**### 8. óra: Haladó ciklusok és ciklusvezérlés**

**- `for in`, `for of` ciklusok, `break` használata**

**- Gyakorlati feladatok: komplexebb ciklusok írása és manipulálása tömbökkel és objektumokkal.**

--

**Óra eleji ismétlés (5 perc)**

* **Kezdés**: "Az előző órán megismerkedtünk a for, while és do...while ciklusokkal. Ma továbblépünk és megvizsgáljuk a for...in és for...of ciklusokat, valamint a ciklusvezérlő utasításokat, például a break és continue használatát."
* **Ismétlő kérdések**:
  + "Ki tudja elmagyarázni, mire jó a for ciklus?"
  + "Mit csinál a while ciklus?"

**Elméleti rész (15 perc)**

1. **for...in ciklus**  
   "A for...in ciklust objektumok tulajdonságainak vagy tömbök indexeinek végigjárására használjuk. Például ha van egy objektumunk, amely egy személy adatait tartalmazza, a for...in ciklussal végigiterálhatunk a tulajdonságokon."
   * **Példa**:



const person = {

name: 'Anna',

age: 25,

city: 'Budapest'

};

for (let key in person) {

console.log(`${key}: ${person[key]}`);

}

* + - **Magyarázat**: "A fenti példában a for...in ciklus az person objektum összes kulcsán végigmegy, és kiírja a kulcsokat és az azokhoz tartozó értékeket."

1. **for...of ciklus**  
   "A for...of ciklust tömbök, iterálható objektumok, például sztringek vagy Set típusok elemeinek végigjárására használjuk."
   * **Példa**:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

javascript

Copy code

const numbers = [1, 2, 3, 4, 5];

for (let num of numbers) {

console.log(num);

}

* + - **Magyarázat**: "Ebben az esetben a for...of ciklus végigmegy a numbers tömb elemein, és minden iterációban az num változóba kerül az aktuális elem értéke."

1. **break utasítás**  
   "A break utasítással kiléphetünk egy ciklusból, mielőtt az a természetes módon befejeződne."
   * **Példa**:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

javascript

Copy code

for (let i = 1; i <= 10; i++) {

if (i === 5) {

break;

}

console.log(i);

}

* + - **Magyarázat**: "Ebben a példában a ciklus megszakad, amint az i értéke 5 lesz."

1. **continue utasítás**  
   "A continue utasítással átugorhatjuk a ciklus aktuális iterációját, és folytathatjuk a következő lépéssel."
   * **Példa**:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

javascript

Copy code

for (let i = 1; i <= 5; i++) {

if (i === 3) {

continue;

}

console.log(i);

}

* + - **Magyarázat**: "Ebben az esetben a ciklus a continue utasítás miatt kihagyja az i = 3 értékű iterációt, és folytatja a következővel."

**Példák és gyakorlatok (15 perc)**

1. **Egyszerű for...in példa**
   * **Feladat**: "Hozz létre egy objektumot, amely tartalmazza egy autó adatait (make, model, year). Írj egy for...in ciklust, amely kiírja az autó összes adatát!"
   * **Megoldás**:

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

javascript

Copy code

const car = {

make: 'Ford',

model: 'Focus',

year: 2020

};

for (let prop in car) {

console.log(`${prop}: ${car[prop]}`);

}

1. **Egyszerű for...of példa**
   * **Feladat**: "Írj egy for...of ciklust, amely kiír egy sztring minden betűjét külön sorba!"
   * **Megoldás**:

A képen képernyőkép, szöveg, Betűtípus látható

Automatikusan generált leírás

javascript

Copy code

const text = 'Hello!';

for (let char of text) {

console.log(char);

}

1. **break és continue használata**
   * **Feladat**: "Írj egy for ciklust, amely 1-től 10-ig számol, de megszakad, ha az aktuális szám 7. Ha a szám páros, ugorja át az iterációt."
   * **Megoldás**:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

javascript

Copy code

for (let i = 1; i <= 10; i++) {

if (i % 2 === 0) {

continue;

}

if (i === 7) {

break;

}

console.log(i);

}

**Gyakorlati feladatok (10 perc)**

1. **for...in feladat**:
   * Hozz létre egy tömböt kedvenc ételekkel, és írd ki az indexeket for...in ciklussal!
2. **for...of feladat**:
   * Hozz létre egy tömböt számokkal, és számold össze az elemek összegét for...of ciklussal!
3. **break használata**:
   * Készíts egy for ciklust, amely 1-től 20-ig számol, de megszakad, ha talál egy 8-cal osztható számot!
4. **continue használata**:
   * Írj egy for ciklust, amely 1-től 15-ig számol, de kihagyja a 4-gyel osztható számokat!

**Óra végi összefoglalás (5 perc)**

* **Kérdések**: "Mi a különbség a for...in és a for...of ciklus között? Mire használható a break és a continue?"
* **Feladat megoldása otthonra**: "Próbáljátok ki a for...of ciklust sztringekkel és Set típusokkal, és írjátok le a tapasztalataitokat!"

Rendben, részletes óratervet készítek a \*\*nyolcadik órára\*\*, amelyben a haladó ciklusokat és ciklusvezérlő utasításokat fogjuk megvizsgálni, különös tekintettel a `for...in` és `for...of` ciklusokra, valamint a `break` és `continue` vezérlő utasításokra. Az óra felépítése:

- \*\*Elméleti ismertetés\*\*: `for...in` és `for...of` ciklusok, `break` és `continue` utasítások.

- \*\*Példák és magyarázatok\*\*.

- \*\*Gyakorlati feladatok\*\*.

- \*\*Óra végi összefoglalás\*\*.

---

### \*\*8. óra: Haladó ciklusok és ciklusvezérlés\*\*

#### \*\*Óra eleji ismétlés (5 perc)\*\*

- \*\*Kezdés\*\*: "Az előző órán megismerkedtünk a `for`, `while` és `do...while` ciklusokkal. Ma továbblépünk és megvizsgáljuk a `for...in` és `for...of` ciklusokat, valamint a ciklusvezérlő utasításokat, például a `break` és `continue` használatát."

- \*\*Ismétlő kérdések\*\*:

- "Ki tudja elmagyarázni, mire jó a `for` ciklus?"

- "Mit csinál a `while` ciklus?"

---

#### \*\*Elméleti rész (15 perc)\*\*

1. \*\*`for...in` ciklus\*\*

"A `for...in` ciklust objektumok tulajdonságainak vagy tömbök indexeinek végigjárására használjuk. Például ha van egy objektumunk, amely egy személy adatait tartalmazza, a `for...in` ciklussal végigiterálhatunk a tulajdonságokon."

- \*\*Példa\*\*:

```javascript

const person = {

name: 'Anna',

age: 25,

city: 'Budapest'

};

for (let key in person) {

console.log(`${key}: ${person[key]}`);

}

```

- \*\*Magyarázat\*\*: "A fenti példában a `for...in` ciklus az `person` objektum összes kulcsán végigmegy, és kiírja a kulcsokat és az azokhoz tartozó értékeket."

2. \*\*`for...of` ciklus\*\*

"A `for...of` ciklust tömbök, iterálható objektumok, például sztringek vagy `Set` típusok elemeinek végigjárására használjuk."

- \*\*Példa\*\*:

```javascript

const numbers = [1, 2, 3, 4, 5];

for (let num of numbers) {

console.log(num);

}

```

- \*\*Magyarázat\*\*: "Ebben az esetben a `for...of` ciklus végigmegy a `numbers` tömb elemein, és minden iterációban az `num` változóba kerül az aktuális elem értéke."

3. \*\*`break` utasítás\*\*

"A `break` utasítással kiléphetünk egy ciklusból, mielőtt az a természetes módon befejeződne."

- \*\*Példa\*\*:

```javascript

for (let i = 1; i <= 10; i++) {

if (i === 5) {

break;

}

console.log(i);

}

```

- \*\*Magyarázat\*\*: "Ebben a példában a ciklus megszakad, amint az `i` értéke 5 lesz."

4. \*\*`continue` utasítás\*\*

"A `continue` utasítással átugorhatjuk a ciklus aktuális iterációját, és folytathatjuk a következő lépéssel."

- \*\*Példa\*\*:

```javascript

for (let i = 1; i <= 5; i++) {

if (i === 3) {

continue;

}

console.log(i);

}

```

- \*\*Magyarázat\*\*: "Ebben az esetben a ciklus a `continue` utasítás miatt kihagyja az `i = 3` értékű iterációt, és folytatja a következővel."

---

#### \*\*Példák és gyakorlatok (15 perc)\*\*

1. \*\*Egyszerű `for...in` példa\*\*

- \*\*Feladat\*\*: "Hozz létre egy objektumot, amely tartalmazza egy autó adatait (`make`, `model`, `year`). Írj egy `for...in` ciklust, amely kiírja az autó összes adatát!"

- \*\*Megoldás\*\*:

```javascript

const car = {

make: 'Ford',

model: 'Focus',

year: 2020

};

for (let prop in car) {

console.log(`${prop}: ${car[prop]}`);

}

```

2. \*\*Egyszerű `for...of` példa\*\*

- \*\*Feladat\*\*: "Írj egy `for...of` ciklust, amely kiír egy sztring minden betűjét külön sorba!"

- \*\*Megoldás\*\*:

```javascript

const text = 'Hello!';

for (let char of text) {

console.log(char);

}

```

3. \*\*`break` és `continue` használata\*\*

- \*\*Feladat\*\*: "Írj egy `for` ciklust, amely 1-től 10-ig számol, de megszakad, ha az aktuális szám 7. Ha a szám páros, ugorja át az iterációt."

- \*\*Megoldás\*\*:

```javascript

for (let i = 1; i <= 10; i++) {

if (i % 2 === 0) {

continue;

}

if (i === 7) {

break;

}

console.log(i);

}

```

---

#### \*\*Gyakorlati feladatok (10 perc)\*\*

1. \*\*`for...in` feladat\*\*:

- Hozz létre egy tömböt kedvenc ételekkel, és írd ki az indexeket `for...in` ciklussal!

2. \*\*`for...of` feladat\*\*:

- Hozz létre egy tömböt számokkal, és számold össze az elemek összegét `for...of` ciklussal!

3. \*\*`break` használata\*\*:

- Készíts egy `for` ciklust, amely 1-től 20-ig számol, de megszakad, ha talál egy 8-cal osztható számot!

4. \*\*`continue` használata\*\*:

- Írj egy `for` ciklust, amely 1-től 15-ig számol, de kihagyja a 4-gyel osztható számokat!

---

#### \*\*Óra végi összefoglalás (5 perc)\*\*

- \*\*Kérdések\*\*: "Mi a különbség a `for...in` és a `for...of` ciklus között? Mire használható a `break` és a `continue`?"

- \*\*Feladat megoldása otthonra\*\*: "Próbáljátok ki a `for...of` ciklust sztringekkel és `Set` típusokkal, és írjátok le a tapasztalataitokat!"

---

Ez az óra felépítése részletesen bemutatja a haladó ciklusokat és vezérlési utasításokat, gyakorlati példákkal, hogy a tanulók könnyen megértsék a koncepciókat.