Kedves Wawona,

Amit tudtam, vagy úgy gondoltam, hogy tudtam beírtam zölddel, amit nem tudtam pirossal.

Kedves jelentkező,

Az alábbi kérdések közül pár darab konkrét megoldást vár, ahol a válasz vagy jó vagy sem de van közöttük több, amiknél nincs jó vagy rossz válasz. Sokkal inkább arra vagyunk kíváncsiak, hogy te mit gondolsz. Ezekre a kérdésekre elege csak röviden, egy-két mondatban válaszolni. Nem Wikipedia definíciókat varunk sokkal inkább a TE gondolataidat. Ha esetleg véleményezed is a kérdéses dolgokat annak külön örülünk (például a második kérdésnél, hogy te használsz-e private tagfüggvényeket vagy ellenzed a használatukat).

1. Ha a párizsi Eiffel torony alatt állsz, Dél-Kelet fele fordulsz majd elindulsz egyenesen akkor hova jutsz (helyismeret nem szükséges a megoldáshoz)?

Első megközelítésben az gondoltam, hogy a Földközi-tengerbe.

A második gondolatom az volt, hogy a pont a torony lábainak mértani közepén állok, akkor előbb utóbb az egyik lábához érkezem.

Aztán meg kellett néznem google mapsen, és láttam, hogy 45 fokkal el van forgatva Észak-Dél és Kelet-Nyugat vektorok által meghatározott sík normál vektora körül, tehát valószínűleg az első válasz volt a jó.

Viszont tovább nézegetve a térképet, úgy gondolom, hogy Olaszországon át végig gyalogolva kb. a csizma sarka és a talpa közötti öbölben a Jón-tengerben kötök ki.

EU-s állampolgár lévén, csak a tenger lesz az akadály, mivel rosszul úszom.

Persze feltéve, hogy minden utamba kerülő akadályt le tudok győzni.

Ezután azon kezdtem gondolkodni, hogy ez a feladat eléggé pongyola megfogalmazású, mert nem definiálja, hogy:

* Járművel vagy gyalog állok a torony alatt.
* Ha járművel, akkor egy éltű, kétéltű vagy esetleg forgószárnyas repülőgéppel (mondjuk azt nem tudom vezetni)?
* Valóban csak Dél-Kelet felé tarthatok egyenes vonalban, vagy a kisebb kitérőkkel kikerülhettem az utamba kerülő objektumokat?
* Ez valójában elmélet kérdés csupán, és bármin áthatolhatok gondolatban?

1. Ebben feladatban egy “tisztító” algoritmust kell megvalósítani. Ki kell szűrni koordináták egy halmazából (szelességi, hosszúsági, időbélyeg) azokat az elemeket, amik feltehetőleg nem illenek a halmazba. Képzeld azt, hogy GPS segítségével rögzítünk egy utazást, de mivel a GPS néha hibázik/pontatlan ezért a listába kerülhetnek teljesen oda nem illő adatok. Ezeket kell kiszűrni.  
   Mellékelve találsz egy points.php fájl-t. Megtalálod benne a **Coordinate** osztályt (és annak interface-t is) aminek példányai egy-egy koordináta pontot képviselnek (szelesség, magasság, időbélyeg). Ezen kívül találsz benne egy **Journey**\_Abstract nevű absztrakt osztályt is. A feladat, hogy implementáld a **Journey** osztályt, ami kiterjeszti a **Journey\_Abstract**-ot.  
     
   Csak a **Journey** osztály megírása a feladat, NEM kell működőkepés, futtatható PHP alkalmazást készíteni. A mellékelt point.csv fájlban találsz pár koordinátát. Ezek egy valós utazást reprezentálnak néhány hibával. A feladat kiírása szerint nem kell sehol behívni, vagy beolvasni ezt a fájlt, de a saját munkád ellenőrzésére hasznos lehet.  
     
   A munkát teszt vezérelt módon (TDD) javasolt elvégezni. Ha az egységtesztek és a TDD világa messze áll tőled akkor nem kötelező így eljárni. A teszt(ek) kerülhetnek külön fájlba is, rád van bízva, csak ne felejtsd el csatolni.

Nem volt megadva, hogy milyen módszerrel kell tisztítani az GPS adatokat. A Googleban ezt találtam: <https://cran.r-project.org/web/packages/CoordinateCleaner/CoordinateCleaner.pdf>, de nem kívántam elmélyedni benne, ezért inkább nekiáltam saját megoldást barkácsolni, úgyis a PHP tudás számít…

Melléklet Excel táblázatban egy X-Y diagramon ábrázoltam a kapott koordinátákat, így ebből meghatároztam a legnagyobb elmozdulást, aszerint szűrtem végül.

1. Mi a különbség a **public**, **protected**, **private** tagfüggvény láthatóságok között a PHP nyelvben. Milyen előnnyel bír a privát tagfüggvény használata. Mondj egy példát.

Public – bárhonnét elérhető,

Private – csak a saját osztályából,

Protected – a saját és örökölt osztályokból érhető el.

1. Adott a következő PHP script:  
     
   <?php  
   $currencies = array(  
    ‘GBP’ => 360,  
    ‘HUF’ => 1  
   );  
     
   echo $currencies[‘GBP’];  
   echo $currencies[‘EUR’];  
   ?>  
     
   Mi fog, mi jelenhet meg a kimeneten? Valtoztatnal-e valamit a scripten és ha igen, mit?

Az első echo a htmlbe 360-at ír, a második 1-et.

Ha az a cél, hogy a teljes tartalmát echozzuk, akkor egy foreach ciklusba tenném az echot.

1. Mire jó a **@** operator PHP nyelvben? Hol használnád?

Függvények annotálására, pl. a Symphonyban találkoztam már vele. Ezen kívül a warningokat lehet vele elnyomni.

1. Van-e lehetőség PHP nyelvben **FATAL ERROR** eseten ‘hibakezelő funkció’ futtatására (pl email küldés, log bejegyzés, adatbázis tranzakció visszafordítás)?

Try-Catch blokkal, és Thorwn Exepctionnel, ha tudjuk a hiba forrását.

1. Adott egy alkalmazás, amiről annyit tudunk, hogy **php/html/js/mysql** technológiákat használ (tehát a jól bevált standard webes stack). Sajnos több panaszt kaptunk, hogy rendkívül lassan ‘tolt be’ az oldal. A feladatot te kapod, hogy nézz utána mi lehet a hiba és optimalizáld a kódot. Mit teszel? Miket nézel meg? Mikre figyelsz oda és milyen esetleges eszközöket használsz fel?

A böngésző developer tooljában megnézem a betöltődési időket, és a betöltődés időrendjét.

A HTML-ben az elemek sorrendjét a fontosság szerint sorba rakom, ha nem lehetséges ezzel optimalizálni, akkor a lassan töltődő elemek, videók, képek kapnak egy classt, amivel a JavaScript segítségével az oldal betöltődés után AJAX-al, Fetchel vagy Pormise-val letöltögetem a tartalmakat. Képek esetén lehet Thumbnailt alkalmazni, és azokat a betöltődés után a JavaScripttel kicserélgetni. Vagy ha nagyon lusta vagyok akkor LayzLoader vagy valami hasonló JS libraryt használok.

A headben hívott CSS, Fontok, és külső forrásból származó JavaScripteket defer/async tagekkel látom el.

1. Linux rendszer eseten mihez fordulnál, ha egy scriptet (legyen az akar php vagy barmi más) bizonyos időközönként automatikusan le kell futtatni.

Nincs még ilyen tapasztalatom, de Node.JS, képes setItervallal erre. PHP-ban nem tudom van-e mód rá?

1. Linux konzolban hogyan íratnád ki egy szöveges fájl tartalmat a standard kimenetre úgy, hogy ha a fájl tartalma változik akkor a kimeneten is egyből megjelenjen az. Tehát mintha egy log fájlt monitorodnál fejlesztés közben.

Még nem tudom.

1. Írj Linux parancsot, ami az aktuális könyvtárban fellelhető fájl-ok közül egy logfiles.txt-be írja azok fájlneveit, amik nevei tartalmazzak a ‘log’ kifejezést (bármilyen formában).

Még nem tudom.

1. Mi a különbség a **GET/POST/PUT/DELETE** keresek között mind **HTTP** mind **PHP** szempontból. Mondj egy-egy példát mikor használnád őket.

A GET a szerverről szerez információt, és az URL-ben utazik az adat,

A POST a szerverrel közöl információt, és az üzenet bodyjában található az információ,

A PUT a szerveren módosít információt, még nem használtam… ☹

A DELETE a szerveren töröl információt, még nem használtam PHP-ben

1. Mi a különbség a **401** és **403** **HTTP** válaszkód között. Mondj mindkettőre egy szituációt amikor használnád.

A 403-at a HTAccess adta eddig nekem, az olyan mappákra, amikre nem engedélyeztem elérést.

A 401-et még nem használtam. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/401> szerint felhasználó hitelesítési probléma esetén kell használni.

1. Miben különbözik a **https** a **http**-tol? Mikor javaslod a használatát a sima **http** helyett?

A https a titkosított és tanúsítvánnyal ellátott szerver kapcsolat. Minden esetben https-t használnék, mivel a kereső optimalizáláshoz elengedhetetlen, a robot hátra sorolja a csak http kapcsolattal rendelkező domaineket.

A http kapcsolatot ipari, vagy magánhálózaton, használom.

1. Hogyan védekeznél a cross-site request forgery (CSRF) támadástípus ellen. Mondj egy példát.

Semmilyen ismeretem nincs róla. Elolvasva a Wikipédia cikket sajnos még ötletem sincs, hogyan lehet védekezni ellene…

1. Milyen esetekben részesítenéd előnyben a **NoSQL** adatbázisokat az **SQL**-ekkel szemben?

Eddig csak MySQL-t használtam. Múlt héten megismerkedtem a MondoDB-vel ami NoSQL, de még nem jöttem rá, mik a hátrányai. Egyelőre nagyon tetszik, hogy bonyolult adatstruktúrákat egyszerűen lehet kezelni vele, sokkal egyszerűbben, mint az SQL adatbázissal.

1. Adott a két következő MySQL tábla:

|  |  |
| --- | --- |
| **Users** | **Logins** |
| user\_id (int – primary key) | login\_id (int – primary key) |
| name (varchar 32) | user\_id (int – foreign key) |
| register\_date (timestamp) | login\_date (timestamp) |

Írj egy SQL lekérést, ami kilistázza azon felhasználók neveit akik 7 napnál korábban regisztráltak és több mint 5 alkalommal leptek be az oldalra. A Users table tartalmazza a felhasználókat a Logins pedig a belépéseket.

SELECT name

FROM Users

WHERE (Users.register\_date < NOW() + 7) OR (COUNT(SELECT login\_id FROM Logins WHERE Logins.user\_id = name .user\_id)) > 5;