

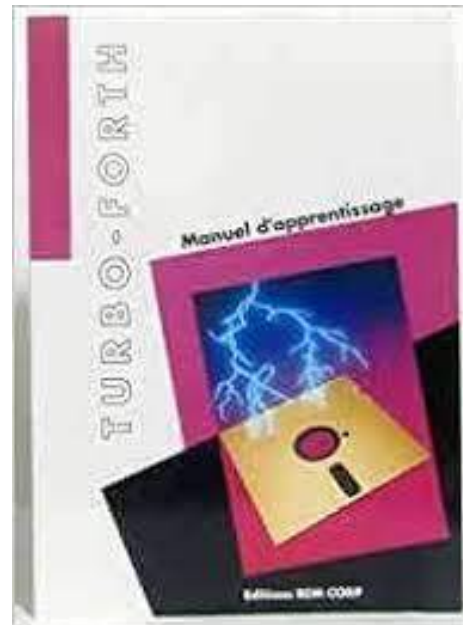
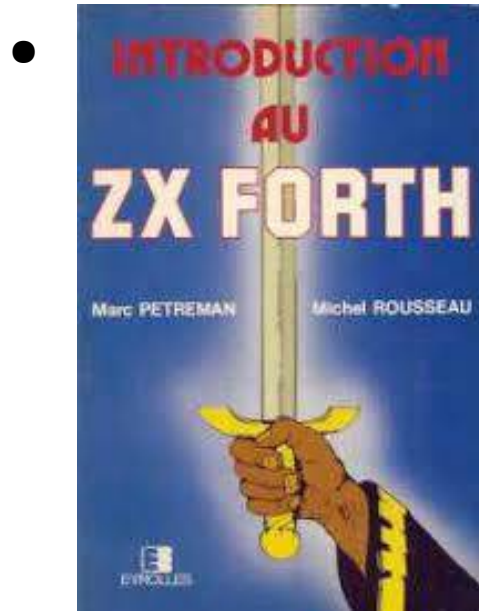
## FORTH & PHP

- FORTH is an interesting language.
- But learning it, for a beginner, requires a lot of research.
- The aim of what will be explained here is to provide assistance to apprentice FORTH programmers...

- INTRODUCTION
- I have been programming in FORTH since 1982
- I started FORTH on ZX81, then ORIC, AMSTRAD, THOMSON, IBM PC....
- I have written numerous articles on FORTH in the French press...

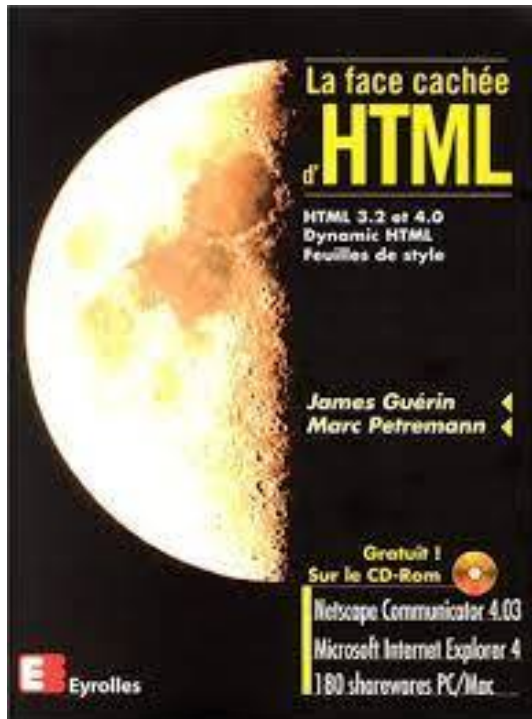
# FORTH & PHP

- I also participated in the writing of several books on FORTH



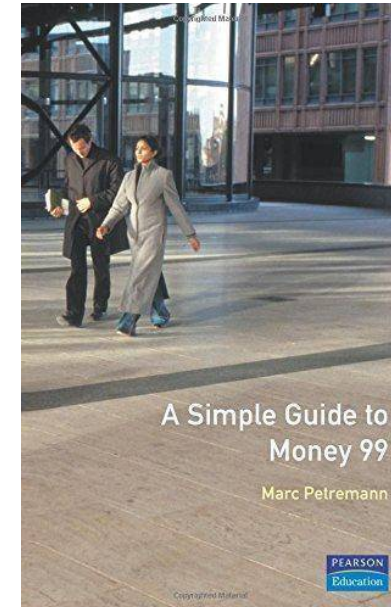
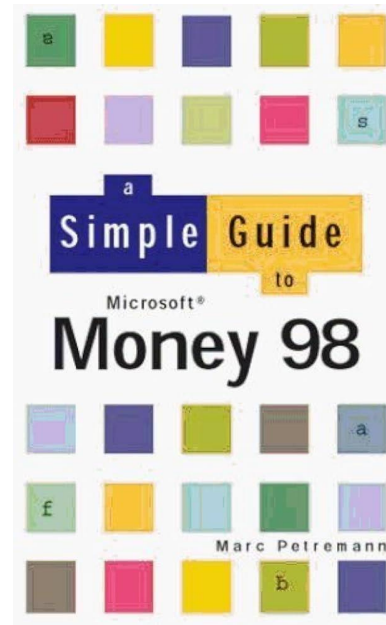
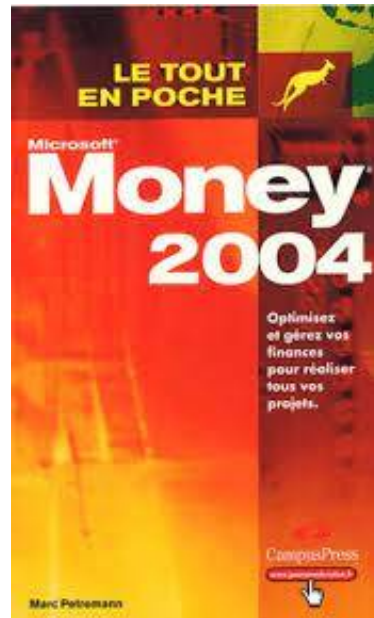
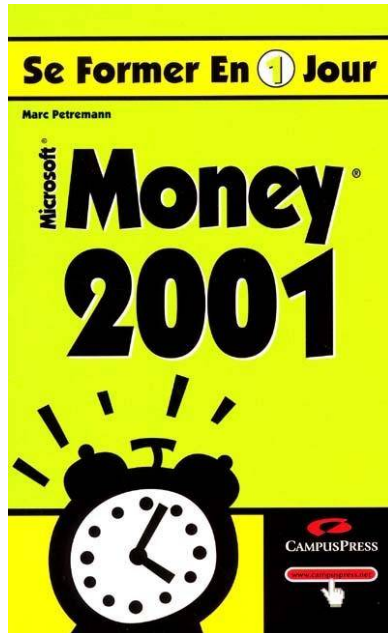
# FORTH & PHP

- One book about HTML



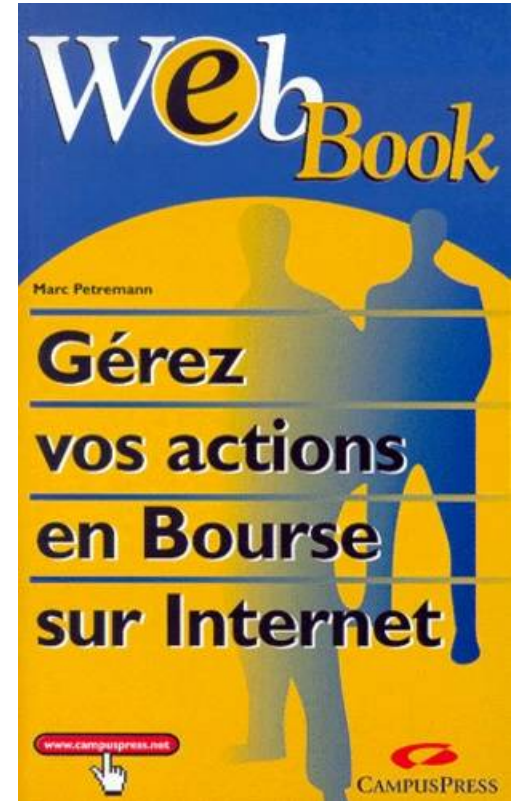
# FORTH & PHP

- Many books about Money



## FORTH & PHP

- My greatest success
- published in **100,000** copies



## FORTH & PHP

- I worked for more than 20 years for ORANGE
- I learned and mastered PHP



# FORTH & PHP

- In 2010 I built a website that explains programming in HTML, PHP, CSS....

```
function tracement($texte) {
    // Si on peut déterminer l'adresse IP
    $adresse_ip = Null;
    if(isset($_SERVER['REMOTE_ADDR'])) {
        $adresse_ip = ''.$_SERVER['REMOTE_ADDR'].'';
    }
    $txt_log=$adresse_ip.';'.date('d/m/Y H:i:s').';'.$texte."\n";
    // écriture dans un fichier de traçage
    $fichier = "tracement_".date('Ymd').".log";
    preg_match("^^(.*\V)([^\V]+)$", $_SERVER['SCRIPT_FILENAME'], $matches);
    $chemin_script = $matches[1];
    $fichierCible = $chemin_script.$fichier;
    $myFile=fopen($fichierCible,'a+');
    fputs($myFile,$txt_log);
    fclose($myFile);
}
```



## FORTH & PHP

- On this site, I fully exploit the GESHI library which highlights the source code of several languages: C, BASIC, PHP, HTML....

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Titre de ma page</title>
</head>
<body>
  <p>Bonjour tout le monde</p>
</body>
</html>
```

## FORTH & PHP

- But GESHI does not know how to manage the FORTH code...
- My various modifications of GESHI to manage the FORTH source code have always proven to be unsatisfactory...

## FORTH & PHP

- But first of all, the question arises: "why absolutely want to highlight the FORTH code"?

```
300 constant #BUFFER_SIZE           \ buffer size  
  
create  UART2-BUFFER                 \ define buffer for UART2  
  #BUFFER_SIZE allot  
  
Serial \ Select Serial vocabulary
```

# FORTH & PHP

- Code without highlighting :

```
-1 1 rshift invert constant high-bit
: odd? ( n -- f ) 1 and ;
: >>1 ( n -- n ) 1 rshift ;
: enmask ( n m -- n )
  0 -rot cell 8 * 1- for
    rot >>1 -rot
    dup odd? if
      over odd? if rot high-bit or -rot then
        swap >>1 swap
    then
    >>1
  next
  2drop
;
```

# FORTH & PHP

- Code with highlighting :

```
-1 1 rshift invert constant high-bit
: odd? ( n -- f ) 1 and ;
: >>1 ( n -- n ) 1 rshift ;
: enmask ( n m -- n )
  0 -rot cell 8 * 1- for
    rot >>1 -rot
    dup odd? if
      over odd? if rot high-bit or -rot then
      swap >>1 swap
    then
  >>1
next
2drop
;
```

# FORTH & PHP

- Known words have a hyperlink to the documentation for that word: :

**RSHIFT** ( x1 u -- x2 )

published: 9 January 2022 / updated 16 February 2022

[Lire cette page en français](#) 

Right shift of the value x1 by u bits.

## Example:

```
64 2 rshift . \ display 16
```

```
\ you can click to the links in this FORTH code to have online documentation
```

## FORTH & PHP

- The goal is to allow someone who has little knowledge of FORTH language to find the explanations of any known word in a single click.
- This avoids tedious searches in sometimes voluminous documentation...

## FORTH & PHP

- The heart of this mechanism is a single PHP function: **explode(...)**.
- This function transforms a character string into an array simply indicating the desired separator. In FORTH language, this will be the “**space**” character :
- `$lines = explode( "\n", str_replace("\r", "", $code));`



- My function explodeSource

```
/**
 * Explode FORTH source code in array of keywords
 * @param type $code
 * @return array
 */
private function explodeSource($code) {
    $lines = explode( "\n", str_replace("\r", "", $code));
    $keySource = array();
    foreach ($lines AS $line) {
        $keySource[] = explode(" ", $line);
    }
    return $keySource;
}
```

## FORTH & PHP

- The array returned by `explodeSource` contains an entry per word FORTH, but also an entry per space character...
- The next step therefore consists of taking the content of this table and making it an HTML code with the appropriate highlighting and hypertext links.

## FORTH & PHP

- This is what the **generateOutsource** function does :

```
foreach ($keySource AS $line) {
    foreach ($line AS $word) {
        if (strlen($word)==0) {
            $outSource .= " ";
        } else {
            $outSource .= $this->styliseWord($word);
        }
    }
}
```

- Highlighting and hyperlinks are handled by the **stylizeWord** function:

```
// test if $word is a FORTH word
foreach ($this->keywords AS $level => $listeMots) {
    if (in_array(strtoupper($word), $listeMots)) {
        return $this->linkifyWord($word, $level);
    }
}
return $word . " ";
```

## FORTH & PHP

- The **styleWord** function will search for each FORTH word in the keyWords table.

```
// test if $word is a FORTH word
foreach ($this->keywords AS $level => $listeMots) {
    if (in_array(strtoupper($word), $listeMots)) {
        return $this->linkifyWord($word, $level);
    }
}
return $word ." ";
```

## FORTH & PHP

- If the word FORTH exists in **\$keyWords**, it is highlighted.
- If the word FORTH does not exist in **\$keyWords**, it is returned without decoration.

```
compileoram          compileoram?          connect-flash
cos                   cos-coef          count           cr           create        ctype
cxor!                cycles          d-             d.            d/            d/mod
d<>                  d=            d>            d0<          d0=          d2*
decimal           deg-90to90    deg0to360     deg2rad      delay-cycles
dictionarynext       dictionarystart      digit          dint
```

# FORTH & PHP

- The PHP script was designed to retrieve FORTH words from a MySQL database.

id	word	word_comment	word_ansi	word_type	word_class	vocabularies	short_explanation	short_explanation_fr	explanation	explanation_fr
1	AND	n1 n2 -- n3	1			forth	Execute logic AND.	Effectue un ET logique.	The words <code>AND</code>, <code>OR</code>, and <...>	Les mots <code>AND</code>, <code>OR</code> et <cod...>
2	Serial	--	0	vocabulary		forth	Select serial vocabulary	Sélectionne le vocabulaire serial		
3	gpio_set_intr_type	pin type -- 0/err	NULL	NULL	interrupts		GPIO set interrupt trigger type.	Sélectionne le type de déclenchement d'une interromp...		<p><b>type</b> est une des constantes suivantes:</...>
4	F-	r1 r2 -- r3	NULL	arithmetic real		forth	Subtraction of two real numbers.	Soustraction de deux nombres réels.		
5	-	n1 n2 - n1-n2	1	arithmetic integer		forth	Subtract two integers.	Soustraction de deux entiers.		
6	,	x --	1			forth	Append x to the current data section.	Ajoute x à la section de données actuelle.	Stores an integer value at the address pointed to ...	Stocke une valeur entière à l'adresse pointée par ...
7	c,	c --	1			forth	Append c to the current data section.	Ajoute c à la section de données actuelle.		

# FORTH & PHP

- The data is managed from an administrator interface:

## Modification mot

\* mot FORTH

\* commentaire

+ ansi

+ type mot

+ type action

+ vocabulaires

---

+ résumé court (EN)



# FORTH & PHP

- This code :
  - `$this->keywords[0] = $Glossaire → getListeMotsOrdinaires();`
- Is similar as :
  - `$this->keywords[0] = array("DUP","DROP","SWAP" ....);`

## FORTH & PHP

- Keywords[0] contain normal words
- Keywords[1] contain definition words ( : ; variable...)
- Keywords[2] contain structures words ( do..loop if..else..then)
- Keywords[3] contain vars and const ( base...)

## FORTH & PHP

- Each word category is associated with a style:

```
0 => 'color: #0000ff;', // normal words
1 => 'color: red; font-weight: bold;" title="definition word"',
2 => 'color: black; background-color: yellow;', // structure words
3 => 'color: #ff00ff; font-weight: bold;', // constants
4 => 'color: black; font-weight: bold; background-color: #d7d7ff', // vocabularies
```

## FORTH & PHP

- Each category has a hyperlink:

```
var $outlink = array(  
    0 => 'help/index-esp32/word/',  
    1 => 'help/index-esp32/word/',  
    2 => 'help/index-esp32/word/',  
    3 => 'help/index-esp32/word/',  
    4 => 'help/index-esp32/word/',  
    5 => 'help/index-esp32/word/',  
    6 => 'help/reg-esp32/word/',  
);
```

## FORTH & PHP

- These links are defined in an MVC programming model operated by Zend Framework :
- <https://esp32.arduino-forth.com/help/index-esp32/word/constant>

## FORTH & PHP

- The full PHP code is available here:
- <https://github.com/MPETREMAN11/php/blob/main/Forth.php>
- Unlike GESHI's code, my PHP code takes **two pages!**

# FORTH & PHP -- conclusion --

- And here is how this PHP code is used on my sites explaining FORTH programming:

```
c
DC
#days
network
isplay
ftp
AC
campes
omponents
PI
WM
schRefs
:Ra
xiqueESP32forthV70715.phtml [-/A]
randLivreESP32forth.phtml [-/A]
xiqueESP32forthV70712.phtml [-/A]
xiqueESP32forthV70710.phtml [-/A]
ctendedESP32forth.phtml [-/A]
xiqueESP32forthV7079.phtml [-/A]
troduction.phtml [-/A]
2bitsForth.phtml [-/A]
rithFields.phtml [-/A]
igsManipulations.phtml [-/A]
farruptions.phtml [-/A]
ructures.phtml [-/A]

87 | <li><b>fileid</b> "File Identifier", un ticket pour un f
88 | </ul>
89 | <?php endif; ?>
90 | <h3 id='createFile'>Création d'un nouveau fichier</h3>
91 |
92 | <?php $str3 = <<<EOT
93 | -i value myfileFid
94 | : myfile ( -- )
95 |     s" /spiffs/myfile.fb"
96 |     2dup r/w open-file
97 |     if
98 |         drop r/w create-file
99 |         throw
100 |     else
101 |         nip nip
102 |     then
103 |     to myfileFid
104 | ;
105 | EOT;
106 | echo $Forth->sourceForView($str3); ?>
107 |
108 | <!--
109 | NOTE: This section describes one mechanism for running code at s
110 | The system will automatically attempt to mount SPIFFS filesystem
```

## FORTH & PHP -- conclusion --

- It's finish...
- Questions?