**### 6. óra: Vezérlési szerkezetek I.**

**- `if`, `else if`, `else` szerkezetek**

**- `switch` szerkezet**

**- Gyakorlati feladatok: különböző feltételekhez igazodó kódok írása.**

**6. óra: Vezérlési szerkezetek I. (if, else if, else és switch)**

**Óravázlat – 45 perc**

**Bevezetés (5 perc)**

* *Tanár:* „Sziasztok! Ma egy rendkívül fontos témáról fogunk beszélni a JavaScript nyelvben: a vezérlési szerkezetekről. Ezek a szerkezetek lehetővé teszik, hogy programjaink különböző feltételek alapján hozzanak döntéseket. Az if, else if, else és switch szerkezeteket fogjuk megismerni.”
* *Tanár:* „Miért van erre szükség? Gondoljatok arra, amikor egy weboldalon valamilyen művelet csak akkor történik meg, ha egy feltétel teljesül, például ha be vagyunk jelentkezve, látunk bizonyos információkat, ha nem, akkor nem látunk semmit. Ez mind a vezérlési szerkezetek ereje.”

**if, else if és else szerkezetek bemutatása (15 perc)**

1. *Tanár:* „Kezdjük az if szerkezettel. Az if feltételvizsgálatot végez. Ha az if mögötti feltétel igaz, akkor a benne lévő kódblokk végrehajtódik.”
   * *Mutass egy példát a táblán vagy kivetítőn:*

javascript

Copy code

let szam = 10;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

}

* + *Tanár:* „Ebben a példában a szam változó értéke 10. Az if feltétele azt vizsgálja, hogy szam > 5, ami igaz, tehát a konzolon megjelenik az üzenet: 'A szám nagyobb, mint 5.'”

1. *Tanár:* „Nézzük meg, mi történik, ha a szam változó értékét 3-ra változtatjuk!”
   * *Írd fel / mutasd be a következő módosítást:*

javascript

Copy code

let szam = 3;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

}

* + *Tanár:* „Ebben az esetben az if feltétele hamis (3 > 5 nem igaz), így a kódblokk nem fut le, és a konzolon semmi sem jelenik meg. Ez az alapja a feltétel alapú végrehajtásnak.”

1. *Tanár:* „Az else és az else if szerkezetek lehetővé teszik, hogy különböző forgatókönyveket kezeljünk. Nézzük meg, hogyan használhatjuk őket!”
   * *Példa:*

javascript

Copy code

let szam = 3;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

} else if (szam === 5) {

console.log("A szám egyenlő 5-tel.");

} else {

console.log("A szám kisebb, mint 5.");

}

* + *Tanár:* „Ebben a példában először megnézzük, hogy a szam nagyobb-e, mint 5. Ha ez nem igaz, ellenőrizzük, hogy pontosan 5-e az értéke. Ha egyik feltétel sem igaz, akkor az else blokk fut le. Így biztosítjuk, hogy minden eset le legyen kezelve.”

**Részletes Magyarázat:**

* *Tanár:* „Fontos megértenünk, hogy az else blokkot kizárólag akkor hajtja végre a program, ha egyik korábbi if vagy else if feltétel sem teljesül. Az else if szerkezetet pedig akkor használjuk, ha egy új feltételt kell ellenőriznünk az első if után.”

**Példák az osztály részére:**

* *Tanár:* „Most nézzünk egy példát, ahol több feltételt is megvizsgálunk!”

javascript

Copy code

let kor = 18;

if (kor >= 18) {

console.log("Nagykorú vagy.");

} else if (kor >= 14) {

console.log("Kamasz vagy.");

} else {

console.log("Gyermek vagy.");

}

* *Tanár:* „Ebben az esetben különböző korcsoportokat kezelünk. Az első feltétel, kor >= 18, azt vizsgálja, hogy valaki nagykorú-e. Ha nem, akkor megnézzük, hogy kamasz-e (kor >= 14). Ha egyik sem igaz, akkor a else blokk fut le.”

**Gyakorlati Feladat (10 perc)**

* *Tanár:* „Most ti jöttök! Írjatok egy programot, amely ellenőrzi, hogy egy szám pozitív, negatív vagy nulla-e, és ennek megfelelően írja ki a konzolra az eredményt!”
  + *Támogatás az osztályban:* Járj körbe, figyeld a diákok munkáját, és segíts azoknak, akik elakadnak.

**switch Szerkezet Bemutatása (10 perc)**

1. *Tanár:* „A switch szerkezet akkor hasznos, ha egy változó több különböző értéke alapján kell döntést hoznunk. Nézzünk egy példát, ahol a napokat szám szerint kezeljük!” *Példa:*

javascript

Copy code

let nap = 3;

switch (nap) {

case 1:

console.log("Hétfő");

break;

case 2:

console.log("Kedd");

break;

case 3:

console.log("Szerda");

break;

default:

console.log("Ismeretlen nap");

}

1. *Tanár:* „A switch összehasonlítja a nap változót az összes case-ben megadott értékkel. Ha talál egy egyezést, a megfelelő blokk fut le. Mi történik a break utasítás nélkül? Igen, a következő blokk is lefutna. A default blokk az, ami akkor hajtódik végre, ha egyik case sem felel meg.”

**Részletes Magyarázat:**

* *Tanár:* „A break azért fontos, mert megakadályozza, hogy az összes többi case blokk is lefusson, miután megtalálta a megfelelő egyezést. Ha kihagyjuk a break-et, akkor az úgynevezett 'fall-through' viselkedés történik, ami általában nem kívánatos.”

**Gyakorlati Feladat (10 perc)**

* *Tanár:* „Készítsetek egy switch szerkezetet, amely az évszakokat adja vissza egy adott hónap száma alapján. Például 1–3 hónapra 'Tél', 4–6 hónapra 'Tavasz' stb. Használjatok case-eket és break-eket!”

**Óraösszegzés (5 perc)**

* *Tanár:* „Ma áttekintettük az if, else if, else és switch szerkezeteket. Ezekkel lehetővé válik, hogy programjaink logikusan döntsenek különböző feltételek alapján. Ezek a vezérlési szerkezetek a programozás egyik legfontosabb eszközei. Kérdések?”
* *Tanár:* „Ha bármi kérdés van, ne habozzatok kérdezni, és legközelebb is fogunk ezekre építeni!”

### 6. óra: Vezérlési szerkezetek I. (`if`, `else if`, `else` és `switch`)

---

#### \*\*Óravázlat – 45 perc\*\*

\*\*Bevezetés (5 perc)\*\*

- \*Tanár:\* „Sziasztok! Ma egy rendkívül fontos témáról fogunk beszélni a JavaScript nyelvben: a vezérlési szerkezetekről. Ezek a szerkezetek lehetővé teszik, hogy programjaink különböző feltételek alapján hozzanak döntéseket. Az `if`, `else if`, `else` és `switch` szerkezeteket fogjuk megismerni.”

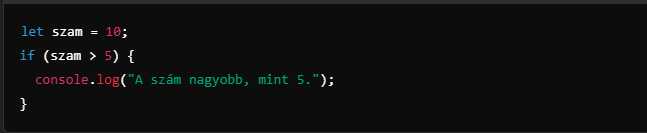
- \*Tanár:\* „Miért van erre szükség? Gondoljatok arra, amikor egy weboldalon valamilyen művelet csak akkor történik meg, ha egy feltétel teljesül, például ha be vagyunk jelentkezve, látunk bizonyos információkat, ha nem, akkor nem látunk semmit. Ez mind a vezérlési szerkezetek ereje.”

---

\*\*`if`, `else if` és `else` szerkezetek bemutatása (15 perc)\*\*

1. \*Tanár:\* „Kezdjük az `if` szerkezettel. Az `if` feltételvizsgálatot végez. Ha az `if` mögötti feltétel igaz, akkor a benne lévő kódblokk végrehajtódik.”

- \*Mutass egy példát a táblán vagy kivetítőn:\*



```javascript

let szam = 10;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

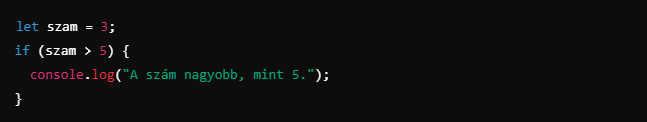
}

```

- \*Tanár:\* „Ebben a példában a `szam` változó értéke 10. Az `if` feltétele azt vizsgálja, hogy `szam > 5`, ami igaz, tehát a konzolon megjelenik az üzenet: 'A szám nagyobb, mint 5.'”

2. \*Tanár:\* „Nézzük meg, mi történik, ha a `szam` változó értékét 3-ra változtatjuk!”

- \*Írd fel / mutasd be a következő módosítást:\*



```javascript

let szam = 3;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

}

```

- \*Tanár:\* „Ebben az esetben az `if` feltétele hamis (`3 > 5` nem igaz), így a kódblokk nem fut le, és a konzolon semmi sem jelenik meg. Ez az alapja a feltétel alapú végrehajtásnak.”

3. \*Tanár:\* „Az `else` és az `else if` szerkezetek lehetővé teszik, hogy különböző forgatókönyveket kezeljünk. Nézzük meg, hogyan használhatjuk őket!”

- \*Példa:\*

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

```javascript

let szam = 3;

if (szam > 5) {

console.log("A szám nagyobb, mint 5.");

} else if (szam === 5) {

console.log("A szám egyenlő 5-tel.");

} else {

console.log("A szám kisebb, mint 5.");

}

```

- \*Tanár:\* „Ebben a példában először megnézzük, hogy a `szam` nagyobb-e, mint 5. Ha ez nem igaz, ellenőrizzük, hogy pontosan 5-e az értéke. Ha egyik feltétel sem igaz, akkor az `else` blokk fut le. Így biztosítjuk, hogy minden eset le legyen kezelve.”

\*\*Részletes Magyarázat:\*\*

- \*Tanár:\* „Fontos megértenünk, hogy az `else` blokkot kizárólag akkor hajtja végre a program, ha egyik korábbi `if` vagy `else if` feltétel sem teljesül. Az `else if` szerkezetet pedig akkor használjuk, ha egy új feltételt kell ellenőriznünk az első `if` után.”

\*\*Példák az osztály részére:\*\*

- \*Tanár:\* „Most nézzünk egy példát, ahol több feltételt is megvizsgálunk!”

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

```javascript

let kor = 18;

if (kor >= 18) {

console.log("Nagykorú vagy.");

} else if (kor >= 14) {

console.log("Kamasz vagy.");

} else {

console.log("Gyermek vagy.");

}

```

- \*Tanár:\* „Ebben az esetben különböző korcsoportokat kezelünk. Az első feltétel, `kor >= 18`, azt vizsgálja, hogy valaki nagykorú-e. Ha nem, akkor megnézzük, hogy kamasz-e (`kor >= 14`). Ha egyik sem igaz, akkor a `else` blokk fut le.”

\*\*Gyakorlati Feladat (10 perc)\*\*

- \*Tanár:\* „Most ti jöttök! Írjatok egy programot, amely ellenőrzi, hogy egy szám pozitív, negatív vagy nulla-e, és ennek megfelelően írja ki a konzolra az eredményt!”

- \*Támogatás az osztályban:\* Járj körbe, figyeld a diákok munkáját, és segíts azoknak, akik elakadnak.

---

\*\*`switch` Szerkezet Bemutatása (10 perc)\*\*

1. \*Tanár:\* „A `switch` szerkezet akkor hasznos, ha egy változó több különböző értéke alapján kell döntést hoznunk. Nézzünk egy példát, ahol a napokat szám szerint kezeljük!”

\*Példa:\*

A képen szöveg, képernyőkép, szoftver, képernyő látható

Automatikusan generált leírás

```javascript

let nap = 3;

switch (nap) {

case 1:

console.log("Hétfő");

break;

case 2:

console.log("Kedd");

break;

case 3:

console.log("Szerda");

break;

default:

console.log("Ismeretlen nap");

}

```

2. \*Tanár:\* „A `switch` összehasonlítja a `nap` változót az összes `case`-ben megadott értékkel. Ha talál egy egyezést, a megfelelő blokk fut le. Mi történik a `break` utasítás nélkül? Igen, a következő blokk is lefutna. A `default` blokk az, ami akkor hajtódik végre, ha egyik `case` sem felel meg.”

\*\*Részletes Magyarázat:\*\*

- \*Tanár:\* „A `break` azért fontos, mert megakadályozza, hogy az összes többi `case` blokk is lefusson, miután megtalálta a megfelelő egyezést. Ha kihagyjuk a `break`-et, akkor az úgynevezett 'fall-through' viselkedés történik, ami általában nem kívánatos.”

\*\*Gyakorlati Feladat (10 perc)\*\*

- \*Tanár:\* „Készítsetek egy `switch` szerkezetet, amely az évszakokat adja vissza egy adott hónap száma alapján. Például 1–3 hónapra 'Tél', 4–6 hónapra 'Tavasz' stb. Használjatok `case`-eket és `break`-eket!”

---

\*\*Óraösszegzés (5 perc)\*\*

- \*Tanár:\* „Ma áttekintettük az `if`, `else if`, `else` és `switch` szerkezeteket. Ezekkel lehetővé válik, hogy programjaink logikusan döntsenek különböző feltételek alapján. Ezek a vezérlési szerkezetek a programozás egyik legfontosabb eszközei. Kérdések?”

- \*Tanár:\* „Ha bármi kérdés van, ne habozzatok kérdezni, és legközelebb is fogunk ezekre építeni!”