



Chaînes
Chaîne de caractères ?
PHP
2

- Série de 0, 1 ou plusieurs caractères.
- 1 caractère = 1 octet → 256 valeurs différentes.
→ Pas de support d'UNICODE.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETH	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8	€	□		f	"	...	†	‡	^	%	§	<	CE	□	Ž	□
9	□	'	"	"	"	•	—	—	™	§	>	œ	□	ž	Ÿ	
A	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

codage "iso-8859-1"

- Il est possible d'avoir des retours chariots, tabulations, etc dans une chaîne.

ISO 8859-1 (latin-1 ou européen occidental)

ISO 8859-2 (latin-2 ou européen central)

ISO 8859-3 (latin-3 ou européen du Sud)

ISO 8859-4 (latin-4 ou européen du Nord)

ISO 8859-5 (cyrillique)

ISO 8859-6 (arabe)

ISO 8859-7 (grec)

ISO 8859-8 (hébreu)

ISO 8859-9 (latin-5 ou turc)

ISO 8859-10 (latin-6 ou nordique)

ISO 8859-11 (thaï)

ISO 8859-12 abandonné

ISO 8859-13 (latin-7 ou balte)

ISO 8859-14 (latin-8 ou celtique)

ISO 8859-15 (latin-9)

ISO 8859-16 (latin-10 ou européen du Sud-Est)

francois.piat@univ-fcomte.fr

Chaînes

Guillemets

PHP 3

- **Guillemets simples**

```
$prenom = 'François';
```
- **Guillemets doubles**

```
$nom = "Piat";  
$c = "Je m'appelle $prenom $nom";
```

→ Je m'appelle François Piat
- **Saut de lignes, tabulations, etc.**

```
$c = "Je  
m'appelle  
$prenom $nom";
```
- **Protection de caractères**

```
'Je m\'appelle François'  
"Je lui ai dit \"Bonjour\""
```

francois.piat@univ-fcomte.fr

Chaînes

heredoc

PHP 4

Syntaxe heredoc

```
<<<IDENTIFIANT  
texte  
IDENTIFIANT;
```

```
$txt = <<<TEXTE  
Bonjour,  
    Je m'appelle $prenom, et je suis enseignant  
    à l'Université de Franche-Comté.  
TEXTE;
```

francois.piat@univ-fcomte.fr

Chânes

Opération sur les chaînes : concaténation

PHP 5

- Opérateur `.` (point)

```
$a = 'François';  
$b = 'Piat';  
$c = $a . ' ' . $b;
```

➔ François Piat
- Concaténation et affectation avec l'opérateur `.=`

```
$c .= ' enseigne PHP';
```

➔ François Piat enseigne PHP
- Eviter les concaténations autant que possible
 - Utiliser la substitution de variables avec les guillemets doubles
 - Utiliser l'instruction `echo` avec des paramètres séparés par des virgules

francois.piat@univ-fcomte.fr

Chânes

Afficher une chaîne

PHP 6

- Afficher une chaîne = envoyer la chaîne au navigateur
- Instruction `echo`

```
echo '<h1>', $titre, '</h1><p>', $txt, '</p>';
```
- Instruction `print`

```
print '<h1>'.$titre.'</h1><p>'.$txt.'</p>';
```
- Préférer d'entourer les chaînes avec des guillemets simples et d'utiliser l'instruction `echo`
- Affichage avec formatage : `printf()` et `sprintf()`;

francois.piat@univ-fcomte.fr

Chaînes
Bibliothèque
PHP 7

- Bibliothèque très complète pour gérer les chaînes de caractères (98 fonctions) **mais**
- Pas de normalisation dans les noms de fonction

`substr() strpos() str_replace() trim()`
`htmlentities() html_entity_decode()`
- Pas de règles dans l'ordre de passage des paramètres
- Utilisation de l'aide de PHP obligatoire
<http://www.php.net/manual/fr/book.strings.php>
- Si texte en UTF8 → utiliser la bibliothèque **mb_string**
<http://www.php.net/manual/fr/ref.mbstring.php>

francois.piat@univ-fcomte.fr

Chaînes
Fonctions incontournables
PHP 8

`$a = 'A l p h a b e t';`

↑	↑	↑	↑	↑	↑	↑	↑
0	1	2	3	4	5	6	7

<ul style="list-style-type: none"> • <code>strlen()</code> • <code>substr()</code> • <code>strpos()</code> • <code>trim()</code> • <code>explode()</code> 	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><code>strlen(\$a)</code></td> <td style="text-align: right;">→ 8</td> </tr> <tr> <td><code>substr(\$a, 1, 3)</code></td> <td style="text-align: right;">→ lph</td> </tr> <tr> <td><code>substr(\$a, 2)</code></td> <td style="text-align: right;">→ phabet</td> </tr> <tr> <td><code>substr(\$a, -2)</code></td> <td style="text-align: right;">→ et</td> </tr> <tr> <td><code>strpos(\$a, 'h')</code></td> <td style="text-align: right;">→ 3</td> </tr> <tr> <td><code>strpos(\$a, 'z')</code></td> <td style="text-align: right;">→ FALSE</td> </tr> <tr> <td><code>strpos(\$a, 'A')</code></td> <td style="text-align: right;">→ 0</td> </tr> </table>	<code>strlen(\$a)</code>	→ 8	<code>substr(\$a, 1, 3)</code>	→ lph	<code>substr(\$a, 2)</code>	→ phabet	<code>substr(\$a, -2)</code>	→ et	<code>strpos(\$a, 'h')</code>	→ 3	<code>strpos(\$a, 'z')</code>	→ FALSE	<code>strpos(\$a, 'A')</code>	→ 0
<code>strlen(\$a)</code>	→ 8														
<code>substr(\$a, 1, 3)</code>	→ lph														
<code>substr(\$a, 2)</code>	→ phabet														
<code>substr(\$a, -2)</code>	→ et														
<code>strpos(\$a, 'h')</code>	→ 3														
<code>strpos(\$a, 'z')</code>	→ FALSE														
<code>strpos(\$a, 'A')</code>	→ 0														

francois.piat@univ-fcomte.fr

Chaînes Fonctions incontournables PHP 9

`$a = 'A l p h a b e t';`

A	l	p	h	a	b	e	t
↑	↑	↑	↑	↑	↑	↑	↑
0	1	2	3	4	5	6	7

- `strpos()`

```

$a = 'Alphabet';
$pos = strpos($a, 'A');
if ($pos === FALSE) {
    echo 'A pas trouvé dans Alphabet';
} else {
    echo 'A est la lettre ', $pos, ' dans Alphabet';
}
  
```

francois.piat@univ-fcomte.fr

Chaînes Codage serveur → client PHP 10

Dans le sens PHP → HTML

- Protéger les caractères de la syntaxe HTML
- Remplacer les caractères accentués

`htmlentities(chaine, gestion_guillemets, encodage)`

<	→	<
>	→	>
é	→	é
è	→	è
à	→	à
î	→	î
...		

```

$x = 'a < b => b > a';
htmlentities($x, ENT_QUOTES, 'ISO-8859-1');
  
```

→ a > b => b < a

```

$x = 'François';
htmlentities($x, ENT_QUOTES, 'UTF8');
  
```

→ François

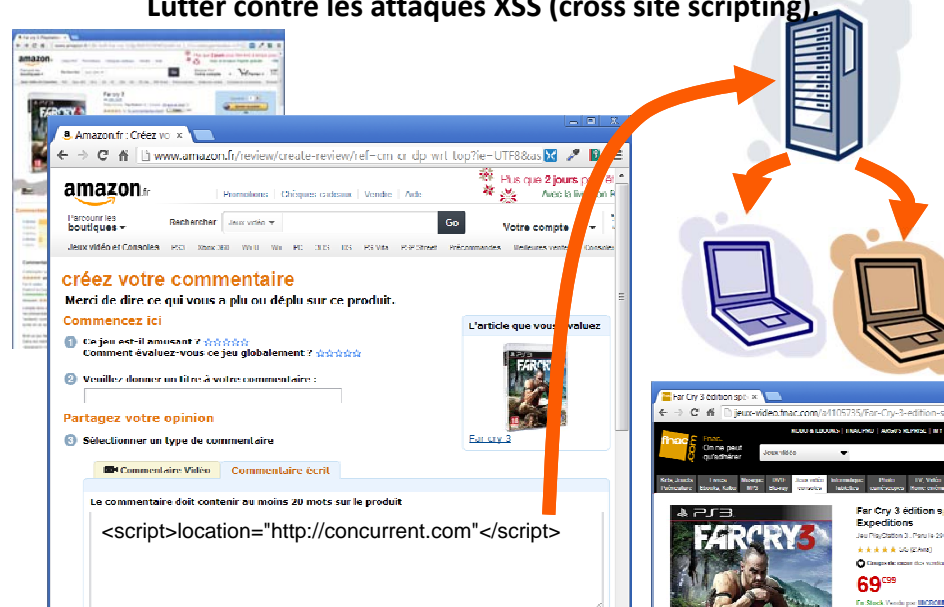
francois.piat@univ-fcomte.fr

Chaînes

Codage serveur → client

PHP 11

Lutter contre les attaques XSS (cross site scripting).



The diagram illustrates a cross-site scripting (XSS) attack. On the left, a browser window shows the Amazon.fr 'create review' page. The user input field contains the malicious payload: `<script>location="http://concurrent.com"</script>`. An orange arrow points from this input field to a server icon in the center. Another orange arrow points from the server to two laptop icons on the right, representing the redirection of the user to the attacker's site.

amazon.fr

Créez votre commentaire

Merci de dire ce qui vous a plu ou déplu sur ce produit.

Commencez ici

Comment évaluez-vous ce jeu globalement ?

Partagez votre opinion

Le commentaire doit contenir au moins 20 mots sur le produit

`<script>location="http://concurrent.com"</script>`

francois.piat@univ-fcomte.fr

Chaînes

Codage serveur → client

PHP 12

Lutter contre les attaques XSS (cross site scripting).

```
htmlspecialchars('<script>location="http://concurrent.com"</script>',
ENT_QUOTES, 'ISO-8859-1');
```

↓

```
&lt;script&gt;location=&quot;http://concurrent.com&quot;&lt;/script&gt;
```

Supprimer les tags HTML dans une chaîne : `strip_tags()`

```
strip_tags('<script>location="http://concurrent.com"</script>');
```

↓

```
location="http://concurrent.com"
```

francois.piat@univ-fcomte.fr

Chaînes

Codage client → BD

PHP
13

- Dans le sens saisie utilisateur → serveur base de données
- Protéger les caractères avec signification particulière

```
mysqli_real_escape_string()  
  
mysqli_escape_string()  
  
addslashes()
```

francois.piat@univ-fcomte.fr



Tableaux

Tableaux ?

PHP 15

- <http://php.net/manual/fr/language.types.array.php> :

*“Ce type est optimisé pour différentes utilisations ; il peut être considéré comme un **tableau**, une **liste**, une **table de hachage**, un **dictionnaire**, une **collection**, une **pile**, une **file d'attente** et probablement **plus**.”*

	1 entier	Tableau de 100 000 entiers
C	8 octets	0,76 Mo
PHP	48 octets	4,7 Mo + 9,3 Mo → 14 Mo

francois.piat@univ-fcomte.fr

Tableaux

2 sortes de tableaux

PHP 16

- Tableaux avec un indice numérique :
l'indice est un entier positif ou égal à 0

\$t[0] \$t[1] \$t[100]

- Tableaux associatifs (table de hachage, dictionary, map) :
l'indice est une chaîne de caractères

\$t['nom'] \$t['prenom'] \$t['tél']

- Nombre d'éléments dans un tableau : `count()`

\$nb = count(\$t);

francois.piat@univ-fcomte.fr

Tableaux

Contenu mixte

PHP 17

- Les éléments d'un tableau peuvent être de n'importe quel type.

```
$t[0] = 'Pierre';  
$t[1] = 'Paul';  
$t[2] = 'Jacques';
```

```
$t[0] = 15;  
$t[1] = 10;  
$t[2] = 18;
```

- Des éléments de types différents peuvent se trouver dans le même tableau.

```
$t[0] = 'Pierre';  
$t[1] = 15;  
$t[2] = true;  
$t[3] = array('abc', 100, false);
```

francois.piat@univ-fcomte.fr

Tableaux

Création indice numérique

PHP 18

- `$var[] = valeur;`

```
$t[] = 10;  
$t[] = 8;  
$t[] = 15;
```

- `array()`

```
$t = array(10, 8, 15);
```

- `range()`

```
$t = range(100, 150);  
$t = range('a', 'z');
```

- `explode()`

```
$t = explode('/', '25/08/2010');
```

- `array_fill()`

```
$t = array_fill(0, 10, true);
```

- `array_merge()` `array_diff()` `array_filter()`

francois.piat@univ-fcomte.fr

Tableaux

Indice numérique

PHP 19

```
$ta[] = 'abc';
```

```
$tb[] = 'abc';  
$tb[] = 10;  
$tb[6] = 1234;  
$tb[8] = -1;
```

**\$tb a des trous dans ses indices :
2, 3, 4, 5 et 7
→ Pas de boucle avec **for****

```
$tc[5] = 'abc';  
$tc[] = 10;  
$tc[] = 1234;
```

**L'index de \$tc ne commence pas à 0
→ Boucle avec **for** hasardeuse**

francois.piat@univ-fcomte.fr

Tableaux

Création tableaux associatifs

PHP 20

- **tableau associatif = table de hashage = dictionnaire**
→ associer une valeur à nom
- `$var['nom_index'] = valeur;`

```
$t['nom'] = 'Pierre';  
$t['prénom'] = 'Paul';  
$t['redoublant'] = false;  
$t['nbAbsence'] = 0;  
$t['moyenne'] = 15.75;
```

francois.piat@univ-fcomte.fr

Tableaux	Fonction array()	PHP 21
<ul style="list-style-type: none">• Sans argument pour créer un tableau vide <div><code>\$t = array();</code></div> <div><code>\$ta = \$tb = \$tc = array();</code></div> <ul style="list-style-type: none">• Avec argument pour créer un tableau non vide <div><code>\$t_num = array('Pierre', false, 10, 14, 'abs');</code></div> <div><code>\$t_asso = array('nom' => 'Pierre', 'absent' => false, 'note1' => 10, 'note2' => 14, 'note3' => 'abs');</code></div> <div><code>\$t_mieux = array('nom' => 'Pierre', 'absent' => false, 'notes' => array(15, 16, 'abs'));</code></div>		

francois.piat@univ-fcomte.fr

Tableaux	Lecture de tableaux	PHP 22
<ul style="list-style-type: none">• Directement quand on connaît l'indice de façon certaine <div><code>\$a = \$t[5]; \$b = \$t['nom'];</code></div> <ul style="list-style-type: none">• En vérifiant si l'indice est présent <div><code>if (isset(\$t[6])) { \$c = \$t[6]; } else { \$c = null; }</code></div> <div><code>\$c = (isset(\$t[6])) ? \$t[6] : null;</code></div> <ul style="list-style-type: none">• Par itérations : for et foreach ou fonctions spécifiques		

francois.piat@univ-fcomte.fr

Tableaux
Boucle for (1)
PHP 23

- **for** est à utiliser
 - avec un tableau à indice numérique
 - si il n'y a pas de trous dans les indice.

```
$t[] = 'abc';
$t[] = 10;
$t[6] = 1234;
$t[8] = -1;
```

0	abc
1	10
6	1234
8	-1

```
$nb = count($t); 4
for ($i = 0; $i < $nb; $i++) {
    echo $t[$i];
}
```

\$i	\$t[\$i]
0	abc
1	10
2	???
3	???

francois.piat@univ-fcomte.fr

Tableaux
Boucle for (2)
PHP 24

Ne jamais appeler une fonction dans le test d'itération.

```
for ($i = 0; $i < count($t); $i++) {
    echo $t[$i];
}
```

❌

```
$nb = count($t);
for ($i = 0; $i < $nb; $i++) {
    echo $t[$i];
}
```

✅

```
for ($i = 0, $nb = count($t); $i < $nb; $i++) {
    echo $t[$i];
}
```

✅✅

francois.piat@univ-fcomte.fr

Tableaux	Boucle for (3)	PHP 25
<p>A utiliser quand l'ordre de traitement n'a pas d'importance :</p>		
<pre>for (\$i = count(\$t) - 1; \$i >= 0; \$i--) { ... }</pre>		
<pre>for (\$i = count(\$t); \$i--;) { ... }</pre>		
francois.piat@univ-fcomte.fr		

Tableaux	Boucle foreach (1)	PHP 26
<ul style="list-style-type: none">• foreach est à utiliser<ul style="list-style-type: none">• avec un tableau associatif• avec un tableau à indice numérique, dont les indices ne sont pas consécutifs• foreach travaille sur une copie du tableau si on modifie le tableau au cours de la boucle. Attention à l'occupation mémoire avec des tableaux importants		
francois.piat@univ-fcomte.fr		

Tableaux

Boucle foreach (2)

PHP 27

```
foreach ($Tableau as $Valeur) {
    ...
}
```

```
$t['nom'] = 'Pierre';
$t['présent'] = true;
$t['moyenne'] = 15.36;
```

```
foreach ($t as $val) {
    echo $val;
}
```

\$val
Pierre
true
15.36

```
$t[0] = 125;
$t[5] = 523;
$t[10] = 4587;
```

```
foreach ($t as $val) {
    echo $val;
}
```

\$val
125
523
4587

francois.piat@univ-fcomte.fr

Tableaux

Boucle foreach (3)

PHP 28

```
foreach ($Tableau as $Clé => $Valeur) {
    ...
}
```

```
$t['nom'] = 'Pierre';
$t['présent'] = true;
$t['moyenne'] = 15.36;
```

```
foreach ($t as $cle => $val) {
    echo $cle, $val;
}
```

\$cle	\$val
nom	Pierre
présent	true
moyenne	15.36

francois.piat@univ-fcomte.fr

Tableaux	Autres fonctions (1)		PHP 29
<ul style="list-style-type: none">• Une bibliothèque de fonctions très fournie (77 fonctions)<ul style="list-style-type: none">• Vérifier dans l'aide de PHP les fonctions• Vérification de clés et/ou de valeurs <code>array_keys()</code> <code>array_values()</code> <code>array_key_exists()</code> <code>in_array()</code> <code>array_search()</code>• Tris<ul style="list-style-type: none">• Par valeur avec changement d'indice : <code>sort()</code> <code>rsort()</code> <code>usort()</code>• Par valeur : <code>asort()</code> <code>arsort()</code> <code>uasort()</code>• Par clés : <code>ksort()</code> <code>krsort()</code> <code>uksort()</code>• Tous les tris sont faits sur le tableau original			
francois.piat@univ-fcomte.fr			

Tableaux	Autres fonctions (2)		PHP 30
<ul style="list-style-type: none">• Opérations ensemblistes sur des tableaux <code>array_diff()</code> <code>array_intersect()</code> <code>array_merge()</code> <code>array_unique()</code>• Supprimer, insérer, remplacer des séries d'éléments <code>array_splice()</code> <code>array_slice()</code>• Piles <code>array_push()</code> <code>array_pop()</code> <code>array_shift()</code> <code>array_unshift()</code>			
francois.piat@univ-fcomte.fr			

Tableaux

Autres fonctions (3)

PHP
31

Transformer les éléments d'un tableau en variables

list()

```
$a = '31/12/1999';  
$t = explode('/', $a);  
list($J, $M, $A) = $t;
```

```
$t[0] = 31  
$t[1] = 12  
$t[2] = 1999
```

```
$J = 31  
$M = 12  
$A = 1999
```

extract()

```
$tab['Pierre'] = 18;  
$tab['Paul'] = 10;  
$tab['Jacques'] = 15;  
extract($tab);
```

```
$Pierre = 18  
$Paul = 10  
$Jacques = 15
```

francois.piat@univ-fcomte.fr