**### 9. óra: Függvények alapjai**

**- Függvények deklarálása és meghívása**

**- Paraméterek és visszatérési értékek**

**- Gyakorlati feladatok: egyszerű függvények készítése és használata.**

**9. Óra: Függvények alapjai**

**1. Bevezetés (10 perc)**

* **Tanár mondja:** "Ma elkezdjük a függvények témakörét, amely a JavaScript egyik legfontosabb eszköze. A függvényeket érdemes egyfajta "kódszerszámnak" tekinteni, amely segít komplex feladatok elvégzésében, anélkül hogy ismétlődő kódot írnánk. Ezekkel a "szerszámokkal" gyorsabban és hatékonyabban tudunk dolgozni."
* **Fontos kérdés:** "Miért éri meg egy kódrészt többször felhasználni? Gondoljatok arra, hogy ha egy weboldalon többször van szükségünk egy adott műveletre, nem lenne jobb, ha ezt egy helyen definiálnánk, és csak meghívnánk, amikor kell?"
* **Kiemelés:**
  + A függvények csökkentik az ismétlődő kód mennyiségét.
  + Strukturáltabbá és olvashatóbbá teszik a kódunkat.
  + Megkönnyítik a hibakeresést, mert ha egy helyen javítunk valamit, az mindenhol hatásos lesz.

**2. Mi a függvény? (10 perc)**

* **Tanár mondja:** "A függvény egy olyan kódrészlet, amely elvégez egy meghatározott feladatot, majd visszatérhet egy eredménnyel. Képzeljük el, hogy van egy automatánk, amely képes különböző italokat elkészíteni. Az automatát csak el kell indítani, és pontosan tudjuk, hogy mit fog csinálni."
* **Példa bemutatása:** "Képzeljük el, hogy van egy gép, ami almákat facsar. Ha megtanuljuk, hogyan kell elindítani, azt bármikor megismételhetjük, ha almalevet szeretnénk. Ez a függvények lényege: egy adott feladatot elvégezhetünk vele újra és újra, egyszerűen."

**3. Függvény deklarálása és meghívása (15 perc)**

* **Példa bemutatása:**
* function koszontes() {
* console.log('Hello, világ!');
* }

koszontes();

* **Tanár mondja:** "Ez egy nagyon egyszerű függvény. A function kulcsszóval definiáljuk, utána megadjuk a nevét, ami jelen esetben koszontes. A kapcsos zárójelek között határozzuk meg, mit csináljon. Amikor meghívjuk a függvényt, az pontosan azokat a műveleteket hajtja végre, amelyeket ott definiáltunk."
* **Példa előtt rövid ismétlés:** "Kérlek, mondjátok el, mit jelent a console.log()!"
* **Gyakorlati feladat:** "Hozzatok létre egy függvényt, amely kiírja a saját neveteket a konzolra, majd hívjátok meg!"

**4. Paraméterek és visszatérési értékek (20 perc)**

* **Paraméterek magyázata:** "Egy függvény képes adatokat fogadni, amelyek segítségével működése sokoldalú lesz. Ezeket az adatokat paramétereknek nevezzük. Gondoljatok egy pizzakészítő gépre. Ha megadjuk, hogy milyen feltéteket szeretnénk, a gép ennek megfelelően dolgozik."
* **Példa bemutatása:**
* function koszontesNevvel(nev) {
* console.log('Hello, ' + nev + '!');
* }
* koszontesNevvel('Anna');

koszontesNevvel('Bálint');

* **Tanár mondja:** "Itt látjátok, hogy a koszontesNevvel nevű függvény egy nev nevű paramétert kap. Ez lehetővé teszi, hogy ugyanaz a függvény több különböző névre is működjön. Ez az újrahasználhatóság lényege."
* **Gyakorlati feladat:** "Készítsetek egy függvényt, amely két számot fogad és kiírja a konzolra az összegüket!"
* **Visszatérési érték magyázata:** "Nemcsak adatokat tudunk átadni a függvénynek, hanem az eredményt is visszakaphatjuk. Ezt a return kulcsszóval tesszük meg."
* **Példa bemutatása:**
* function osszeadas(a, b) {
* return a + b;
* }
* let eredmeny = osszeadas(5, 3);

console.log('Az eredmény: ' + eredmeny);

* **Tanár mondja:** "A függvény visszatér egy értékkel, amit eltárolhatunk egy változóban, majd felhasználhatjuk a kód többi részében. Ez a logikus programozás alapja."
* **Gyakorlati feladat:** "Készítsetek egy függvényt, amely visszaadja egy szám négyzetét, majd teszteljétek különböző értékekkel!"

**5. Óra zárása és kérdések (10 perc)**

* **Tanár mondja:** "Ma megtanultuk, hogyan hozzunk létre függvényeket, hogyan adjunk át nekik paramétereket és hogyan kapjunk vissza értékeket. Most kérem, hogy akinek van kérdése, térjen rá, és nézzük meg együtt a megoldásokat."
* **Otthoni feladat:** "Otthon hozzatok létre egy függvényt, amely egy tömb elemeinek az átlagát számítja ki, majd hívjátok meg különböző tömbökkel!"

### 9. Óra: Függvények alapjai

#### 1. Bevezetés (10 perc)

- \*\*Tanár mondja:\*\*

"Ma elkezdjük a függvények témakörét, amely a JavaScript egyik legfontosabb eszköze. A függvényeket érdemes egyfajta "kódszerszámnak" tekinteni, amely segít komplex feladatok elvégzésében, anélkül hogy ismétlődő kódot írnánk. Ezekkel a "szerszámokkal" gyorsabban és hatékonyabban tudunk dolgozni."

- \*\*Fontos kérdés:\*\*

"Miért éri meg egy kódrészt többször felhasználni? Gondoljatok arra, hogy ha egy weboldalon többször van szükségünk egy adott műveletre, nem lenne jobb, ha ezt egy helyen definiálnánk, és csak meghívnánk, amikor kell?"

- \*\*Kiemelés:\*\*

- A függvények csökkentik az ismétlődő kód mennyiségét.

- Strukturáltabbá és olvashatóbbá teszik a kódunkat.

- Megkönnyítik a hibakeresést, mert ha egy helyen javítunk valamit, az mindenhol hatásos lesz.

#### 2. Mi a függvény? (10 perc)

- \*\*Tanár mondja:\*\*

"A függvény egy olyan kódrészlet, amely elvégez egy meghatározott feladatot, majd visszatérhet egy eredménnyel. Képzeljük el, hogy van egy automatánk, amely képes különböző italokat elkészíteni. Az automatát csak el kell indítani, és pontosan tudjuk, hogy mit fog csinálni."

- \*\*Példa bemutatása:\*\*

"Képzeljük el, hogy van egy gép, ami almákat facsar. Ha megtanuljuk, hogyan kell elindítani, azt bármikor megismételhetjük, ha almalevet szeretnénk. Ez a függvények lényege: egy adott feladatot elvégezhetünk vele újra és újra, egyszerűen."

#### 3. Függvény deklarálása és meghívása (15 perc)

- \*\*Példa bemutatása:\*\*

```javascript

function koszontes() {

console.log('Hello, világ!');

}

koszontes();

```

- \*\*Tanár mondja:\*\*

"Ez egy nagyon egyszerű függvény. A `function` kulcsszóval definiáljuk, utána megadjuk a nevét, ami jelen esetben `koszontes`. A kapcsos zárójelek között határozzuk meg, mit csináljon. Amikor meghívjuk a függvényt, az pontosan azokat a műveleteket hajtja végre, amelyeket ott definiáltunk."

- \*\*Példa előtt rövid ismétlés:\*\*

"Kérlek, mondjátok el, mit jelent a `console.log()`!"

- \*\*Gyakorlati feladat:\*\*

"Hozzatok létre egy függvényt, amely kiírja a saját neveteket a konzolra, majd hívjátok meg!"

#### 4. Paraméterek és visszatérési értékek (20 perc)

- \*\*Paraméterek magyázata:\*\*

"Egy függvény képes adatokat fogadni, amelyek segítségével működése sokoldalú lesz. Ezeket az adatokat paramétereknek nevezzük. Gondoljatok egy pizzakészítő gépre. Ha megadjuk, hogy milyen feltéteket szeretnénk, a gép ennek megfelelően dolgozik."

- \*\*Példa bemutatása:\*\*

```javascript

function koszontesNevvel(nev) {

console.log('Hello, ' + nev + '!');

}

koszontesNevvel('Anna');

koszontesNevvel('Bálint');

```

- \*\*Tanár mondja:\*\*

"Itt látjátok, hogy a `koszontesNevvel` nevű függvény egy `nev` nevű paramétert kap. Ez lehetővé teszi, hogy ugyanaz a függvény több különböző névre is működjön. Ez az újrahasználhatóság lényege."

- \*\*Gyakorlati feladat:\*\*

"Készítsetek egy függvényt, amely két számot fogad és kiírja a konzolra az összegüket!"

- \*\*Visszatérési érték magyázata:\*\*

"Nemcsak adatokat tudunk átadni a függvénynek, hanem az eredményt is visszakaphatjuk. Ezt a `return` kulcsszóval tesszük meg."

- \*\*Példa bemutatása:\*\*

```javascript

function osszeadas(a, b) {

return a + b;

}

let eredmeny = osszeadas(5, 3);

console.log('Az eredmény: ' + eredmeny);

```

- \*\*Tanár mondja:\*\*

"A függvény visszatér egy értékkel, amit eltárolhatunk egy változóban, majd felhasználhatjuk a kód többi részében. Ez a logikus programozás alapja."

- \*\*Gyakorlati feladat:\*\*

"Készítsetek egy függvényt, amely visszaadja egy szám négyzetét, majd teszteljétek különböző értékekkel!"

#### 5. Óra zárása és kérdések (10 perc)

- \*\*Tanár mondja:\*\*

"Ma megtanultuk, hogyan hozzunk létre függvényeket, hogyan adjunk át nekik paramétereket és hogyan kapjunk vissza értékeket. Most kérem, hogy akinek van kérdése, térjen rá, és nézzük meg együtt a megoldásokat."

- \*\*Otthoni feladat:\*\*

"Otthon hozzatok létre egy függvényt, amely egy tömb elemeinek az átlagát számítja ki, majd hívjátok meg különböző tömbökkel!"