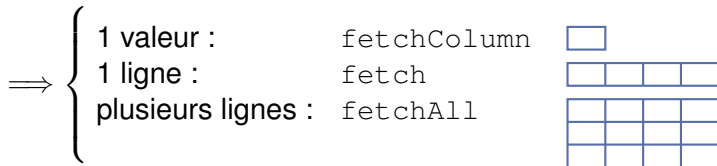
```
<?php
// définition de la requête
$query = "SELECT * FROM $mysqlTable ".
        "WHERE field1=? AND field2 IN (?,?)";
$data = array ( 'f1', 3, 5 );

// envoi et exécution de la requête à la base
$stmt = $pdo->prepare( $query ); // préparation
$exec = $stmt->execute( $data ); // exécution

// récupération du résultat
$resultats = $stmt->fetchAll ( PDO::FETCH_ASSOC );

// affichages
foreach ( $resultats as $un_resultat )
    echo $un_resultat['field1'] ...;
?>
```

► Récupérer des données de la BD



► Options pour fetch et fetchAll :

- PDO::FETCH_NUM : **indices = nombres**
- PDO::FETCH_ASSOC : **indices = noms des champs dans la BD**

PDO::FETCH_NUM :

```
array (  
    0 => 'toto',  
    1 => 'titi'  
);
```

PDO::FETCH_ASSOC :

```
array (  
    'nom' => 'toto',  
    'prenom' => 'titi'  
);
```


► Documentation :

<https://www.php.net/manual/fr/funcref.php>

► Visualisation des erreurs :

En début de programme :

```
ini_set('display_errors', 'On');  
error_reporting(E_ALL);
```

Javascript

Exécution de javascript

- Browsers contiennent un moteur Javascript :



Chakra



Nitro

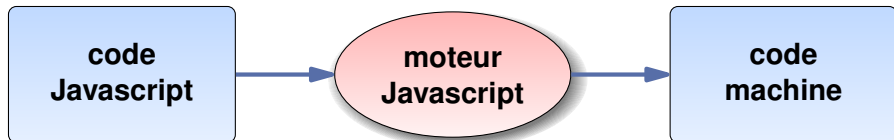


SpiderMonkey



V8

- Vérifient les normes ECMAScript (ES6 = ES2015)
- Principe de fonctionnement :



- Browser exécute ce code machine
- Depuis 2009 : compil/exec hors browser :



HTML et javascript

The screenshot displays the WebStorm IDE interface with a file named `page1.html` open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>intro à javascript</title>
6 </head>
7 <body>
8   <h1>contenu de la page web</h1>
9
10  <!-- insertion d'un code javascript -->
11  <script>
12    console.log('toto');
13  </script>
14 </body>
15 </html>
16
```

To the right, a Chromium browser window titled "intro à javascript" shows the rendered page with the heading "contenu de la page web". The browser's developer tools are open to the Console tab, showing the log output:

```
toto
```

The bottom status bar of WebStorm indicates the file encoding is UTF-8, uses 4 spaces for indentation, and is on the master branch of a Git repository.

Inclusion d'un fichier javascript

The screenshot displays the WebStorm IDE interface. The main editor shows the file `page1.html` with the following HTML structure:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>intro à javascript</title>
6 </head>
7 <body>
8   <h1>contenu de la page web</h1>
9
10  <!-- insertion d'un code javascript -->
11  <script src="page1.js"></script>
12 </body>
13 </html>
14
```

The file explorer on the left shows the project structure with `page1.html` and `page1.js`. The `page1.js` file is open in the editor, showing the following JavaScript code:

```
1 console.log('toto');
2
```

On the right, a Chromium browser window titled "intro à javascript" displays the rendered HTML page, which shows the text "contenu de la page web". The browser's developer tools are open, showing the "Console" tab with the output "toto".

The bottom status bar of the IDE indicates the following details:

- WebStorm 2019.3.2 available: // Update... (1 hour ago)
- 14:1
- LF
- UTF-8
- 4 spaces
- Git: master
- Event Log

Création de variables en javascript : 4 manières



let nom_variable = valeur;

const nom_variable = valeur;

portée de bloc si déclaration dans un bloc

portée de fichier/module sinon

~ global.nom_variable = valeur;

~ window.nom_variable = valeur;

création/modification d'une variable globale

✗ nom_variable = valeur;

modification de la valeur d'une variable si elle existe déjà

création d'une variable globale sinon

✗ **var** nom_variable = valeur;

portée de fonction si déclaration dans une fonction

variable globale si déclaration hors fonction

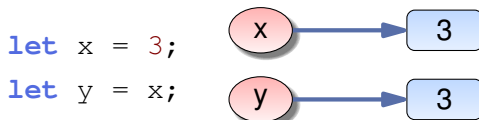


Types des variables (1/2)

Types primitifs :

- ▶ string : `let x = 'toto', y = 'to' + 'to';`
- ▶ number : `let x = 30, y = 4.5;`
- ▶ boolean : `let x = true;`
- ▶ undefined : `let x, y = undefined;`
- ▶ null : `let x = null;`

Types primitifs \Rightarrow copie par valeur :

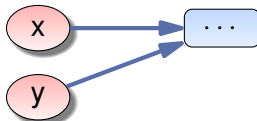


Types référence :

- ▶ Object
- ▶ Array
- ▶ Function

Types référence \Rightarrow copie par référence :

```
let x = ...;  
let y = x;
```



- ▶ **Noms autorisés** : mêmes règles qu'en Java ou C

- ▶ **Convention** : notation Camel

Exemple : `firstName`

- ▶ **Attention** : Javascript sensible à la casse :

Exemple : `firstName` \neq `FirstName`

Les fonctions

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: mobile-19-20 > prog > javascript > function.js. The editor displays the following JavaScript code:

```
1 // les fonctions : function nom_fonc (arguments) { code }, comme en php
2 function f(arg1, arg2) {
3     arg1++;
4     console.log(arg1, arg2);
5 }
6
7 // passage de paramètres :
8 // par référence pour les types référence, copie par valeur pour les types primitifs
9 let x = 3;
10 f(x,x); console.log(x);
11 f(x); console.log(x);
12
```

The bottom panel contains a terminal window with the following output:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<3> node function.js
4 3
3
4 undefined
3
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<4> 
```

The status bar at the bottom shows: External file changes sync may be slow: The current notify(7) watch limit is ... (today 4:06 PM) 12:1 LF UTF-8 4 spaces Git: master.

Les objets (1/3)

The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: `mobile-19-20 > prog > javascript > objects.js`. The editor window displays the following JavaScript code:

```
1 // objets = {}
2 // création d'un objet :
3 let guy = {
4     prenom : 'guy', // couples cle-valeur
5     nom : 'nipigue' // séparation des champs
6                 // par des ','
7 };
8
9 console.log(guy);
10
11 // accès aux champs : 2 possibilités :
12 // notation '.' ou '[]'
13 console.log(guy.nom);
14 console.log(guy['prenom']);
15 let field = 'prenom';
16 console.log(guy[field]);
17
```

The terminal window on the right shows the command `node objects.js` being executed, with the output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<8> node objects.js
{ prenom: 'guy', nom: 'nipigue' }
nipigue
guy
guy
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<9>
```

The bottom status bar indicates the version of WebStorm (2019.3.2) and the current file encoding (UTF-8).

Les objets (2/3)

php [~/enseignement/mobile-19-20/prog/php] - ~/enseignement/mobile-19-20/prog/javascript/cours2.js - WebStorm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

mobile-19-20 > prog > javascript > cours2.js Add Configuration...

1 // creation d'un objet avec une méthode
2 let obj = {
3 champ1 : 3,
4 champ2 : 5,
5 // définition d'une méthode avant ES6
6 display1 : function () {
7 console.log(this.champ1);
8 },
9
10 // définition possible depuis ES6
11 display2 () {
12 console.log(this.champ2);
13 }
14 };
15
16 obj.display1();
17 obj.display2();
18

Terminal: Local x +

[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<56> node cours2.js
3
5
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<57>

1 Event Log

WebStorm 2019.3.2 available: // Update... (today 8:34 AM) 18:1 LF UTF-8 4 spaces Git: master

Les objets (3/3)

php [~/enseignement/mobile-19-20/prog/php] - ~/enseignement/mobile-19-20/prog/javascript/obj_class.js - WebStorm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

home > gonzaes > enseignement > mobile- Add Configuration...

1: Project

obj_class.js x

```
1 // création d'un objet par classe : depuis ES6
2 class MonObjet {
3     // on définit explicitement une méthode constructeur
4     // on ne peut définir qu'un seul constructeur
5     constructor (val1, val2) {
6         // déclaration, initialisation des champs
7         this.champ1 = val1;
8         this.champ2 = val2;
9     }
10
11     // pas besoin de mot clef "function" pour les
12     // méthodes
13     display () {
14         console.log(this.champ1, this.champ2);
15     }
16 }
17
18 let obj = new MonObjet(3,4);
19 obj.display ();
20
```

MonObjet > constructor()

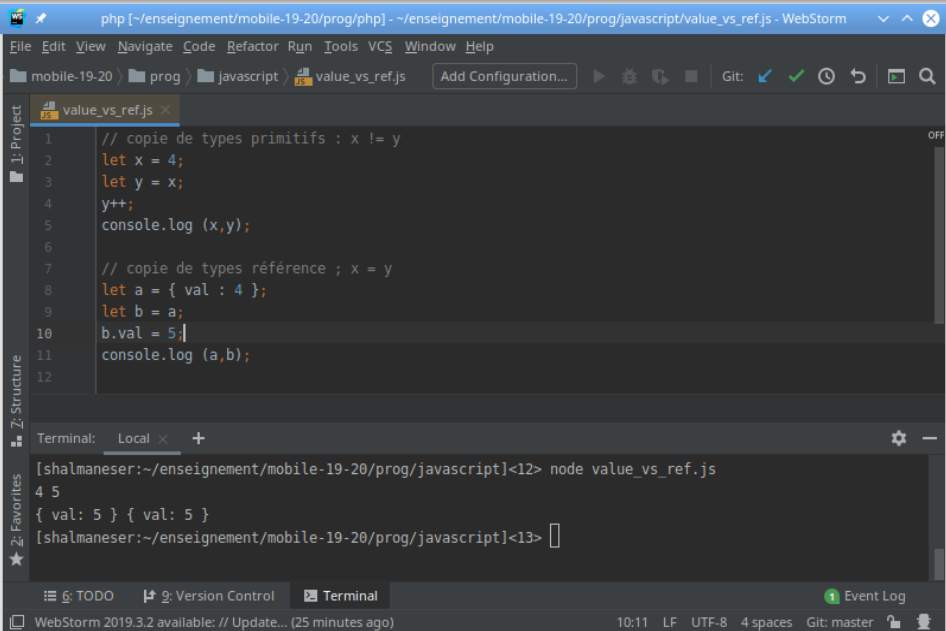
Terminal: Local x +

```
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<1> node obj_class.js
3 4
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<2> 
```

1 Event Log

WebStorm 2019.3.2 available: // Update... (3 minutes ago) 7:28 LF UTF-8 4 spaces Git: master

Comparaison types primitifs / référence



The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: mobile-19-20 > prog > javascript > value_vs_ref.js. The editor displays the following JavaScript code:

```
1 // copie de types primitifs : x != y
2 let x = 4;
3 let y = x;
4 y++;
5 console.log (x,y);
6
7 // copie de types référence ; x = y
8 let a = { val : 4 };
9 let b = a;
10 b.val = 5;
11 console.log (a,b);
12
```

The bottom panel shows the Terminal with the following output:

```
Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<12> node value_vs_ref.js
4 5
{ val: 5 } { val: 5 }
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<13> 
```

The status bar at the bottom indicates: WebStorm 2019.3.2 available: // Update... (25 minutes ago), 10:11, LF, UTF-8, 4 spaces, Git: master.

Les tableaux (1/2)

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: mobile-19-20 > prog > javascript > arrays.js. The editor window displays the following JavaScript code:

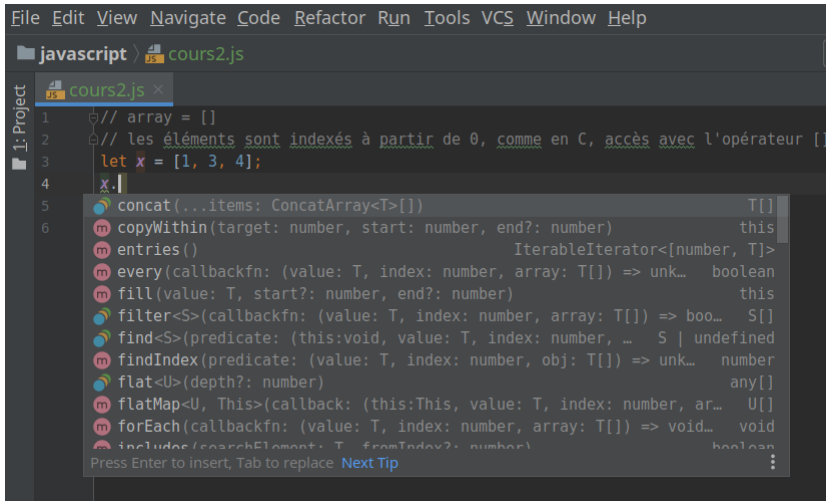
```
1 // array = []
2 // les éléments sont indexés à partir de 0, comme en C, accès avec l'opérateur []
3 let x = [1, 3, 4];
4 console.log(x);
5
6 x[4] = 5;
7 console.log(x);
8
9 // on peut mettre de éléments de types différents dans un même array :
10 x[5] = 'toto';
11 console.log(x);
12
```

Below the editor is a terminal window titled "Terminal: Local". It shows the command `node arrays.js` being executed, with the following output:

```
[ 1, 3, 4 ]
[ 1, 3, 4, <1 empty item>, 5 ]
[ 1, 3, 4, <1 empty item>, 5, 'toto' ]
```

The bottom status bar indicates the current version of WebStorm (2019.3.2), the file encoding (UTF-8), the number of spaces (4), and the current Git branch (master).

Les tableaux sont des objets :



The screenshot shows an IDE with a menu bar (File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help) and a project explorer on the left showing '1: Project' and 'cours2.js'. The main editor displays the following code:

```
1 // array = []
2 // les éléments sont indexés à partir de 0, comme en C, accès avec l'opérateur []
3 let x = [1, 3, 4];
4 x.
5
6
```

A dropdown menu is open below the cursor, listing various array methods with their signatures and return types:

- concat(...items: ConcatArray<T>[]) T[]
- copyWithin(target: number, start: number, end?: number) this
- entries() IterableIterator<[number, T]>
- every(callbackfn: (value: T, index: number, array: T[]) => unknown) boolean
- fill(value: T, start?: number, end?: number) this
- filter<S>(callbackfn: (value: T, index: number, array: T[]) => boolean) S[]
- find<S>(predicate: (this: void, value: T, index: number, ... S | undefined) => boolean) S | undefined
- findIndex(predicate: (value: T, index: number, obj: T[]) => boolean) number
- flat<U>(depth?: number) any[]
- flatMap<U, This>(callback: (this: This, value: T, index: number, array: T[]) => U[]) U[]
- forEach(callbackfn: (value: T, index: number, array: T[]) => void) void
- includes(searchElement: T, fromIndex?: number) boolean

At the bottom of the menu, it says 'Press Enter to insert, Tab to replace' and 'Next Tip'.

⇒ contiennent des méthodes (length, filter, forEach, etc.)

Typage dynamique

The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: `mobile-19-20 > prog > javascript > typage.js`. The editor window displays the following JavaScript code:

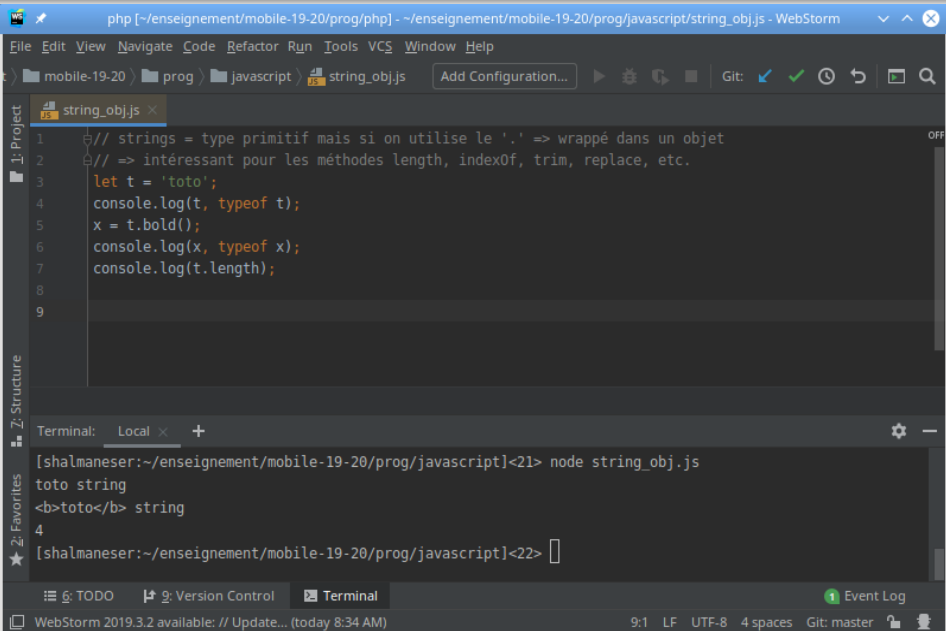
```
1 // chaque variable a un type, qui peut changer dynamiquement
2 let x = 4;
3 console.log(typeof x);
4
5 x = 'toto';
6 console.log(typeof x);
7
8 let y;
9 console.log(y, typeof y); // undefined = type et valeur
10 y = null;
11 console.log(y, typeof y);
12
```

Below the editor is a terminal window with the title "Terminal: Local x +". It shows the command `node typage.js` being executed, with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<16> node typage.js
number
string
undefined 'undefined'
null 'object'
```

The bottom status bar indicates the current state: "WebStorm 2019.3.2 available: // Update... (today 8:34 AM)", "12:1", "LF", "UTF-8", "4 spaces", "Git: master", and "1 Event Log".

Retour sur les string et les objets



The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: mobile-19-20 > prog > javascript > string_obj.js. The editor window displays the following JavaScript code:

```
1 // strings = type primitif mais si on utilise le '.' => wrappé dans un objet
2 // => intéressant pour les méthodes length, indexOf, trim, replace, etc.
3 let t = 'toto';
4 console.log(t, typeof t);
5 x = t.bold();
6 console.log(x, typeof x);
7 console.log(t.length);
8
9
```

The left sidebar shows the Project and Structure views. The bottom panel contains a Terminal window with the following output:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<21> node string_obj.js
toto string
<b>toto</b> string
4
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<22> 
```

The status bar at the bottom indicates: WebStorm 2019.3.2 available: // Update... (today 8:34 AM), 9:1, LF, UTF-8, 4 spaces, Git: master.

Template literals

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: javascript > template_literal.js. The main editor displays the following JavaScript code:

```
1 // concaténation de strings : +
2 let x = 4;
3 let y = 'x vaut : ' + x + '\n' + '3 fois' moins que ' + (3 * x);
4 console.log(y);
5
6 // template literals (backquotes : `` ): depuis ES6s
7 let z = `x vaut : ${x}
8 '3 fois' moins que ${3 * x}`;
9 console.log(z);
10
```

Below the editor is the Terminal panel, titled 'Terminal: Local'. It shows the command `node template_literal.js` being executed, with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<1> node template_literal.js
x vaut : 4
'3 fois' moins que 12
x vaut : 4
'3 fois' moins que 12
```

The bottom status bar indicates the current file encoding as UTF-8, 4 spaces indentation, and the Git status as master.

Les constantes

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb path is: t > mobile-19-20 > prog > javascript > constante.js. The editor displays the file 'constante.js' with the following code:

```
1 // constante = on ne peut plus réassigner une valeur à la variable
2 // mais si c'est un objet ou un tableau, on peut modifier son contenu
3 const x = 4;
4 // x = 3; => erreur
5
6 const y = { val : 3 };
7 // y = { val : 4 }; => erreur
8 y.val = 5; // valide, ce n'est pas une réaffectation de y
9 console.log(y);
10
11
12
```

The terminal at the bottom shows the execution of the script:

```
Terminal: Local x +
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<19> node constante.js
{ val: 5 }
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<20> 
```

The status bar at the bottom indicates: WebStorm 2019.3.2 available: // Update... (today 8:34 AM), 11:11, LF, UTF-8, 4 spaces, Git: master, and 1 Event Log.

Itérer les champs d'un objet (1/2)

The screenshot shows the WebStorm IDE interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: `mobile-19-20 > prog > javascript > iter.js`. The editor window displays the following JavaScript code:

```
1 class MonObjet {
2   constructor (val1, val2) {
3     this.champ1 = val1;
4     this.champ2 = val2;
5   }
6
7   // affichage des champs de l'objet
8   display () {
9     // for .. in : parcourt des clefs de
10    // l'objet ou des index pour un tableau
11    for (let key in this)
12      console.log(key, this[key]);
13  }
14 }
15
16 let obj = new MonObjet(3,4);
17 obj.display ();
18
```

The right sidebar shows the 'Terminal' tab with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<> node iter.js
champ1 3
champ2 4
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<>
```

The bottom status bar indicates 'WebStorm 2019.3.2 available: // Update... (5 minutes ago)', '15:1', 'LF', 'UTF-8', '4 spaces', 'Git: master', and '1 Event Log'.

Itérer les champs d'un objet (2/2)

The screenshot shows the WebStorm IDE interface. The main editor displays the file `iterOf.js` with the following JavaScript code:

```
1 class MonObjet {  
2   constructor (val1, val2) {  
3     this.champ1 = val1;  
4     this.champ2 = ['toto', 'titi', 'tutu'];  
5   }  
6  
7   // affichage des champs de l'objet  
8   display () {  
9     // for .. of : parcourt des valeurs d'un objet  
10    // itérable (par exemple un tableau)  
11    for (let key of this.champ2)  
12      console.log(key);  
13  }  
14 }  
15  
16 let obj = new MonObjet( val1: 3, val2: 4);  
17 obj.display ();  
18 |
```

The terminal on the right shows the execution of the code:

```
Terminal: Local x +  
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<5> node iterOf.js  
toto  
titi  
tutu  
[shalmaneser:~/enseignement/mobile-19-20/prog/javascript]<6> |
```

The bottom status bar indicates the WebStorm version (2019.3.2) and the current file encoding (UTF-8).

Copier (cloner) un objet

The screenshot shows the WebStorm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb path is: mobile-19-20 > prog > javascript > clone.js. The editor displays the following JavaScript code:

```
1 let obj = {
2   champ1 : 3,
3   champ2 : [1,2,3],
4   display () { console.log(':::', this.champ1);}
5 };
6
7 // création d'un objet vide puis recopie de tous
8 // les champs de obj dans obj2 => copie obj
9 const obj2 = {};
10 for (let key in obj) {
11   console.log(key);
12   obj2[key] = obj[key];
13 }
14 obj2.display();
15
16 // ... = spread operator
17 // expanse l'ensemble des champs de obj2
18 const obj3 = { ...obj2 };
19 obj3.display ();
20
```

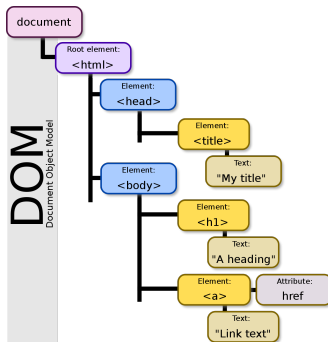
The right sidebar shows the 'Terminal' tab with the following output:

```
[shalmaneser:~/enseignement/mobile-19-20/
prog/javascript]<7> node clone.js
champ1
champ2
display
::: 3
::: 3
[shalmaneser:~/enseignement/mobile-19-20/
prog/javascript]<8>
```

The bottom status bar indicates: WebStorm 2019.3.2 available: // Update... (8 minutes ago), 20:1, LF, UTF-8, 4 spaces, Git: master.

Javascript et le DOM

DOM : Document Object Model



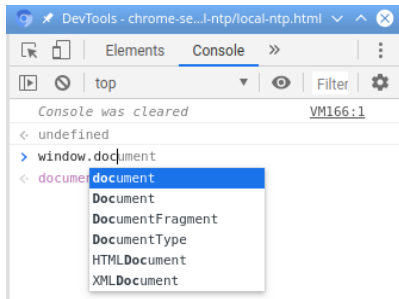
[image wikipedia]

- ▶ Représente la page web affichée
- ▶ Structure d'arbre — racine : document

Javascript : window vs document

Pour Javascript :

- ▶ Dans un browser, `window` = objet global
- ▶ `document` = `window.document`
⇒ `document` est une propriété de `window`
- ▶ `document` = l'objet « visible » du browser



Manipuler le DOM

php [~/enseignement/mobile-19-20/prog/php] - ~/enseignement/mobile-19-20/prog/javascript/DOM2.html - WebStorm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

mobile-19-20 > prog > javascript > DOM2.html Add Configuration...

DOM2.html x

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Titre</title>
6 </head>
7 <body>
8   <h1 id="titre">Titre de la page</h1>
9
10  <!-- on crée un bouton et on associe l'exécution de
11  la fonction javascript maFonction quand on clique -->
12   <button onclick="maFonction ();">mon bouton</button>
13
14  <script>
15    function maFonction () {
16      document.getElementById('titre').innerHTML = 'mon nouveau titre';
17    }</script>
18 </body>
19 </html>
20 |
```

1: Project

2: Structure

3: Favorites

mon bouton

Titre de la page

mon bouton

mon nouveau titre

Event Log

WebStorm 2019.3.2 available: // Update... (10 minutes ago)

20:1 LF UTF-8 4 spaces Git: master

Rajouter des éléments au DOM

The screenshot shows the WebStorm IDE with a project named 'mobile-19-20'. The file 'DOM3.html' is open, displaying the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head><meta charset="UTF-8"></head>
4   <body>
5     <button onclick="maFonction ();">mon bouton</button>
6     <ul id="liste"></ul>
7
8   <script>
9     let nb = 1; // le numéro de l'item (variable globale)
10    function maFonction () { // à chaque appel : rajoute un nouvel item
11      let node = document.createElement("li"); // crée un nouveau noeud <li></li>
12      let textnode = document.createTextNode('item ${nb++}'); // crée un noeud contenant du texte
13      node.appendChild(textnode); // place le texte à l'intérieur du <li></li>
14
15      document.getElementById("liste").appendChild(node); // rajoute le nouvel item au <ul></ul>
16    }
17  </script>
18 </body>
19 </html>
```

On the right side, a preview of the rendered HTML is shown, featuring a button labeled 'mon bouton' and a list containing two items:

- item 1
- item 2

The bottom status bar indicates 'WebStorm 2019.3.2 available: // Update... (32 minutes ago)', '20:1', 'LF', 'UTF-8', '4 spaces', 'Git: master', and '1 Event Log'.