



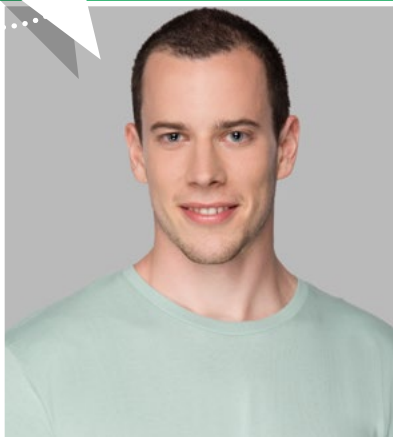
GREEN FOX ACADEMY

Szakiránytű

1 bootcamp - 9 szakirány

Mi a legfelkapottabb technológiákat oktatjuk.

Válaszd ki, melyik érdekel a legjobban!



index.html - matches x index.html

```
1 <li>= t "controllers.backend" </li>
```

```
2 </ul>
```

```
3 <div class="navbar navbar-default">
```

```
4 <div class="navbar-inner">
```

```
5 <ul class="nav navbar-nav">
```

```
6 <li class="current"><a href="backend_matches.html">
```

```
7 <span class="glyphicon glyphicon-home">
```

```
8 <span>= t "actions.index"</span>
```

```
9 <li class="current"><a href="polymorphic.html">
```

```
10 <span class="glyphicon glyphicon-home">
```



Front-end



Miről szól?

Kifejezetten weboldalokról, azoknak is a megjelenítés részéről: itt nem az a kérdés, hogy mi az adat, hanem hogy hogy néz ki és hogyan működik.



Mi a kihívás benne?

A fejlesztésnek millió fajtájú, márkájú, kijelzőjű eszközön és különböző böngészőkben is jól kell kinéznie. Továbbá a front-endhez tartozik bármilyen bonyolultabb akció leprogramozása is: zoom, drag and drop... pl. mikor egy ruhavásárló site-on nagyobb lesz a kép, ha fölé húzod az egered.



Mikor válaszod?

Ha fontos számodra, hogy látványos, színes-szagos legyen a munkád.



Példa

A kedvenc közösségi oldalon a back-endtől jön az adat, hogy milyen képeid vannak a galériában, de az, hogy ezek hogyan jelennek meg és tudod nyilakkal léptetni őket, már front-end.

Tudtad?.....



- A tévhittel ellentétben nem feltétlen szükséges hozzá szépérezék, mert általában van külön dizájner vagy UX-es erre a célra. (Persze örülnek, ha valakinek van pluszban szépérezéke, de ezek inkább a céges kultúrától és igényektől függenek.)
- Sokkal egyszerűbbnek tűnik, mint a back-end, de nem az!



Back-end



Miről szól?

Ez a szerveroldali fejlesztés, tehát minden, ami a szerveren történik: adatbáziskezelés, interneten történő kommunikáció fogadása, stb. Ez a weblapfejlesztésnek az a része, ami az adatokat kezeli, azaz nem a képernyőn látszik, hanem a háttérben történik. Back-end szinte minden cégnél, webalkalmazásnál, mobilalkalmazásnál van a világban. Hol nincs? Pl. mobilos számológépben vagy egy olyan játékban, ahol nem játszol mással.



Mi a kihívás benne?

Sok felhasználó kiszolgálása, biztonságos adatkezelés és -tárolás.



Mikor válaszd?

Ha nem feltétlen a színes-szagos munka, hanem az absztrakt feladatok tetszenek.



Példa

Az, hogy egy közösségi oldalra tudsz regisztrálni, be tudsz jelentkezni, le tudod kérdezni az ismerőseid listáját, meg tudod nézni a képeiket, tudsz velük chatelni, a back-end része a történetnek. Tehát bármi olyan interakció, ami megjelenik másnak, vagy más csinált, és te akarsz megnézni, vagy másik számítógépen is eléred.

Tudtad?.....

■ Erre a területre erősebb logikus, algoritmikus gondolkodás szükséges (főleg a későbbiekben, de ez nem feltétlen jelenti azt, hogy a back-endesek okosabbak is).





Full-Stack



Miről szól?

A full-stack fejlesztő a back-endes és front-endes keveréke, aki mélyen vagy csak az egyik, vagy egyik területhez sem ért, de mindkét területen elboldogul a tudásával, tudja, hogy mire van szüksége az egyik és a másik oldalnak. Projektszervezetekben általában a techlead szerepkör ilyen, akinek tudnia kell kommunikálni mind a front-endes, mind a back-endes munkatársakkal.



Mi a kihívás benne?

Valamennyire mindkét területre rá kell látni, kicsit “minden lében kanálnak” kell lenni, aki bevethető mindkét oldalon.



Mikor válaszsd?

Ha nem szeretnél választani front-end és back-end között.:)



Példa

Inkább kis cégeknél jellemző, ahol még kevesen vannak, vagy ha freelancerként kis weboldalak elkészítésével szeretnél foglalkozni. Nagyobb projekteken, ahogy ezt már írtuk, többnyire a techlead pozíció kívánja meg a full-stack tudást.



Tudtad?

- Ezt a területet és kifejezést mostanság nagyon felkapták, szinte buzzword lett.
- Hiába vagy full-stackes, az egyiket mindig jobban fogod szeretni.
- Igazából ma már elvárás mind a back- és front-endestől is, hogy ne legyen szakbarbár.



Desktop



Miről szól?

Nem weboldalak, hanem olyan programok készítéséről, amik számítógépen futnak. Inkább az a lényege, hogy a számítógépen tudsz valamit megoldani/csinálni vele.



Mi a kihívás benne?

A legtöbb helyzetben olyan technológiákat használnak desktop applikációk fejlesztéséhez, amik közelebb állnak a hardverhez. Így sokszor bonyolultabb dolgokkal kell foglalkozni, ami kezdő programozóknak félelmetes lehet. (Persze hamar bele lehet jönni.)



Mikor válaszd?

Ha nem ijedsz meg attól, hogy mélyebben hogyan működik a számítógép.



Példa

Minden, aminek ikonja van a számítógépeden: böngészők, videóvágók, hangszerkesztők, számítógépes játékok (ez ne tévesszen meg, sajnos nincs igény sok játékfejlesztőre)...



Tudtad?

■ Nemcsak hétköznapi felhasználásra fejlesztenek desktop applikációkat, hanem sokszor az ipari automatizálásban a vezérlési feladatokat is ilyen alkalmazások végzik.



Mobile



Miről szól?

Kifejezetten a mobil felületen való megjelenésről, tehát a mobil megfelelője a front-endnek. Nálunk jelenleg Androidra tudsz megtanulni fejleszteni.



Mi a kihívás benne?

Kicsi az átjárás a területen belül, egy Androidra írt app csak Android operációs rendszerű telefonon fut.



Mikor válaszd?

Ha olyan dolgokat fejlesztenél, amik kis képernyőn látványosak.:)



Példa

Minden applikáció, amit le tudsz tölteni a telefonodra és kis ikonként jelenik meg a képernyőn.



Tudtad?.....

■ Nem minden esetben a webapplikáció születik meg előbb, sőt van, ami csak mobilapp formájában létezik.



Embedded

(beágyazott rendszerek)



Miről szól?

Minden olyan program fejlesztéséről, ami nem számítógépen/mobilon/weben fut, hanem egyéb eszközön: autóban, hűtőben, liftben...



Mi a kihívás benne?

Meg kell ismerned annak az eszköznek a működését, alkatrészeit, amire fejlesztes. A programozói tudáson túl ehhez gyakran erősebb fizikai, matematikai ismeretekre is szükség van.



Mikor válaszod?

Ha életre kellenéd a kódsorod, azaz a fejlesztésed végeredményét nemcsak egy képernyőn, hanem egy eszközön is látni szeretnéd (mozgás, villogás, hangadás, stb. formájában).



Példa

Ide tartozik minden okoseszköz is, pl. az okosotthon okosmosógépe.

Tudtad?.....

- Ezt a fejlesztést hardver programozásnak is hívhatjuk.





DevOps



Miről szól?

DevOpsról webalkalmazások esetén beszélhetünk. Ők felelnek a szerverek működéséért. Mivel azonban ezeket az alkalmazásokat sok-sok szerver szolgálja ki, olyan programokat kell írniuk, amik ezeket automatizálják, ezáltal az adatok gyorsan futnak, mérhetőek, monitorozhatóak.



Mi a kihívás benne?

A DevOpsosnak ismernie kell az egész rendszert: hogy milyen hardveren futnak a kódok, melyiknek mi az előnye/hátránya, hogyan kapcsolódnak egymáshoz a különálló szoftvermodulok. Ezáltal ő nevezhető a "karmesternek", aki a leginkább képben van, aki hívja a fejlesztőt, ha valami nem működik.



Mikor válaszod?

Ha kicsit másképp szeretnél alkotni: nem úgy mint egy hegedűs, hanem mint egy karmester. Itt nem kevesebb a kreativitás, hanem komplexebb, még a back-endnél is absztraktabb. Nem árt egy kis geekség sem..)



Példa

Amikor a kedvenc közösségi oldalad zökkenőmentesen, gyorsan fut, elérhető minden funkciója.



Tudtad?

- A DevOpsos nemcsak karmester, hanem kapus is: általában ő észleli a hackertámadásokat.



>> Test Automation



Miről szól?

Mielőtt bármilyen kód élesedik, fontos meggyőződni annak helyességéről, hogy ne okozzon gondot a felhasználóknak. A tesztautomatizáló olyan programot ír, ami nagyon rövid idő alatt ellenőrzi, hogy az alkalmazás az elvárt módon működik-e, akár automatizáltan "végigkattintgat" egy weblapot, és még publikálás előtt detektálja benne a hibákat, sérülékenységeket.



Mi a kihívás benne?

A DevOpsos karmesteres hasonlatnak megfelelően itt sem elég programozni, tisztában kell lenni a pontos elvárt működéssel, tudni kell az ördög ügyvédjeként gondolkozva fogást találni a megírt programon.



Mikor válaszd?

Ha úgy érzed, hogy bármit el tudsz rontani, akkor a tesztautomatizálás neked való, hiszen az lesz a munkád lényege, hogy rávilágíts a hibákra. És ha bírod a rossz zsaru szerepet!



Példa

A tesztautomatizálás magas szintje az etikus hackerkedés (penetration testing), illetve a stressztesztelés, amikor azt próbálsz ki, hogy a program elbír-e egyszerre több százezer vagy millió felhasználót.

Tudtad?.....

- Sokan összekeverik a manuális teszttel, de ma már korántsem erről van szó.



ÚJ!

Data Science



Miről szól?

Napjainkban végtelen mennyiségű adat termelődik az online térben. Ennek a sok adatnak viszont csak akkor van értelme, ha értékes információt tudunk kinyerni belőlük, azokból pedig le tudjuk vonni a megfelelő következtetéseket. Az adattudós az, aki olyan programot ír, ami kibányássza ezeket az információkat, és azok alapján válaszol a felmerülő üzleti kérdésekre.



Mi a kihívás benne?

Nemcsak programozni kell tudni hozzá, hanem érteni kell azt a közeget, amiből az adatok érkeznek. Tudni kell értelmet adni a rengeteg, önmagában jelentés nélküli vagy nem sok jelentéssel bíró adatnak.



Mikor válaszod?

Ha valamilyen pénzügyi vagy tudományos területről jössz (társadalomtudomány, természettudomány), mert sokkal jobban tudod értelmezni az adatokat, ha ismered azt a közeget, amiből azok érkeznek, és nem csak a számokat követed. De például egy étteremlánc adatainak értelmezéséhez a vendéglátós tapasztalat nyújt előnyt.



Példa

Íde tartoznak a különféle kép-, hang- vagy szövegfelismerő szoftverek, például amikor az autóban a kamera felismeri a gyalogost, vagy amikor egy telefonodon futó alkalmazással felismerteted a körülötted szóló zenét. De data scientistek dolgoznak azon is, hogy bármilyen weboldalon járva olyan tartalmat vagy hirdetést kapj, ami téged a legvalószínűbben érdekel.

Tudtad?

■ Egyik alterülete az adatvizualizáció, amikor egy olyan programot írsz, ami megjeleníti az adatokat, pl. készít belőlük egy interaktív grafikont.



JUNIOR PROGRAMOZÓ KÉPZÉS

Következő kurzus: július 30.

Jelentkezési határidő: június 17.

www.greenfoxacademy.com