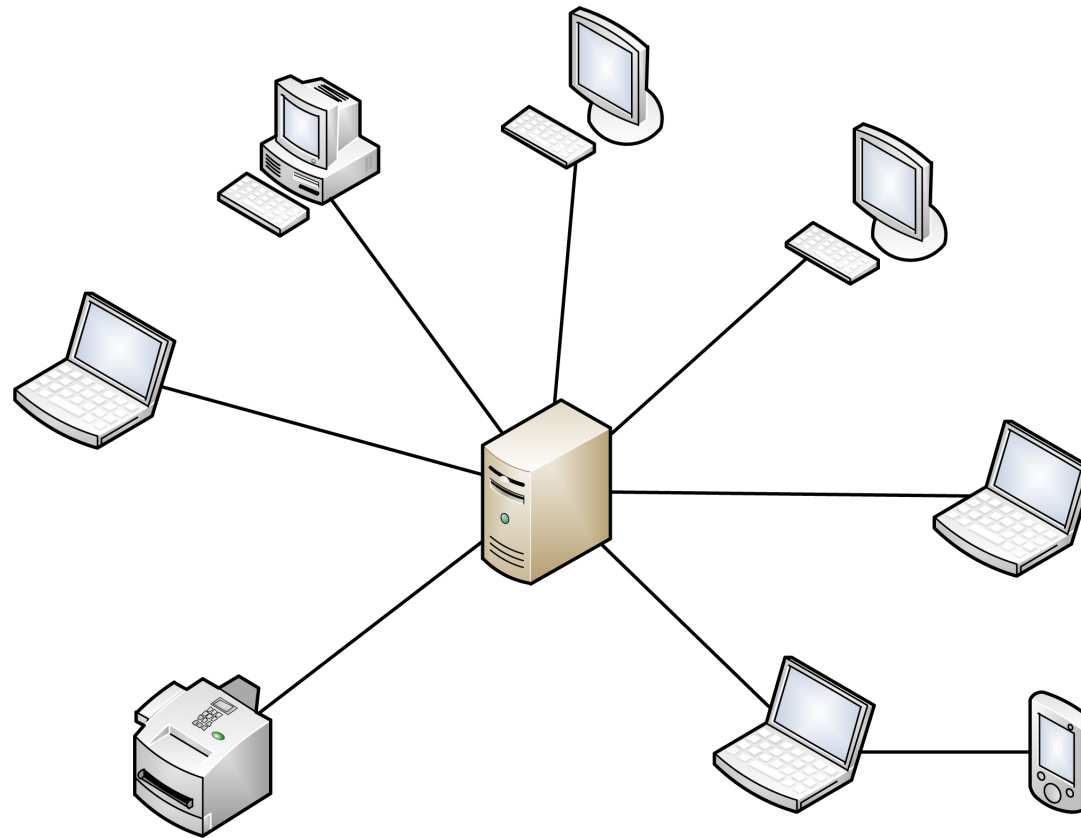
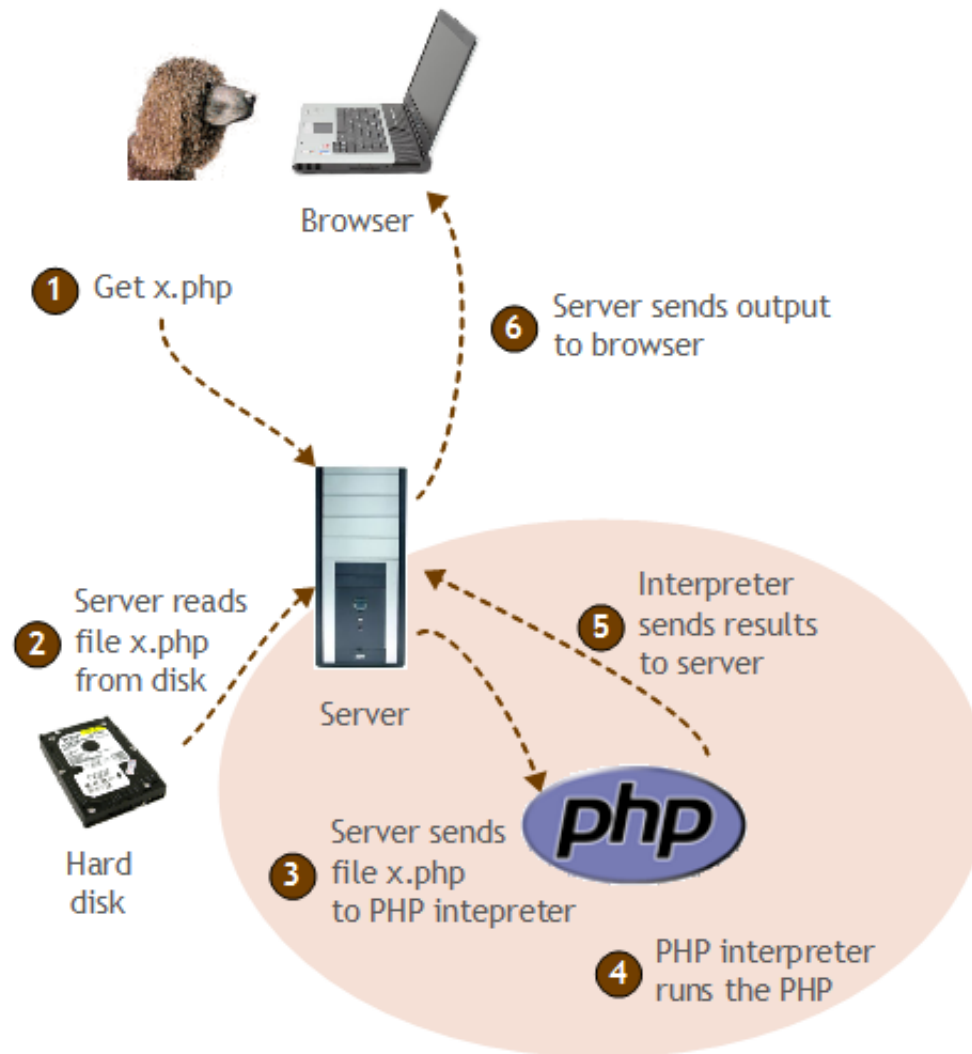


01. Динамично генериране на HTML

Клиент-сървър



PHP



PHP

- Браузъра изпраща HTTP заявка към сървъра.
- Сървъря я обработва.
- Той може да прочете информация от база от данни.
- Накрая генерира отговор под формата на HTML документ.

Пример за PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hi, I'm a PHP script!";
    ?>
  </body>
</html>
```

Пример за PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    Hi, I'm a PHP script!
  </body>
</html>
```

Пример за PHP

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "<h2>Hi, I'm a PHP script!</h2>";
    ?>
  </body>
</html>
```

Пример за PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <h2>Hi, I'm a PHP script!</h2>
  </body>
</html>
```


Променливи

```
<?php
    $var = 'Bob';
    $Var = 'Joe';
    echo "$var, $Var";           // outputs "Bob, Joe"
?>
```

Типове в PHP

```
<?php
    $myAge = 16;                // a PHP Integer - always available
    $yourAge = 15.5;            // a PHP Float - always available
    $hasHair = true;            // a PHP Boolean - always available
    $greeting = "Hello World!"  // a PHP String - always available

    $person = new Person();     // a PHP Person - available NOW!
?>
```

Оператори за сравнение

Example	Name	Result
<code>\$a == \$b</code>	Equal	TRUE if \$a is equal to \$b after type juggling.
<code>\$a === \$b</code>	Identical	TRUE if \$a is equal to \$b, and they are of the same type.
<code>\$a != \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a <> \$b</code>	Not equal	TRUE if \$a is not equal to \$b after type juggling.
<code>\$a !== \$b</code>	Not identical	TRUE if \$a is not equal to \$b, or they are not of the same type.
<code>\$a < \$b</code>	Less than	TRUE if \$a is strictly less than \$b.
<code>\$a > \$b</code>	Greater than	TRUE if \$a is strictly greater than \$b.
<code>\$a <= \$b</code>	Less than or equal	TRUE if \$a is less than or equal to \$b.

\$a	to	
\$a >= \$b	Greater than or equal to	TRUE if \$a is greater than or equal to \$b.

Арифметичні оператори

Example	Name	Result
$-\$a$	Negation	Opposite of $\$a$.
$\$a + \b	Addition	Sum of $\$a$ and $\$b$.
$\$a - \b	Subtraction	Difference of $\$a$ and $\$b$.
$\$a * \b	Multiplication	Product of $\$a$ and $\$b$.
$\$a / \b	Division	Quotient of $\$a$ and $\$b$.
$\$a \% \b	Modulus	Remainder of $\$a$ divided by $\$b$.

Арифметика

```
<?php
    echo (5 % 3)."\n";    // prints 2
    echo (5 % -3)."\n";   // prints 2
    echo (-5 % 3)."\n";   // prints -2
    echo (-5 % -3)."\n";  // prints -2
?>
```

if

```
<?php
    if ($a > $b) {
        echo "a is bigger than b";
    } elseif ($a == $b) {
        echo "a is equal to b";
    } else {
        echo "a is smaller than b";
    }
?>
```

While

```
<?php
    $i = 1;
    while ($i <= 10) {
        echo $i++;
    }
?>
```


For

```
<?php
    for ($i = 1; $i <= 10; $i++) {
        echo $i;
    }
?>
```

Масиви

В PHP има само една структура от данни - масив.

```
<?php
$array = array(
    "foo" => "bar",
    "bar" => "foo",
);
?>
```

Foreach

Обхожда последователно елементите на колекция.

```
<?php
    $arr = array(1, 2, 3, 4);
    foreach ($arr as $value) {
        echo $value;
    }
?>
```

Foreach

```
<?php
    $a = array(
        "one" => 1,
        "two" => 2,
        "three" => 3,
        "seventeen" => 17
    );

    foreach ($a as $k => $v) {
        echo "$k -> $v\n";
    }
?>
```

Функции

```
<?php
    function foo($arg_1, $arg_2, /* ..., */ $arg_n) {
        echo "Example function.\n";
        return $retval;
    }
?>
```

Функции

```
<?php
function takes_array($input) {
    echo "$input[0] + $input[1] = ", $input[0] + $input[1];
}
takes_array(array(1, 2)); // 1 + 2 = 3
?>
```

Функции

```
<?php
function makecoffee($type = "cappuccino")
{
    return "Making a cup of $type.\n";
}
echo makecoffee(); //Making a cup of cappuccino.
echo makecoffee(null); //Making a cup of .
echo makecoffee("espresso"); //Making a cup of espresso.
?>
```

Дебъгване в php

- `var_dump($var, $var2...)`
- `print_r($var)`
- `exit()/die()`
- [XDEBUG](#)

Задача

- Система за събиране на мнения за избираеми дисциплини.
- Всяка избираема дисциплина си има страница.
- На страницата има информация за дисциплината и коментари за нея.
- През семестъра ще създадем такава система от начало до край.
- Всяко упражнение ще подобрява системата.

Задача 1

Имплементирайте функцията `show_page`, която показва информацията за даден избираем курс. Тя приема два аргумента. Първия е `pageId`, който е идентификатора на курса който искаме да покажем. Втория е масив с информация за курсовете.

Информацията за курсовете се пази в следния масив:

```
<?php
$data = array(
    'webgl' => array(
        'title' => 'Компютърна графика с WebGL',
        'description' => '...',
        'lecturer' => 'доц. П. Бойчев',
    ),
    'go' => array(
        'title' => 'Програмиране с Go',
        'description' => '...',
        'lecturer' => 'Николай Бачийски',
    )
);
?>
```

Задача 1

Пример, следното извикване:

```
<?php  
    show_page( 'webgl' , $data );
```

Тя трябва да върне(не да изпринтира) следния низ:

```
<h1>Компютърна графика с WebGL</h1>  
<h2>доц. П. Бойчев</h2>  
<p>...</p>
```

Като в h1 заглавие се намира 'title' информацията, в h2 заглавие се намира 'lecturer' информацията, а в параграф се показва 'description' информацията за курса с ключ, съвпадащ с параметъра \$pagelId.

В предаденото решение не трябва да има нов ред или празно място между html таговете.

[Пример за краен резултат](#)

Задача 2

Имплементирайте функцията `show_nav`, която показва меню за навигация между отделните курсове. Тя приема два аргумента. Първият е масив с информация за курсовете. Той е същият, като в предишната задача. Вторият аргумент е текущият курс, който се преглежда в момента.

[Пример за краен резултат](#)

Задача 2

Ако извикаме функцията по следния начин:

```
<?php  
show_nav($data, 'webgl');
```

Тя трябва да върне(не да изпринтира) следния низ:

```
<nav>  
  <a href="?page=webgl" class="selected">Компютърна графика с WebGL</a>  
  <a href="?page=go">Програмиране с Go</a>  
</nav>
```

Хиперлинковете към кусовете трябва да се съдържат в nav секция. Всеки хиперлинк трябва да води към съответната страница на курс и за текст на хиперлинка трябва да се изписва името на курса.

Ако адреса на хиперлинка съвпада с текущо отворения курс, подаден като втори параметър, към хиперлинка трябва да се добави клас selected, както е показано в примера.

В предаденото решение не трябва да има нов ред или празно място между html таговете.

Въпроси

- <http://moodle.openfmi.net/course/view.php?id=1911>