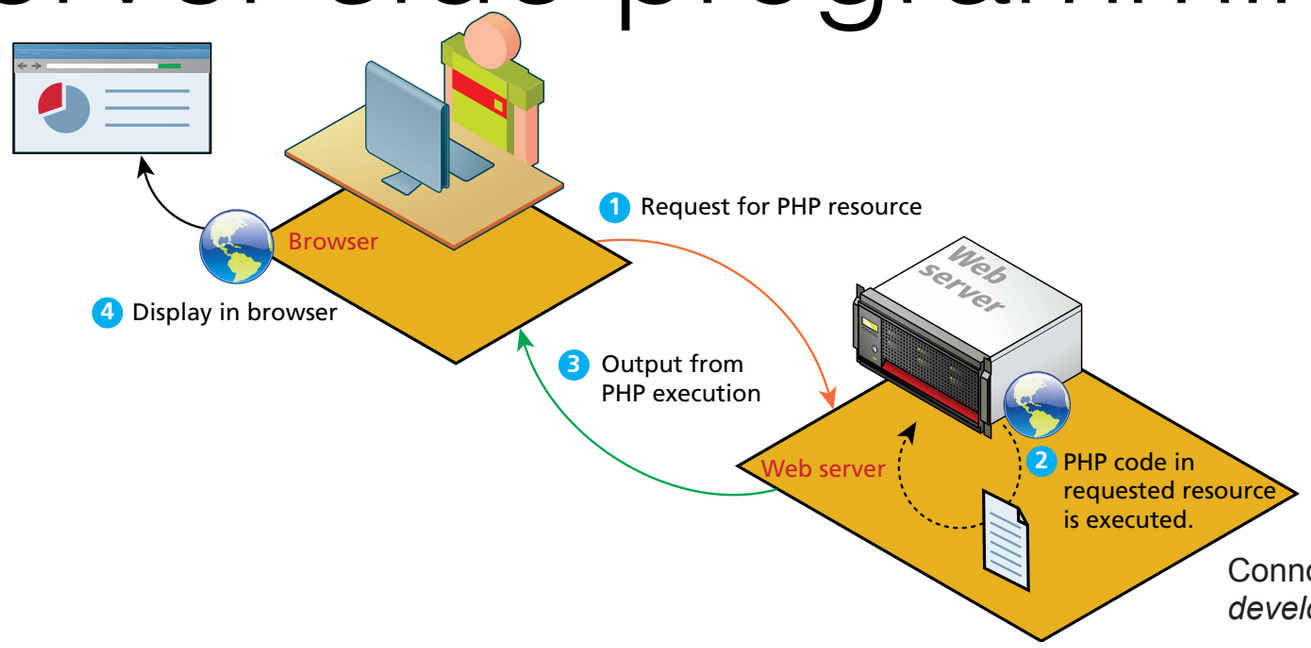
ΤΕΙ Κρήτης
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Θέματα Προγραμματισμού Διαδικτύου
~ PHP ~

Στελιος Σφακιανάκης
Εαρινό 2019

Server-side programming



Connolly, Randy. *Fundamentals of web development*. Pearson Education, 2015.

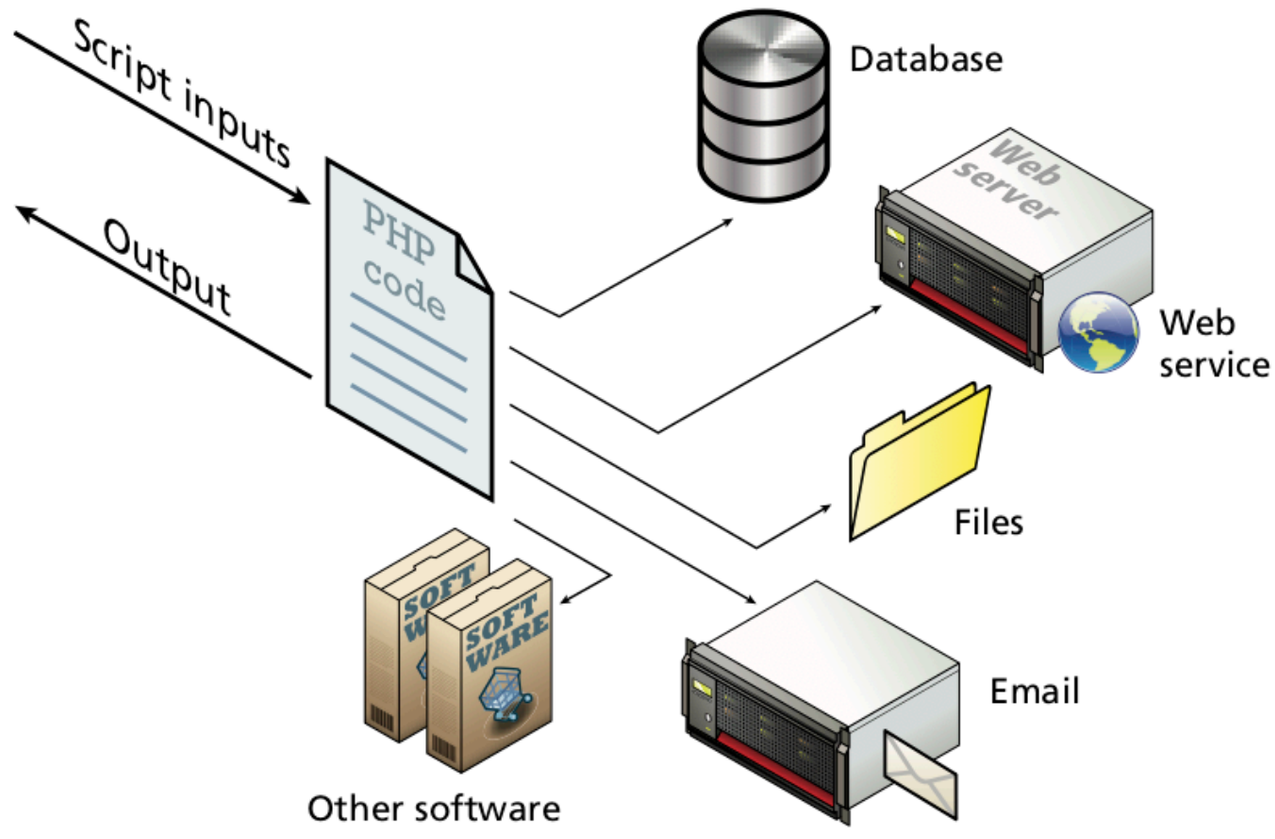
- Ο προγραμματισμός στην πλευρά του εξυπηρετητή (server-side programming) επιτρέπει τη δημιουργία **δυναμικών** σελίδων (σε αντιδιαστολή με τις στατικές σελίδες που είναι απλή HTML)
    - Παράδειγμα δυναμικής σελίδας: Μια σελίδα που δείχνει τις τρέχουσες κρατήσεις σε ένα εστιατόριο. Αυτή η σελίδα δεν μπορεί να είναι στατική γιατί οι κρατήσεις αλλάζουν συνεχώς.

- Ο **εξυπηρετητής ιστού** (web server) δέχεται κλήσεις από χρήστες (μέσω του browser) και εκτελεί εντολές -- γραμμένες σε σενάρια εντολών (**scripts**) -- οι οποίες παράγουν το τελικό HTML που στέλνεται στο χρήστη.

- Στη μεγάλη πλειοψηφία των δυναμικών σελίδων, το περιεχόμενο παράγεται από τα δεδομένα που αποθηκεύονται σε μια **βάση δεδομένων** στο server. Το script επομένως θα πρέπει να έχει εντολές με τις οποίες επικοινωνεί με βάσεις δεδομένων για να αποθηκεύσει, ανακτήσει, και να αλλάξει τα δεδομένα

Τι μπορεί να κάνει ένα script;

- Στον Server ένα script μπορεί να κάνει πάρα πολλά:
  - Να διαβάσει/γράψει στο σύστημα αρχείων
  - Να στείλει mail
  - Να επικοινωνήσει με άλλους servers και διαδικτυακές υπηρεσίες
  - Να γρά̈ψει/διαβάσει σε μια βάση δεδομένων
  - …



"PHP: Hypertext Preprocessor"

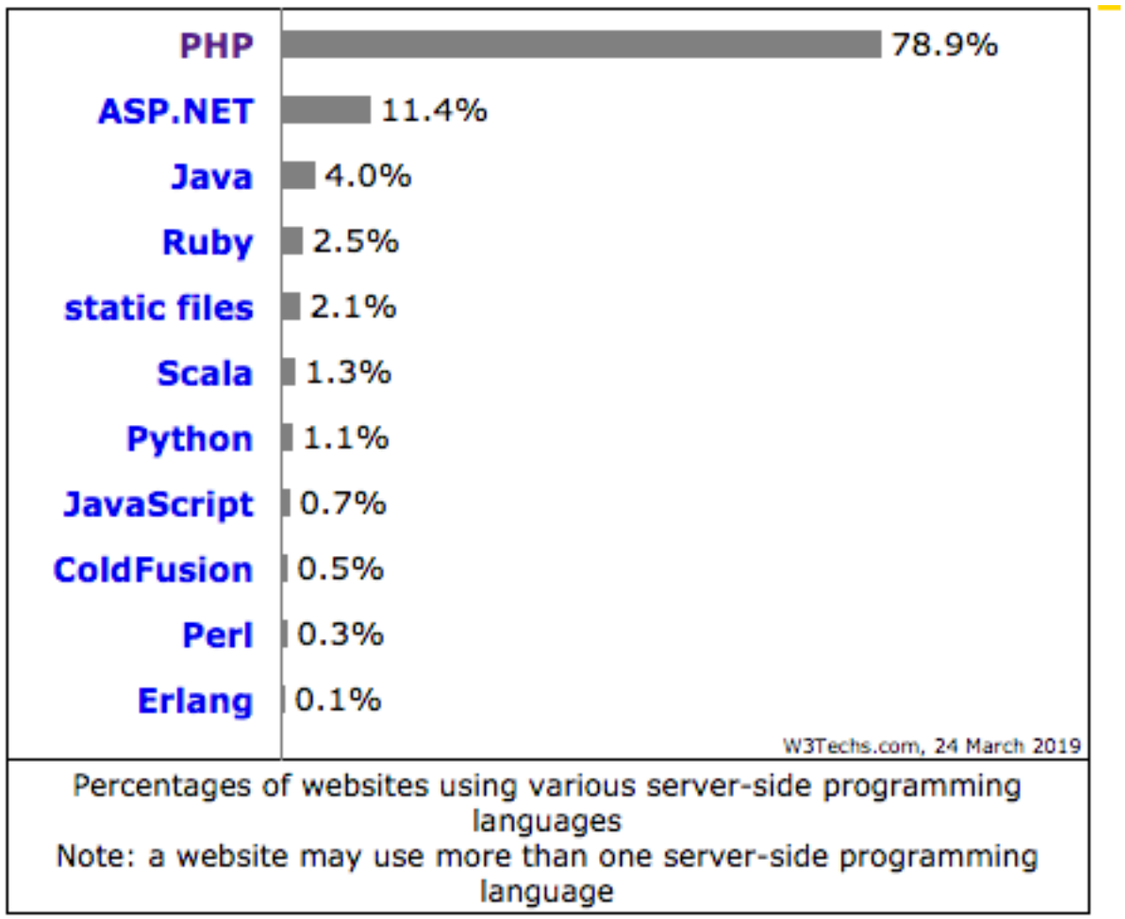
- Η πιο δημοφιλής γλώσσα προγ/μού για server-side programming (δείτε επόμενο slide)

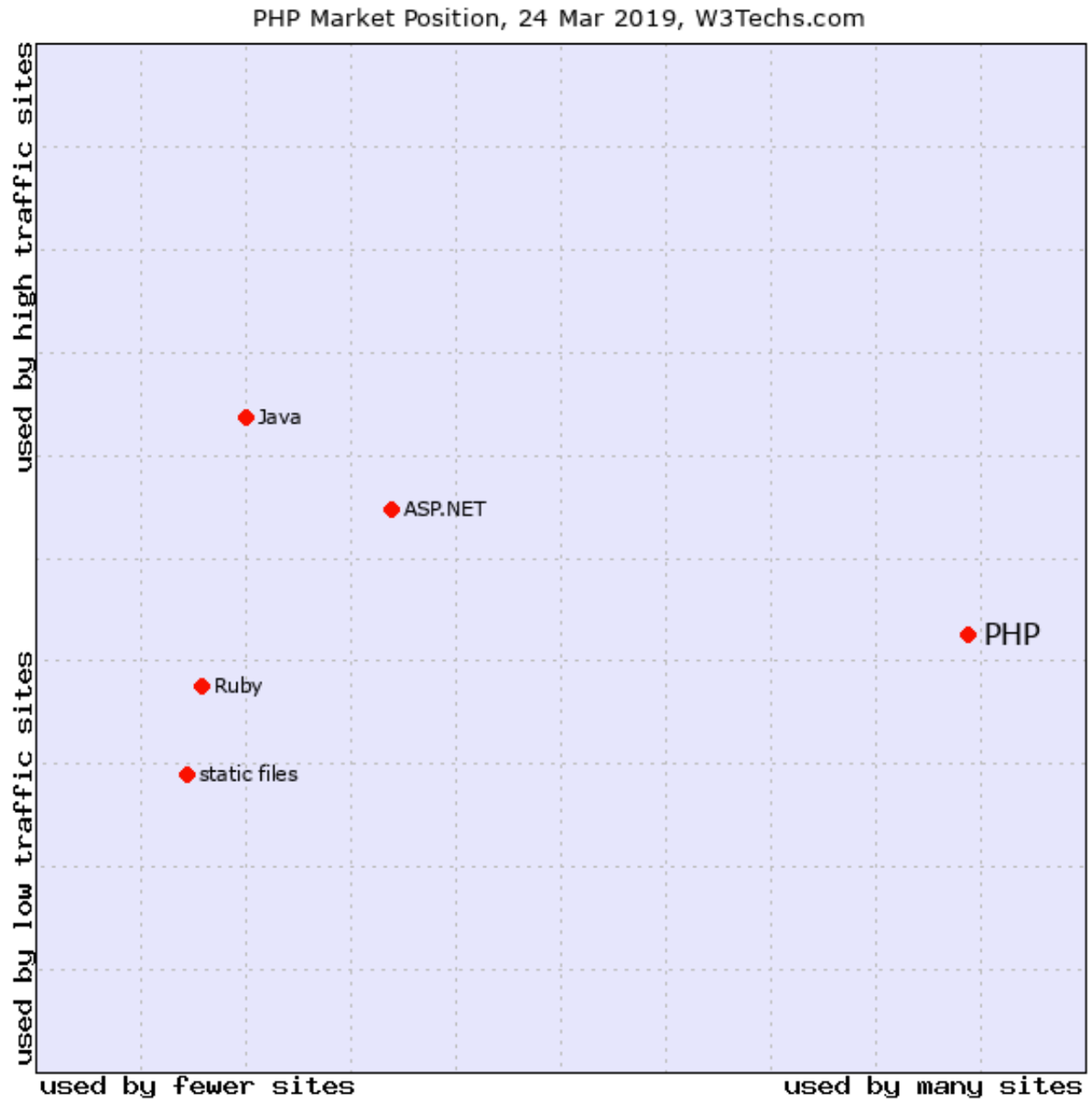
- Δημιουργήθηκε το 1994 για προσωπική χρήση από τον Rasmus Lerdorf, όντας φοιτητής, για προσωπική χρήση: "*Personal Home Page*"

- Έκτοτε είχε σταθερά ανοδική πορεία, τρέχουσα έκδοση : 7.3.3 ("stable" version)

- Τεκμηρίωση, νέα, downloads : https://www.php.net

~80% των ιστοχώρων χρησιμοποιεί PHP!



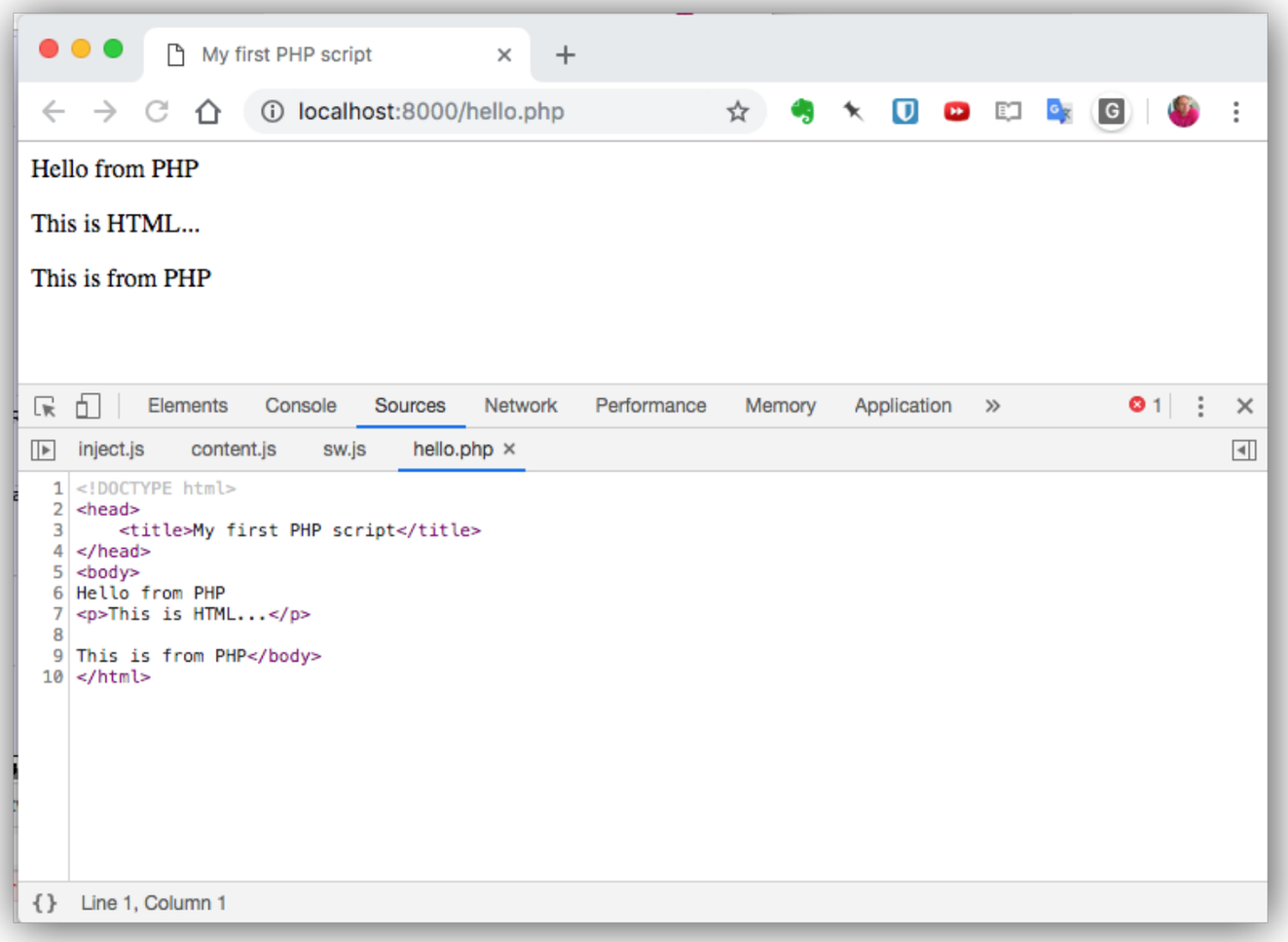


Δεδομένα Μαρτίου 2019, https://w3techs.com/technologies/details/pl-php/all/all

Βασικά στοιχεία προγ/μου PHP

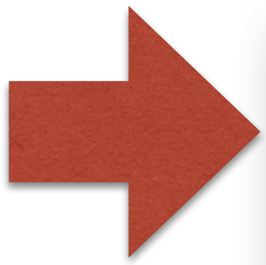
- Τα PHP scripts εισάγονται μέσα σε αρχεία με κατάληξη ".php"
- Μπορούμε να αναμείξουμε εντολές PHP και κώδικα HTML μέσα στο ίδιο script: **οι PHP εντολές γράφονται πάντα μέσα σε** `<?php` **και** `?>`
- Μεταβλητές εισάγονται με `$` (δολάριο). Τα ονόματα τους είναι case-sensitive, πρέπει να περιέχουν μονο γράμματα και αριθμούς αλλά μπορούν να ξεκινούν με την κάτω παύλα ("underscore"), π.χ. `$_myName`
- Σύνταξη βασισμένη στη γλώσσα C, π.χ.
  - Οι εντολές χωρίζονται με `;` (ερωτηματικό)
  - Σχόλια πολλαπλών γραμμών περικλείονται σε `/*` και `*/` ενώ μέχρι το τέλος της γραμμής εισάγονται με το `//`
  - Εντολές ελέγχου (π.χ. επανάληψη) όπως και στη C: `if`/`then`/`else`, `while`, `for`, `do`/`while`
- `echo` και `print` κάνουν "εκτύπωση"
  - Ό,τι τυπώνουν όμως εμφανίζεται ως HTML στον browser
- Στο ίδιο αρχείο PHP μπορούμε να έχουμε πολλά τμήματα με εντολές PHP, δείτε επόμενο παράδειγμα

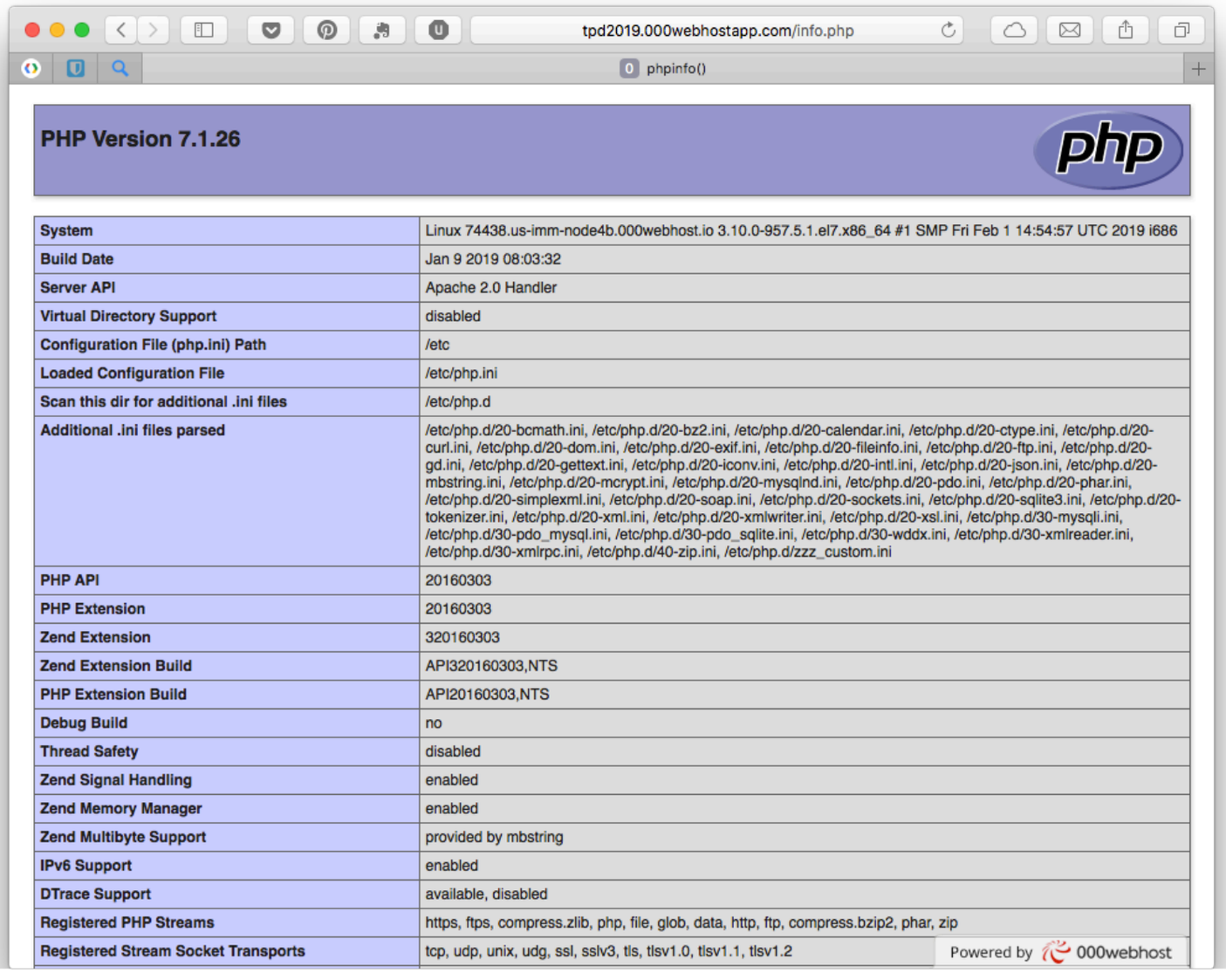
```php
<!DOCTYPE html>
<head>
    <title>My first PHP script</title>
</head>
<body>

<?php
    echo "Hello from PHP";
?>

<p>This is HTML...</p>

<?php
    echo "This is from PHP";
?>
</body>
</html>
```



Παίρνοντας πληροφορία για τον server και την έκδοση του PHP

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





- Δημιουργούμε ένα αρχείο με κατάληξη .php στον web host, π.χ `info.php`
- Βάζουμε την κλήση της `phpinfo` όπως φαίνεται παραπάνω
- Ανοίγουμε με τον browser το url που αντιστοιχεί σε αυτό το script

Τύποι Δεδομένων

- Η PHP είναι weakly-typed (π.χ. όπως η Python), δηλ. δεν χρειάζεται να δηλώνουμε ότι μια μεταβλητή έχει ακέραιο τύπο δεδομένων

- Κλασικοί τύποι δεδομένων:

  - Ακέραιοι, με μέγεθος τουλάχιστον 4 bytes (32 bits)

  - Λογικές (boolean) με τιμές `true` ή `false`

  - Πραγματικοί (δεκαδικοί, κινητής υποδιαστολής) "διπλής" ακρίβειας (double)

  - Αλφαριθμητικά (strings) μέσα σε μονά (`'`) ή διπλά εισαγωγικά (`"`) π.χ. `"PHP is great!"`

- Σύνθετοι τύποι : αντικείμενα (objects, δεν θα μας απασχολήσουν), και **arrays**!

Arrays

- Ένας πολύ χρήσιμος τύπος δεδομένων
    - Μια σειρά από "αντικείμενα" στα οποία μπορούμε να αναφερθούμε με βάση της "θέση" τους (index)
- Στην PHP τα arrays είναι επί της ουσίας ζευγάρια από keys και values ("αν δώσεις ένα key παίρνεις το αντίστοιχο value")
- Τα keys όμως εκτός από ακέραιοι μπορεί να είναι και strings!
    - Και τα values μπορεί να είναι οτιδήποτε, ακόμα και άλλα arrays!
- Εισάγεται με τη συνάρτηση "`array`" ή με `[...]` (από την έκδοση 5.4). Η σύνταξη `array("key" => "value")` δημιουργεί ένα array όπου το `key` γίνεται map στο `value`

```php
$fruits = array("μήλο", "πορτοκάλι", "μανταρίνι");

$fav_fruits = array("john" => "πορτοκάλι", "helen" => "μήλο");

echo $fruits[0]; // "μήλο"

echo $fav_fruits["john"]; // "πορτοκάλι"

$people = array("john" => array("first_name" => "John",
                                "surname" => "Doe", "age" => 25));
echo $people["john"]["surname"]; // "Doe"
```

Τελεστές

- Συνήθεις αριθμητικοί: `+` , `-`, `*`, `/`, `%` (υπόλοιπο διαίρεσης), `**` (ύψωση σε δύναμη)
  - Μοναδιαίοι αύξησης και μείωσης κατά ένα, π.χ. `$a++`, `--$a`
  - Και πολλοί περισσότεροι μέσω μαθηματικών συναρτήσεων: https://www.php.net/manual/en/ref.math.php
- Λογικοί, π.χ. `&&`, `||`, `!`
- Ένωση αλφαριθμητικών (strings) με την τελεία, π.χ. `"Hi " . "there"` -> `"Hi there"`
- Συγκρίσεις: `==` (ισότητα), `===` (ισότητα αλλά με ίδιο τύπο), `!=` , `!==`, `<`, `>`, `>=`, `=<`

★ Συμβουλευτείτε το https://www.w3schools.com/php/php_operators.asp

Συναρτήσεις για arrays

- `count`: επιστρέφει το μήκος του array
- `array_push`: εισάγει ένα (ή περισσότερα) στοιχεία στο τέλος ενός array
- `array_slice`: επιστρέφει ένα υπο-array (τμήμα του αρχικού array)
- `sort` : ταξινόμηση των στοιχείων του array
- Δείτε τα στο https://www.php.net/manual/en/function.array.php

"foreach"

- Εντολή επανάληψης
- Πολύ βολικό όταν θέλουμε να κάνουμε κάτι σε κάθε στοιχείο ενός array.. χωρίς να γνωρίζουμε το μήκος του array!
- Δείτε το παράδειγμα:

```php
$fruits = array("μήλο", "πορτοκάλι", "μανταρίνι");
// Ο κλασσικός τρόπος:
for($i=0; $i < count($fruits); $i++) {
  echo $fruits[$i];
}

// Το ίδιο με το foreach:
foreach ($fruits as $i => $value) {
  echo $value;
}
```

Ενσωμάτωση μεταβλητών σε strings

- Υπάρχει διαφορά μεταξύ των μονών (`'`) και διπλών (`"`) εισαγωγικών για strings:

- Με τα διπλά εισαγωγικά (`"..."`) μπορούμε να ενσωματώσουμε μεταβλητές έτσι ώστε να αντικατασταθούν με την τιμή τους

- Αν έχουμε πολύπλοκες εκφράσεις μπορούμε να τις βάλουμε μέσα σε αγκύλες `{ }`.

- Δείτε το παρακάτω παράδειγμα:

```php
$name = "John";
echo 'Hello $name';  // "Hello $name"
echo "Hello $name";  // "Hello John"

$arr = array("name" => "John");
echo "Hello {$arr["name"]}!"; // Hello John!
```

Χρήσιμα (κυρίως για debugging)

- `var_dump` : τυπώνει πληροφορία για μια μεταβλητή (π.χ. τον τύπο δεδομένων της)
- `print_r` : τυπώνει την τιμή μιας μεταβλητής σε πιο κατανοητή μορφή
- `die("...")` : προκαλεί τερματισμό του script με ένα μήνυμα λάθους

```php
<?php
$a = array(100, 200, "name"=> array("John", "Doe"));
var_dump($a);
?>
```

```
array(3) {
  [0]=>
  int(100)
  [1]=>
  int(200)
  ["name"]=>
  array(2) {
    [0]=>
    string(4) "John"
    [1]=>
    string(3) "Doe"
  }
}
```

```php
<?php
$a = array(100, 200, "name"=> array("John", "Doe"));
print_r($a);
?>
```

```
Array
(
    [0] => 100
    [1] => 200
    [name] => Array
        (
            [0] => John
            [1] => Doe
        )

)
```

Συναρτήσεις Strings

- `strlen` : μήκος ενός string
- `strrev` : αντιστροφή string
- `str_replace` : αντικατάσταση ενός string με κάτι άλλο μέσα σε ένα μεγαλύτερο string
- `implode`/`explode`: ένωση/διαχωρισμός στοιχείων array σε string με κάποιο delimiter
- Και πολλά περισσότερα: https://www.php.net/manual/en/ref.strings.php

```php
echo strrev("robot"); // "tobor"
echo str_replace("John", "Helen", "Hello John!"); // Hello Helen!

$names = array("john", "helen", "george", "maria");
echo implode(", ", $names); // "john, helen, george, maria"
```

Συναρτήσεις Χρήστη

- Μπορείτε να ορίσετε δικές σας συναρτήσεις με το `function`
- Επιστρέφουμε τιμές με το `return`
- Μεταβλητές που παίρνουν τιμές μέσα σε συναρτήσεις έχουν "τοπικό" scope (local scope), δείτε παράδειγμα παρακάτω:

```php
function sayHi($name) {
  echo "Hello $name!";
}

function addOne($a) {
  return $a + 1;
}
```

```php
$a = 5;
function f() {
  $a = 6;
  echo "inside function, a=$a";
}
f(); // Τυπώνει 6
echo "outside function, a=$a"; // Τυπώνει 5
```

Include και Require

- Πολλές φορές είναι προτιμότερο να έχουμε τον κώδικα μας μοιρασμένο σε περισσότερα του ενός αρχεία.

  - Π.χ. μπορούμε να έχουμε τις συναρτήσεις μας σε ένα *functions.php* αρχείο το οποίο να (επανα)χρησιμοποιείται από άλλα αρχεία PHP

- Με την εντολή `include 'functions.php'` μπορούμε να ενσωματώσουμε τις εντολές του *functions.php* σε άλλο αρχείο

- Το `require` είναι όπως το `include` αλλά αν αποτύχει τότε τερματίζεται το script!