




Javascript

Domonkos Sándor Balázs Phd hallgató

Programozási alapismeretek felzárkóztató
kurzus

Jelen tananyag a Szegedi Tudományegyetemen
készült az Európai Unió támogatásával.

Projekt azonosító: EFOP-3.4.3-16-2016-00014

 <p>Oktató: Domonkos Sándor Balázs</p>	 <p>Programozási alapismeretek</p>	 <p>Becsült olvasási idő: 15 perc</p>
--	--	---

Javascript Első Lecke

Mi is ez? Miért éppen JavaScript?	3
HTML minta	3
CSS MINTA	4
Javascript Minta	4
Mi Köze a Javahoz?	5
References / Segédletek	6
Fejlesztői Eszközök (Integrated development environment)	6
Webstorm példa	7
Debug/problémakeresés megoldások	7
Google Chrome Fejlesztői mód	7
Példákon Keresztüli Tanulás	8
Gyakorlásra felületek böngészőből	8
Ellenőrző kérdések	10
Válaszok:	11

Mi is ez? Miért éppen JavaScript?

JS az egyik legfontosabb dolog a 3 dolog közül amit minden webfejlesztőnek tudnia kell legalább:

1. **HTML** megadja a weboldal tartalmi részét
2. **CSS** weboldalnak magának a kinézetét
3. **JavaScript** a programozása a weblap funkcióknak-viselkedéseknek

Nem csak webes felületeken használhatunk javascriptet, sok asztali és szerver oldali alkalmazás is használ JS-t, mint a Node.js a legismertebb, egyes adatbázisok is mint a mongoDB és couchDB

HTML minta

```
<!DOCTYPE html>
<html>
<head>
<title>Oldalunk címe</title>
</head>
<body>
<h1>Ez egy H1 osztályú fejléc</h1>
<p>Ez pedig itt egy bekezdés.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
</body>
</html>
```

Ez egy H1 osztályú fejléc

Ez pedig itt egy bekezdés.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

CSS MINTA

```
HTML
1 <h1>Html H1 header osztályú szöveg</h1>
2 <p>Ez pedig itt egy bekezdés a bekezdés színét css-el formáztuk.</p>
```

```
CSS (SCSS)
1
2 h1 {
3   font-size: 2rem;
4   font-weight: bolder;
5   margin-bottom: 1rem;
6   color: green;
7 }
8
9 p {
10  margin-bottom: 1rem;
11  font-size: 20px;
12  color: tomato;
13 }
```

Html H1 header osztályú szöveg

Ez pedig itt egy bekezdés a bekezdés színét css-el formáztuk.

Javascript Minta

Előző példát kiegészítve egy Elbújok gombbal, ami kattintásra a <p> osztályt átlátszóvá teszi **CSS felülírással** (erről később bővebben) :

```
HTML
1 <h1>Html H1 header osztályú szöveg</h1>
2 <p>Ez pedig itt egy bekezdés a bekezdés színét css-el formáztuk.</p>
3 <button>Elbújok</button>
```

```
JS
1 $('button').on('click', function() {
2     $('p').css('opacity', 0);
3 });
```

A button elemre van rákötve az **on click** funkció ami a kattintást érzékeli az elemen (beépített JS vezérlő) és erre reagálva a **p** osztály css-ben átállítja az átlátszóságot(opacity) 0-ra.

Html H1 header osztályú szöveg

Ez pedig itt egy bekezdés a bekezdés színét css-el formáztuk.

Elbújok

Kattintás **Előtt**

Html H1 header osztályú szöveg

Elbújok

Kattintás **Után**

Mi Köze a Javához?

Röviden: Csak névrokonok.

A Java igényel fordítót/compiliert és virtuális gépen vagy böngészőben is képes futni, de mindenképp szükséges a "fordítás", ipari felhasználásra van leginkább, van kliens szerver opció.

Filekiterjesztése a .java. Objektumai Class alapúak.

A javascript pedig egy "lightweight" script programozási nyelv webes felületekre és .js a filekiterjesztés. Objektumai Prototype alapúak ami sokkal egyszerűbb a Class alapúaknál. Kiens oldalon fut mindig.

References / Segédletek

Itt található útmutatókat és leírásokat az összes javascriptben előforduló kifejezéshez

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

Pl egy **for ciklus** leírása formálisan:

```
for ([initialization]; [condition]; [final-expression])  
    statement
```

Alatta kifejtve, hogy melyik kifejezés mit is jelent. Illetve egy egyszerű példával megtámogatva

```
for (let i = 0; i < 9; i++) {  
    console.log(i);  
    // more statements }
```

Csak a példa kedvéért itt:

let i= 0 az i változót inicializálja (azért nem var, mert a var az globális a let pedig csak blokkon belül határoz meg így optimálisabb a használata)

i<9 addig csinálja amíg i kisebb mint 9

++ minden lépés után i értékét 1-el növeli

Console.log(i) consolera kiírja i értékét (console kiíratásról később)

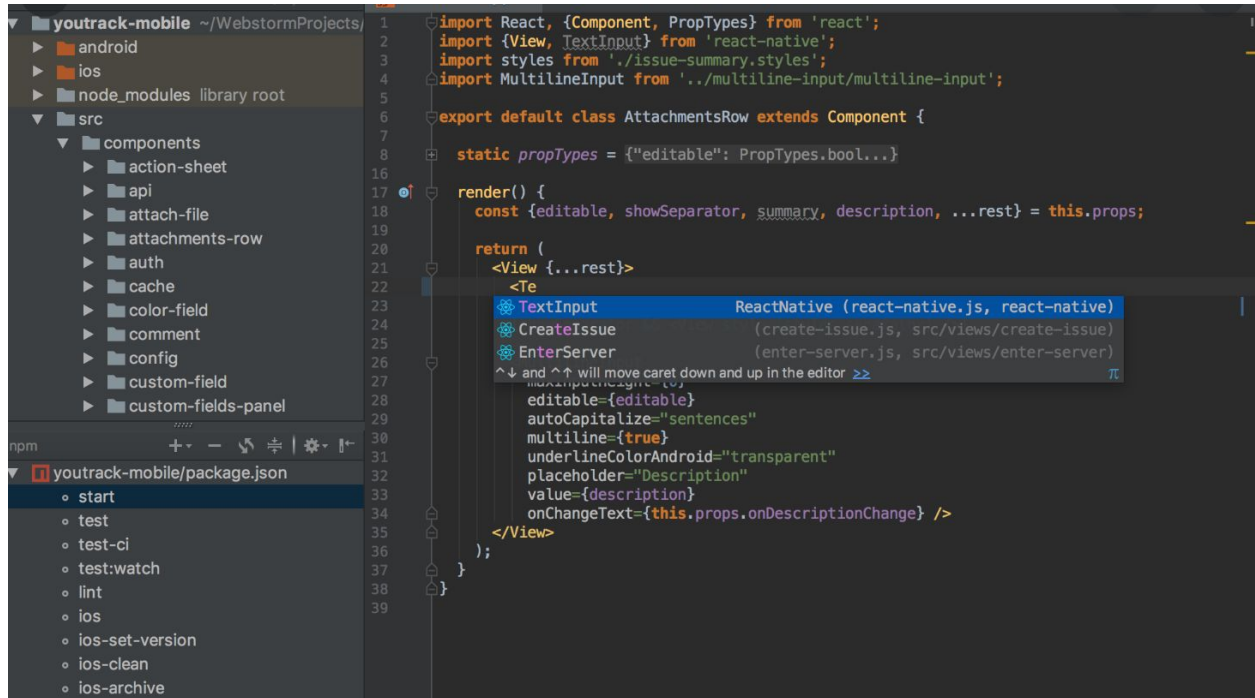
Fejlesztői Eszközök

(Integrated development environment)

- Webstorm
- Komodo Edit
- Sublime Text 3
- Notepad++
- Atom
- PhpStorm
- IntelliJ IDEA

- Egyszerű jegyzetomb is megfelelő példákhoz vagy kicsi projektekhez

Webstorm példa



```
1 import React, {Component, PropTypes} from 'react';
2 import {View, TextInput} from 'react-native';
3 import styles from './issue-summary.styles';
4 import MultilineInput from '../multiline-input/multiline-input';
5
6 export default class AttachmentsRow extends Component {
7
8   static propTypes = {"editable": PropTypes.bool...}
9
10  render() {
11    const {editable, showSeparator, summary, description, ...rest} = this.props;
12
13    return (
14      <View {...rest}>
15        <Te
16          <TextInput
17            ReactNative (react-native.js, react-native)
18            CreateIssue (create-issue.js, src/views/create-issue)
19            EnterServer (enter-server.js, src/views/enter-server)
20            and ^ and ^ will move caret down and up in the editor >>
21            editable={editable}
22            autoCapitalize="sentences"
23            multiline={true}
24            underlineColorAndroid="transparent"
25            placeholder="Description"
26            value={description}
27            onChangeText={this.props.onDescriptionChange} />
28        </View>
29      );
30    }
31  }
32 }
33
34
35
36
37
38
39
```

Komplex sok összetevős projektnél érdemes IDE-t használni legfőképp. Illetve a JS parancsokat segítik kiegészíteni gépelés közben és az esetleges elgépeléseket is tudja jelezni.

Debug/problémakeresés megoldások

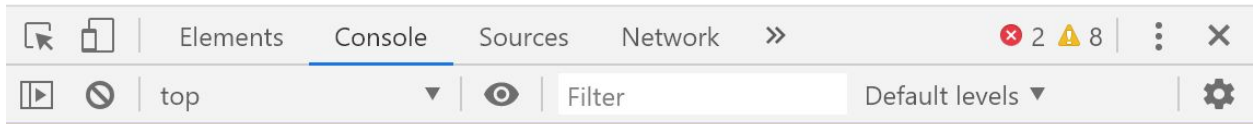
Kódban lévő hibák felkutatása a kód lépésről lépésre való nyomkövetésével, esetleg bizonyos változók és funkciók megjelölésével és ezek jobban megfigyelhetőségével támogatva..

A fentebb említett IDE-k külön kezelik a debug kiírásokat, breakpointokat és parancsokat etc, de pl a Chrome beépített eszközei is nagyszerű társaink lehetnek a bug-ok felderítésében és kijavításában.

Google Chrome Fejlesztői mód

Jobb gomb és vizsgálat vagy ctrl shift I és ott console tab rész ide tudunk mindenféle üzeneteket-változókat-állapotokat kiírni illetve az esetleges hiba üzenetek is itt fognak

megjelenni.



Példákon Keresztüli Tanulás

Egy adott példa sokszor többet érhet mint több órás magyarázás, ez főleg igaz a JavaScript esetében ahol úgymond **“szép”** kódot láthatunk magunk előtt és könnyen átlátható legtöbbször a szerkezete a kódnak. De legtöbb esetben az utolsó 4-5 leckéknél kommentálni is fogom a kódot a könnyebb értelmezhetőség kedvéért.

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var day;
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case 6:
    day = "Saturday";
    break;
  default:
    day = "unknown";
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script>

</body>
</html>
```

Today is Saturday

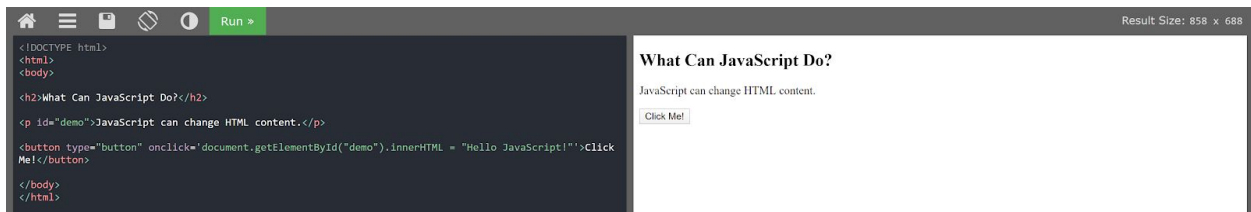
Esetek kezelése

Gyakorlásra felületek böngészőből

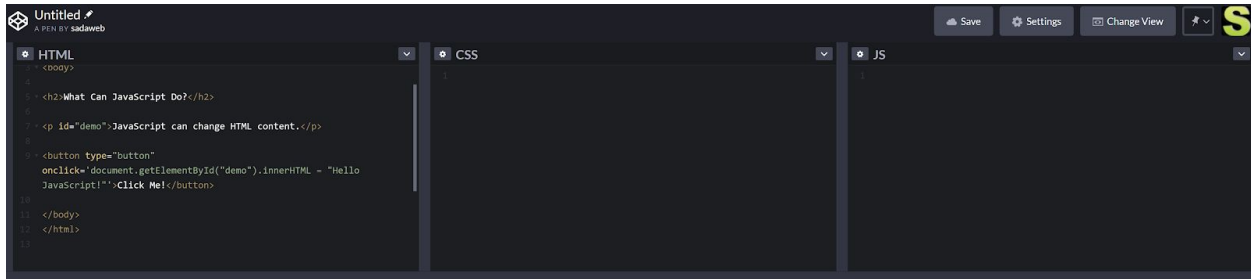
[Codepen.io](https://codepen.io) (van beépített console is)

w3schools.com/js/tryit.asp

W3 példa felülete így néz ki :



Ez jóval egyszerűbb mint a codepenes egyszerűbb 1 fileos js scripteket lehet jól kipróbálni itt. Codepen példa felülete:



What Can JavaScript Do?

JavaScript can change HTML content.

Click Me!

Sokkal komplexebb itt alpból már a JS részt külön kezeli és leválasztja a html tartalomtól, illetve a css részt is külön lehet választani a jobb átláthatóság kedvéért.

Ellenőrző kérdések

1. Sorolj fel három IDE-t ami megfelelő javascript fejlesztésre?
2. A JAVA nyelvből származik-e a JAVASCRIPT?
3. Böngészőn kívül tudom-e futtatni a JAVASCRIPTET?
4. A google chrome megfelelő eszköz lehet hibakeresésre/debugolásra?
5. Codepen felületen tudok futtatni saját javascript kódot tesztelésre?

Válaszok:

Sorolj fel három IDE-t ami megfelelő javascript fejlesztésre?

Webstorm

Komodo Edit

Sublime Text 3

Notepad++

Atom

PhpStorm

IntelliJ IDEA

A JAVA nyelvből származik-e a JAVASCRIPT?

Nem

Böngészőn kívül tudom-e futtatni a JAVASCRIPTET?

Nem, csak szimulátorral az IDE környezetekben.

A google chrome megfelelő eszköz lehet hibakeresésre/debugolásra?

Igen, van debug console beépítve.

Codepen felületen tudok futtatni saját javascript kódot tesztelésre?

Igen