Wordpress, dinamikus weblapok és php

Eddigi előadások

- Eddigi előadások letöltési linkjei
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/01.zip
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/02.zip
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/03.zip
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/04.zip
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/05.zip
- https://dl.dropboxusercontent.com/u/93156461/!deleteme/!Tanfolyam/06.zip

Információtechnológia (IT)

- Az információtechnológia követhetetlenül gyorsan fejlődik, ezért a piaci igények abszurddá váltak
- Ahhoz, hogy valaki a szakmában versenyképes legyen folyamatos tanulás, tájékozódás szükséges
- Minden programozási nyelv folyamatosan fejlődik, átalakul
- Minden nyelv naprakész tudását várják el a cégek, munkáltatók azért, hogy versenyképesek legyenek az óriási konkurenciákkal szemben
- Ebben a nehéz szakmában egyre nehezebb kiemelkedni, ezért tegyünk meg mindent azért, hogy sikeresek legyünk
- Sajnos eléggé kevés óránk van, megpróbálok minél több hasznos tudást átadni ilyen rövid idő alatt, ezért sokszor sietnem kell. Az legyen a szemünk előtt, hogy az információtechnológiában óriási a rohanás és nem lehet mindent tökéletesen tudni, de azért törekedni kell erre
- Tehát az igazi elsajátításhoz nagyon sok gyakorlásra, tanulásra van szükség, sokat lehet abból is tanulni, ha példakódokat megértünk, módosítunk

Wordpress telepítés

- A wordpress telepítéséről még egyszer összefoglalva: phpmyadmin-ban hozzunk létre egy új adatbázist *wordpress* néven
- Ha Wampservert használunk, akkor a wordpress telepítőjét (.zip) bontsuk ki a www mappába
- Nyissuk meg a telepítőt a böngészőnkben (ha sajátgépen van a szerver, akkor a cím: localhost)
- Telepítés során adatbázis neve: wordpress, user: root, pass: nincs, database host: localhost (vagy inkább a külső IP címünk, webszerverünk külső IP címe/domén nevünk)
- Utólag mindig átírhatjuk a weboldalunk URL elérését (database host) az: admin felületen Settings->General-nál a két URL beállítást írjuk át (a külső IP címünkre, webszerverünk külső IP címere/domén nevünkre), ezt fontos pontosan beállítani, mert különben a wordpress oldalunk nem fog jól megjelenni másoknak

Wordpress Page Builder

- Rakjuk fel a Responsive By CyberChimps.com témát
- És a Page Builder by SiteOrigin plugint
- Hozzunk létre egy új oldalt (Pages beállítás) váltsunk át a Page Builder szerkesztő fülre
- Adjunk hozzá új widgetet itt válasszuk a Layout Buildert, ezen belül hozzunk létre Text-eket és a sorok elrendezését külön beállíthatjuk a csavarkulcs ikonnal
- Az oldal alapsémáját beállíthatjuk a Page Attributes résznél, ott válasszuk ki a kívánt Templatet, pl full width page (no sidebar)

Wordpress Theme Customize

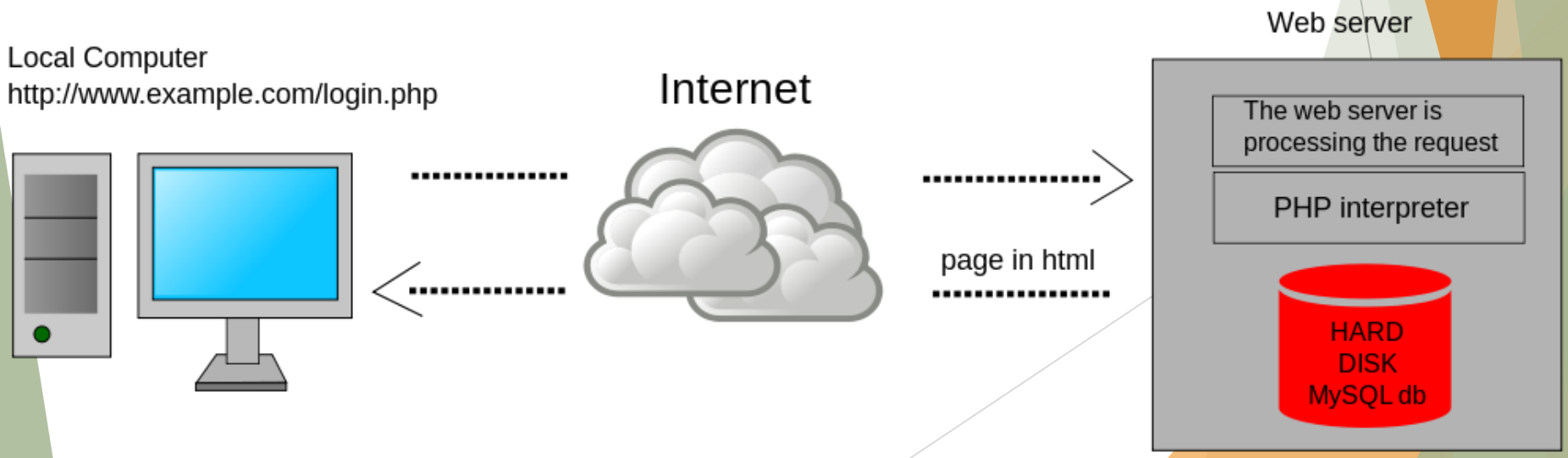
- Appearance beállításnál válasszuk a Customizet
- Itt a Site Identitynél megadhatjuk a weboldalunk főcímét és egy leírást hozzá
- Theme Elementsnél eltűntethetjük pl az aloldal útvonalát (breadcrumb list)
- Home page: kezdőoldal
- Megadhatunk saját CSS kódot, php skriptet is akár
- Színeket is beállíthatunk
- Képeket is könnyen beszúrhatunk (ez a Media Libraryba kerül)
- Wordpress admin felületnél ha az Appearance Widgets részre megyünk, ott átállíthatjuk a különböző sávok, kitüntetett dobozok összeállítását

Wordpress forráskód

- A Wordpress igazi ereje abban rejlik, hogy minden forráskódhoz hozzáférhetünk és minden átírhatunk
- A Wordpress a PHP szkriptnyelvet használja a weboldalak generálására
- Hozzáférhetünk a CSS téma stíluskódhoz is
- Appearance->Editor résznél elérhetjük és módosíthatjuk a forrásfájlokat

Dinamikus weblapok

- A dinamikus weblapok pl. honlapmotor segítségével generálódnak a szerveren lekérési paraméterek és jelenlegi állapot alapján. A kéréseket (request) a kliens küldi a szerverre és így jön létre az interneten keresztül a kommunikáció.
- A dinamikus weblapok kimenete egy legenerált html weblap, aminek a konkrét html forráskódja nem található meg a szerveren!
- A generálás szkriptnyelvek (javascript, PHP) segítségével történik



A szkriptnyelvek

- A szkriptnyelvek elemi utasítások sorozatából állnak, így algoritmusokból álló programot kapunk
- Algoritmusok segítségével ugyanaz az eredmény, kimenet rövidebb forráskódot igényel mint a leíró nyelvekben, így a számítógép programozási problémákhoz közelebb kerülünk
- Nincs olyan sok kulcsszó, mint a leíró nyelvekben
- Ezért könnyebben karbantartható, strukturáltabb a programkód, nincs olyan sok mellékhatás, könnyebben megérthetjük a működését a kódunknak, kevesebb a hibázási lehetőség
- Ilyen nyelvek pl. JavaScript, PHP

A PHP szkriptnyelv

- A PHP (Hypertext Preprocessor) általános szerveroldali szkriptnyelv dinamikus weblapok készítésére
- A szkriptkód akár html nyelvbe is ágyazható, fájlkiterjesztése .php
- A hagyományos HTML lapokkal ellentétben a szerver a PHP-kódot nem küldi el a kliensnek, hanem a szerver oldalán a PHP-értelmező motor dolgozza fel
- Az esetleges PHP-kódok kimeneteivel a megadott HTML elemekkel együtt html weblap formájában küldi a szerver a kliensnek
- A kódok végezhetnek adatbázis-lekérdezéseket, automatizált generálási feladatokat, létrehozhatnak képeket, fájlokat olvashatnak és írhatnak, kapcsolatot létesíthetnek távoli kiszolgálókkal

A PHP szintaxis

- Minden PHP szkriptkód *<?php* kezdődik és *?>* végződik
- Az elnevezéseknél (fájlnévnél, változóneveknél) kerüljük az ékezetes betűket
- Minden utasítás pontosvesszővel van lezárva
- Az utasítások kiértékelése sorfolytonosan történik
- Kommentelés történhet: // vagy # egysoros komment elején vagy /* … */ több soros komment esetén. Ezeket használjuk a saját kódunk magyarázatához, hogy mások is megértsék a kódunkat és hogy mi is tudjuk később mit akartunk csinálni. Kommenteket használhatjuk kódrészek kihagyásához is, hibakereséskor pl.
- Az igényes forráskód megkövetel bizonyos konvenciókat, legyünk következetesek a változók elnevezésénél, használatánál (pl. változók mind kisbetű, konstansnevek mind nagybetű)

PHP változók

- Változóneveket: $valtozonev=érték; segítségével hozhatunk létre (definiálhatunk), amiknek adunk egy kezdőértéket, ez lehet szám, vagy akár karakterlánc (string), így a változónevek ezeket a változóértékeket fogják tárolni
  - $x = 10;
    $y = „Bela”;
- A változónév a $ után betűvel kell kezdődjön és csak a következő karaktereket tartalmazhatja: (A-z, 0-9, and _)
- Számot tartalmazó változókkal összeadhatunk, kivonhatunk, szorozhatunk, oszthatunk, majd az eredményeket vagy új változóban, vagy valamelyik létező változóban tároljuk
- A változók értékéhez hozzáférhetünk, ha leírjuk a változónevét ($valtozonev) és ezt az értéket felhasználhatjuk utasításokban
  - $x = 10;
    $y = 20;
    $z = $x + $y; //Az eredmeny 30 lesz

PHP echo kiíratás

- A kiíratás utasítása: echo $valtozonev;
  vagy: echo „Hello Vilag!”;
- A PHP kimenete HTML-ként lesz értelmezve, ezért HTML tageket is megadhatunk az echo-nak:
  - echo „Elso sor
Masodik sor”;
- A karakterláncokat (stringeket) pont segítségével köthetjük össze esetleg változókkal (pl: echo „Az autom szine:”.$color;)
- A PHP megengedő programozási nyelv (szkriptnyelv), ezért az adattípusok értelemszerűen kezelődnek
- Ezért pl ezt:
  - echo $x."+".$y."=" . $z;
- írhatjuk így is:
  - echo "$x + $y = $z";

PHP függvényhívás

- A CSS-hez hasonlóan történik:
  - function függvénynév($paraméter1,$paraméter2…);
- Létrehozhatunk saját függvényeket így:
  - function fuggvenynevem($param1,$param2) {
    $belsovaltozo=1;
    echo „Megadott parameterek: $param1 és $param2 , belso valtozom: $belsovaltozo”
    }
    //Fuggveny hivasa a főszkriptben:
    fuggvenynevem(„Egyik parameter”,”Masik paramter”);
- Vannak beépített PHP függvények is (ezek a függvénynevek foglaltak)

PHP változó érvényességi határ

- Alapvetően háromféle érvényességi határ (variable scope) létezik:
- Lokális (local) érvényessége akkor van, ha egy függvényen belül definiáltunk változót, akkor ezt a változót csak a függvényen belül lehet elérni, ezen kívül nem elérhető
- Globális (global) érvényessége akkor van, ha bármilyen konkrét függvényen kívül definiáltunk egy változót (tehát a <?php … ?> között függvényen kívül), ekkor a szkriptben bárhol elérhető a változó, de ha egy függvényben szeretnénk használni globális változót, akkor sajnos oda külön definíciót kell írni, hogy elérhető legyen azon a függvényen belül, ezt kell beírni:
  - global $egyikvaltozo,$masikvaltozo;
- Statikus (static) érvényessége akkor van, ha a változónk típusa static a függvényünkön belül:
  - static $valtozonev=0;
- Ekkor ez a változónk megjegyzi az értékét a függvényen belül, így ha újra meghívjuk a függvényt, akkor nem áll vissza $valtozonev 0-ára (mint lokális esetben), hanem annyi lesz amennyi a legutóbbi függvényhívás végén volt

PHP adattípusok

- PHP String
  - Karakterlánc „…” vagy ‘…’ között
- PHP Integer
  - Egész decimális szám:
    -2,147,483,648 és 2,147,483,647 között
- PHP Float
  - Tizedesjegyű szám: $x = 10.365;
- PHP Boolean
  - Igaz/Hamis értékek: $x = true;
- A PHP var_dump() függvény segítségével megtudhatjuk a változónk adattípusát:
  - $x = 10.365;
    var_dump($x); //Kimenet: float(10.365)

PHP string

- Karakterlánc típusú változókkal sok mindent csinálhatunk akár beépített függvények segítségével is
- Ha pl a változónk: $hello = "Hello world!";
- Változó karakterszáma:
  - echo strlen($hello); // outputs 12
- Szavak száma:
  - echo str_word_count($hello); // outputs 2
- Szöveg megfordítása:
  - echo strrev("Hello world!"); // outputs !dlrow olleH
- Keresett szó pozíciója:
  - echo strpos("Hello world!", "world"); // outputs 6
- Szó kicserélése:
  - echo str_replace("world", "Dolly", "Hello world!");
    // outputs Hello Dolly!

PHP konstans

- Konvenció szerint a konstans változóneveink nagybetűsek legyenek (programozói szokás)
- PHP-ban konstans létrehozása:
  define(„KONSTANSNEV", “KONSTANS ERTEK");
- Ezek olyan globális változók, amiket mindenhol elérhetünk global kulcsszó nélkül is
- Pl:
- define("GREETING", "Welcome to W3Schools.com!");

```
function myTest() {
    echo GREETING;
}

myTest();
```

PHP adattömb

- Az adattömbök (array) definiálásakor egy adattömbnévhez rendelünk több értéket:
  - $cars = array("Volvo", "BMW", "Toyota");
- Az adattömbnév értékeit indexeléssel érhetjük el
- Az indexelés 0-tól kezdődik!!
  - echo $cars[0]; //Volvo
    echo $cars[1]; //BMW
    echo $cars[2]; //Toyota
- Itt most lényegében index=>érték párosokat definiáltunk, programozási-elméletben ezt úgy hívjuk, hogy: index kulcsokhoz (key) rendelünk értékeket, így létrehozhatunk asszociatív tömböket is akár:
  - $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
- Elérés index helyett kulccsal történik:
  - echo $age['Peter']; //35
    echo $age['Ben']; //37
    echo $age['Joe']; //43
- Adattömb értékei számának a meghatározása beépített függvény használatával: count($cars);

PHP operátorok, műveletek

- A különböző változókat bizonyos szempontok szerint összehasonlíthatunk, vagy akár számolhatunk is velük műveletek segítségével pl: $z=$x+$y;
- Minden művelet valamilyen eredményt (értéket) visszaad, amit felhasználhatunk pl értékadásnál: $z=$x+$y , az eredmény $z-ben lesz
- Változók összehasonlítása során egy logikai értéket kapunk vissza, amit feltételes elágazásoknál felhasználhatunk pl
  - $x == $y //true értéket ad vissza, ha a két változó megegyezik
  - $x != $y //true értéket ad vissza, ha a két változó nem egyezik meg
  - $x > $y (x nagyobb mint y) vagy $x < $y vagy
    $x >= $y (x nagyobb egyenlő mint y) vagy $x <= $y
- Növelő, csökkenő egyváltozós operátorok:
  - Előnövelő/csökkenő: ++$x vagy --$x
    Növeljük/csökkentjük a változót, majd visszaadjuk az értéket
  - Utónövelő/csökkenő: $x++ vagy $x--
    Visszaadjuk az értéket, majd növeljük/csökkentjük a változót

PHP logikai operátorok

- A különböző bool változók, vagy összehasonlítások során true/false értékeket kapunk vissza, amiket összehasonlíthatunk logikai operátorok (kapcsolók) segítségével pl:
  - $x=$a==$b;
    //$a egyenlő-e $b-vel? Ha igen, akkor $x=true
    $y=$a>=$b;
    //$a nagyobb egyenlő-e mint $b? Ha igen, akkor $y=true
  - $x and $y
    $x && $y
    true if both $x and $y are true
  - $x or $y
    $x || $y
    true if either $x or $y is true
  - $x xor $y true if either $x or $y is true, but not both
  - !$x true ha $x is not true

PHP feltételes elágazások

- A programozási-/szkriptnyelvek igazi ereje a feltételes elágazásokban rejlik
- Szintaxis:
- if (feltétel) {
  ha a feltétel igaz volt, akkor ez a kódrész a kapcsos zárójeleken belül lefut, különben kihagyjuk és az else részbe lépünk
  } else {
  ez csak akkor fut le, ha a feltétel nem volt igaz
  }
- A feltétel helyén logikai összehasonlító operátorokat használunk, pl:
- $t = date("H"); //A mostani időt órában eltároljuk $t-ben
  if ($t < "20") {
  echo "Jo napot!";
  } else {
  echo "Jo estet!";
  }

PHP switch elágazások

- A switch elágazás segítségével rövidebb forráskóddal könnyebben számba vehetjük a lehetőségeket
- A switchben {} között a lehetőségeket felsoroljuk case lehetőség:-el, majd utána az ahhoz tartozó programkódot írjuk és ezt végül lezárjuk a break; utasítással, azért hogy megszakítsa ebben a pontban a sorfolytonos programfutást, azért hogy más lehetőséghez tartozó kódrész ne fusson le, ekkor a switch záró kapcsos zárójele } után ugrik a programfutás

PHP while ciklusok

- Ciklusok segítségével egy programrészt addig futtathatunk amíg a feltételünk igaz
- While szintaxis (elöltesztelő ciklus):
  while(feltétel) {
  amíg while feltétele igaz, addig ez a kódrész újra lefut
  }
- Do..while szintaxis (hátultesztelő ciklus):
  ez egyszer mindenképpen lefut, aztán ellenőrzi a feltételt:
  do {
  code to be executed;
  } while (condition is true);

PHP for ciklusok

- For ciklusok segítségével egy definíció alá vehetjük az iterátort (ciklus számláló) és a feltételt
- Így legrövidebb formában megadhatjuk, hogy hányszor fusson le egy kódrész
- Szintaxisa:

```
for (számláló; feltétel; számláló növelés) {
    kódrész;
}
```

- Pl:

```
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
```

- Foreach ciklus segítségével tömbök elemein mehetünk végig egyszerűen:
-

```
$colors = array("red", "green", "blue", "yellow");
foreach ($colors as $value) {
    echo "$value <br>";
}
```

PHP Superglobals

- Speciális beépített PHP globális tömbök segítségével lekérdezhetünk szerverinformációkat, .php szkriptnek küldött adatokat kaphatunk meg az adatküldés típusától (lásd űrlapkezelés $_POST vagy $_GET tömbök)
- PHP $_SERVER tömbből szerverinformációkat kérhetünk le
- $_POST tömbben html űrlapnál method=„post” adatküldési típus esetén ebben a tömbben lesznek a küldött adatok adatnév=>adatérték asszociatív tömb
- $_GET tömbben html űrlapnál method=„get” adatküldési típus esetén ebben a tömbben lesznek a küldött adatok adatnév=>adatérték asszociatív tömb
  (és az adatnevek, értékek láthatóak lesznek az URL címben is, így itt könnyen átírhatóak: URL?adatnev=adatertek)
- http://www.w3schools.com/php/php_superglobals.asp