

Webprogramozás szakkör

Előadás 4 (2012.03.26)

Bevezető

Mi is az a programozási nyelv, mit láttunk eddig (HTML+CSS)?

Az eddig tanult két nyelven is mondhatni programoztunk, de ez nem a klasszikus értelemben vett programozás volt.

Programozási nyelv definíció: a számítástechnikában használt olyan, az ember által olvasható és értelmezhető utasítások sorozata, amivel közvetlenül, vagy közvetve (például: gépi vagy köztes kódra fordítás után) közölhetjük a számítógéppel egy adott feladat elvégzésének módját.

Egyszerűbben fogalmazva egy programozási nyelvvel valamilyen módon, megadott szabályok szerint leírhatjuk, hogy egy feladatot miként szeretnénk elvégezni, milyen lépések szükségesek ahhoz. A közvetettség pedig arra utal, hogy vagy olyan a nyelv, hogy megírjuk, és máris láthatjuk az eredményét (például a HTML vagy a CSS ilyen, hiszen a böngésző értelmezte azt a kódot, amit mi írtunk). Vagy pedig először egy program segítségével le kell fordítanunk olyan kódra, amit a gép tud értelmezni (gépi kód a processzor által értelmezhető, a köztes kód az adott értelmező, pl. Java által), és ezt tudjuk majd futtatni. Idén csak az előbbi kategóriába tartozó nyelvekkel foglalkozunk.

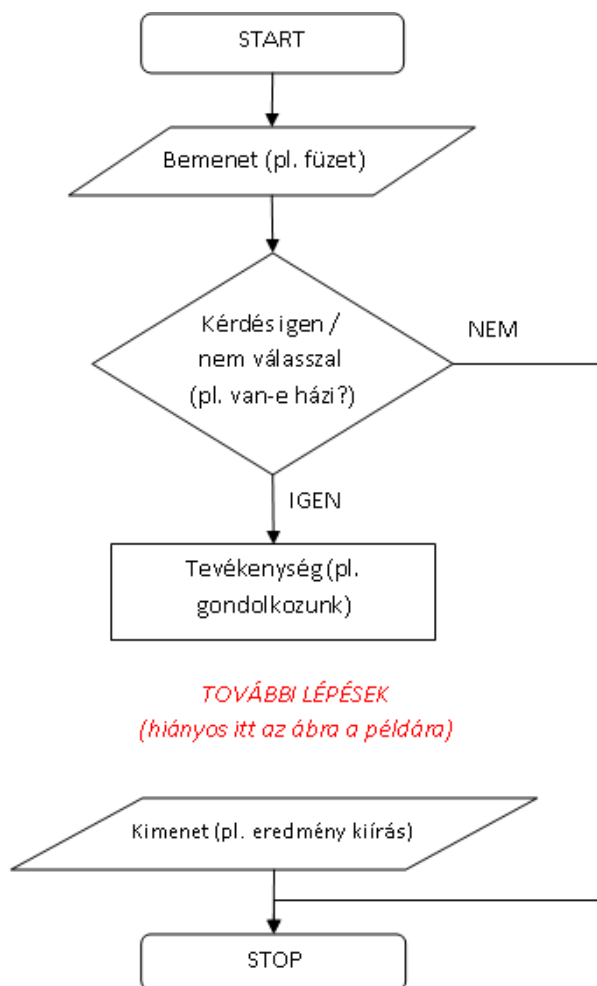
A HTML illetve a CSS úgynevezett leírónyelvek voltak. Ha valamiből 10 darab kellett, akkor megírtuk 10x. Hogyan lehetne másképpen? Mondhatnánk azt is, hogy ismétljük meg 10x. Ehhez más szemléletű nyelvek szükségesek, ilyen például a webprogramozásban gyakran használt JavaScript nyelv, mely jól kiegészíti majd az eddig tanult HTML-CSS ismereteinket.

Mi is az a programozás?

- Adott egy feladat.
- **Kérdés:** hogyan tudjuk leírni az adott feladatot?
- **Elvárás:** adott bemenetre adott kimenetet adjon
- Pl: Szervezzük mozizást
 - START
 - Nézzük meg a moziműsort, **UTÁNA**
 - **HA** találunk jó filmet, **AKKOR** döntsük el, mikor és hol néznénk meg
 - **HA NEM, és még nem unjuk, AKKOR** nézzük meg újra a moziműsort, **AMÍG** nem találunk megfelelőt
 - **EGYÉBKÉNT** hagyjuk az egész mozizást

- **HA** a kiválasztott időpontban az adott helyen van elég szabad hely, **AKKOR** foglaljunk
 - **HA NEM**, akkor keressünk új időpontot és helyet
 - A foglalás **UTÁN** a rendelést igazoljuk vissza, **UTÁNA**
 - Menjünk el a moziba, **UTÁNA**
 - Vegyük át a jegyet és nézzük meg a filmet
 - STOP
- **Sokféle megoldás lehetséges!**

Ezt ábrázolhatjuk egy folyamatábrán (az alábbi példa a házi feladat írásának kezdetleges folyamatát szemlélteti)



A folyamatábra elemei:

- **Dőlt téglalap** – bemenet (mi kell a feladat megoldásához) / kimenet (milyen eredménye van, pl: sikeres/nem sikeres vagy igen/nem...stb.)
- **Téglalap** – tevékenység
- **Rombusz** – eldöntendő kérdés, igen/nem válasszal, kétfelé megy ki nyíl
- **Lekerekített téglalap vagy kör vagy ellipszis** – START illetve STOP

Hogyan programozzuk le?

- Lépésről lépésre való megvalósítás
 - A folyamatot modellezzük, felkészülve a **hibákra** is
- Nincs megadott recept
 - Ismerni kell hozzá egy **nyelvet** + ötlet + gyakorlás
 - Matematika az alapja
 - Algoritmusok

Készítsünk el egy egyszerű algoritmust, arra, hogy 20 szám közül hogyan találjuk meg a legnagyobbat. Ehhez írjunk **pszeudokódot** (az algoritmus programszerű leírása érthető nyelven, mely lehet emberi, matematikai, DE nem maga a programozott leírás).

Írjunk fel mindent ami szükséges:

Feladat: 20 szám közül a legnagyobb megtalálása

Bemenet: 20 szám

Kimenet: a legnagyobb szám

Tegyük fel, hogy a számok tetszőleges sorrendben megvannak.

Lépések a megoldáshoz:

- Jegyezzük meg az első számot, **UTÁNA**
- Menjünk végig az összes számon és **ismételjük** a következőket:
 - **Ha** az adott szám nagyobb, mint az utoljára megjegyzett, **akkor** az eddig megjegyzett számot elfelejthetjük, és jegyezzük meg ezt a számot
 - **Utána** továbblépünk a következő számra
 - **Ha** nincs több szám (elértük a huszadikat), **akkor** végeztünk és a legutoljára megjegyzett szám lesz a legnagyobb

A programozás menete

- **Specifikáció:** a feladat megfogalmazása – MIT SZERETNÉNK CSINÁLNI?
- **Algoritmizálás:** tervezés – HOGYAN CSINÁLJUK MEG?
 - Módszer a feladat megoldására, véges sok lépésből
- **Kódolás:** megvalósítjuk egy programozási nyelven – MEGCSINÁLJUK!
- **Tesztelés:** megvizsgáljuk, hogy helyes-e a működés – JÓL CSINÁLTUK MEG?

Szerkesztő:

- Minden bonyolultabb (felsőbb szintű) nyelvhez saját szerkesztő társul
- Az egyszerűbbekhez elég egy jegyzetömb is (mindenhez elég, de sokszor kényelmetlen)

- Kódkiegészítés
- Szintaxis kiemelése
- Mi a Notepad++ programot fogjuk használni
 - Egyszerű, ingyenes

Az első nyelv megtanulása a legnehezebb, utána más nyelveken többnyire ugyanazok az alapelvek használhatóak, csak más szabályok vannak arra, hogy mit hogyan fogalmazhatunk meg.

Programozás alapok

Programkód szerkezete

Utasítás: a programozás alapegysége, tipikusan egy művelet elvégzését jelenti

Kód: adott programnyelven megírt utasítások sorozata

Az utasítások végrehajtási sorrendje lehet:

- Egymás után következő utasítások (szekvenciális)
- Ismétlés
- Elágazás (ha-akkor-egyébként) – csak az adott feltételnek megfelelő utasítások hajtódnak végre

JavaScript

A JavaScript nyelv szoros kapcsolatban van a HTML és CSS nyelvvel, azok lehetőségeit kiegészíthetjük vele. Böngészőben lehet futtatni, így könnyen hordozható – hasonlóan a HTML és CSS nyelvekhez.

Kiíratás

- `document.write("szöveg");` // szöveget idézőjelek közé tesszük
- `document.write('szöveg');` // aposztrófokkal is jó
- `document.write(2);` // szám (nincs idézőjel/aposztróf)

Ez egy utasítás, melynek a lezárása egy pontosvessző.

Feladat 0: Hello World!

Szükséges ismeretek: kiíratás

Megoldás:

Készítsünk Notepad++ programban egy új file-t (File/New), majd egyből üresen mentjük is el (File/Save as). A név megadásánál válasszuk ki, hogy ez egy HTML fájl legyen (HyperText Markup Language file - *.html, *.htm...). A <head> részbe írjuk a következőket:

```
<script type="text/javascript">
  document.writeln("Hello World!");
</script>
```

A `<script>` közti rész a JavaScript kód, ezt mindig meg kell adnunk (ha .html fájlba dolgozunk). A `<script>` csak egy HTML tag, ami azt jelöli, hogy utána például JavaScript kód következik.

Feladat 1 – írjuk ki 1-től 3-ig a számokat

Szükséges ismeretek: kiíratás

```
document.writeln(1);
document.writeln(2);
document.writeln(3);
```

Az utasítások egymás utániságát **szekvenciának** nevezzük. Itt egy utasítás egy kiírást végző kód. Az utasítások végén pontosvesszőt írunk, majd új sorban folytatjuk a kódolást az áttekinthetőség érdekében.

Kommentek (tetszőleges megjegyzés beírása a kódba)

- Egysoros: `//` ez itt a komment
- Többsoros: `/*` ez itt a komment, a lezárását jelző csillag-per jelekig tart `*/`

```
<script type="text/javascript">
  // egysoros komment
  document.writeln("Hello World!");
  /*
     Többsoros komment
     akár ide is jöhet szöveg
  */
</script>
```

Miért jó, ha kommenteket írunk? Hosszú kódoknál könnyebben találhatjuk meg, hogy mit hol és hogyan csináltunk meg. Ezt csak a saját magunk (a programozók) segítségére írjuk, a kommentben megadott szöveget a felhasználók a böngészőben nem látják. Akkor is segíthet, hogy ha később szeretnénk módosítani valamit a kódon és már nem emlékszünk, hogy mi hogyan volt.

Kód formázása

- Minden új beágyazott egységet egy tabulátorny (TAB gomb) hellyel beljebb kezdünk
- Beágyazott egység tipikusan kapcsolósárjel után következik

Változók

- Matematika
 - $x + 4 = 7 \rightarrow x = 3$
 - Itt az „x” az egy változó
- A változónak
 - neve van
 - értéket adhatunk neki

- Inicializálás (kezdeti értékadás)
 - `var x = 5;`
 - `var y = "Szöveg";` VAGY `var y = 'Szöveg';`
- **var** jelzi, hogy változót adunk meg
 - pontos jelentéséről későbbi alkalommal lesz szó
 - más nyelvekben helyette áll a típus, pl: `int x = 5;` (int jelenti, hogy egész szám típusú a változó)

Feladat 2 – hozzunk létre egy változót, írjuk ki az értékét

Szükséges ismeretek: kiíratás, változók

Inicializáljunk (adjunk neki értéket) néhány változót, és írjuk ki a következőket (kifejezéseket):

```
var elso = 5;
document.write(a);
document.write("Az elso nevű változó értéke: " + a);
```

A plusz jel segítségével tudunk két különböző dolgot összefűzni, egymás után tenni. Itt egy szöveget és a változó értékét írtuk ki az utolsó sorban.

Otthoni feladatok

1. Készíts **folyamatábrát** ahhoz, hogy hogyan tudsz eljutni az iskoládtól egy adott helyig. Legyél kreatív, használj benne váratlan eseményeket (pl. nem jár egy busz) vagy olyan helyre menj, amit nem ismersz, így meg kell tudnod valakitől, hogy jó úton jársz-e?
2. Írj **pseudokódot** (az algoritmus programszerű leírása érthető nyelven, mely lehet emberi, matematikai, DE nem maga a programozott leírás) két szám összeadására.

Mindkettőre példa látható korábban a leírásban.