Objects

JavaScript

1. Property-k kilistázása

|  |
| --- |
| */\* Írj egy függvényt, ami paraméterként egy objectet kap és kilistázza egy tömbbe a property-jeit, majd a tömbbel tér vissza.  Input: { alma: 1, körte: 2, saláta: 3, paradicsom: 4 } Output: ["alma", "körte", "saláta", "paradicsom"]  \*/* |

1. Értékek listázása

|  |
| --- |
| */\* Írj egy függvényt, amely paraméterként egy objectet kap, és egy tömbbe átmásolja minden property-jének értékét, visszatérési értéke pedig a tömb. Input: { Lannister: "oroszlán", Stark: "farkas", Targaryen: "sárkány", Baratheon: "szarvas" } Output ["oroszlán", "farkas", "sárkány", "szarvas"] \*/* |

1. Object hossza

|  |
| --- |
| */\* Írj egy függvényt, ami paraméterként egy object-et kap, és visszatérési értéke az object hossza.  Input: {a: 1, b: 2, c: 3} Output 3 \*/* |

1. Property értékének kiíratása

|  |
| --- |
| */\* Írj egy függvényt, amely paraméterként megkapja az library tömböt és kiírja a konzolra a minden könyv readingStatus értékét. \*/* **let** library = [   {  author: 'Bill Gates',  title: 'The Road Ahead',  readingStatus: true  },  {  author: 'Steve Jobs',  title: 'Walter Isaacson',  readingStatus: true  },  {  author: 'Suzanne Collins',  title: 'Mockingjay: The Final Book of The Hunger Games',   readingStatus: false  }]; |

1. Átalakítás tömbbé

|  |
| --- |
| */\* Írj egy függvényt, ami egy objectet kap paraméterként, és egy tömböt ad vissza, amelyben az eredeti object key:value párjai egy újabb tömb elemei.  Input {  alma: "gyümölcs", répa: "zöldség" } Output [["alma", "gyümölcs"], ["répa", "banán"]] \*/*  **function** **transformObjToArray** (obj) {  *// ide írhatod a kódodat* } |

1. Létezik-e?

|  |
| --- |
| */\* Írj egy függvényt, amely két paramétert kap: egy objectet és egy sztringet. A függvény pedig ellenőrzi, hogy az object rendelkezik-e a sztringgel azonos property-vel. Visszatérési értéke egy boolean (true/false). \*/* |

1. Létezik-e?++

|  |
| --- |
| */\* Írj egy függvényt, amelynek két paramétere van: egy object és bármilyen primitív adattípus. Az függvény ellenőrzni, hogy az object bármely property-je rendelkezik a megadott értékkel vagy sem. Visszatérési értéke egy boolean. \*/* |

1. Törlés property alapján

|  |
| --- |
| */\* Írj egy függvényt, ami két paramétert kap: egy objectet, és egy sztringet. A függvény pedig törli a sztringgel egyező property-t az objectből, majd visszatér az új object-tel. Input {1: "a", 2: "b", 3: "c"}, "2" Output {1: "a", 3: "c"} \*/*  **function** **deleteByProperty** (obj, prop) {  *// ide írhatod a kódodat* } |

1. Üres?

|  |
| --- |
| */\* Írj egy függvényt, ami egy object-et kap paraméterként, és megmondja, hogy az object üres-e vagy sem. Az eredmény függvényében a visszatérési értéke true/false.  Input {a: 1} Output false -> az object nem üres  Input {} Output true -> az object üres \*/* |

1. Azonos-e

|  |
| --- |
| */\* Írj egy függvényt, ami két object-et kap paraméterként, és eldönti, hogy a két object azonos-e. Az eredmény függvényében a visszatérési értéke true/false.  Input #1: { a: 1, b: 2, c: 3 }; Input #2: { a: 1, b: 2, c: 3 } Output -> true  Input #1: { a: 1, b: 2, c: 3 }; Input #2: { d: 4, e: 5 } Output -> false \*/* |

1. Ez egy object?

|  |
| --- |
| */\* Írj egy függvényt, ami egy paramétert kap, és eldönti, hogy az a paraméter egy sima object vagy sem. (Tehát nem tömb, vagy szám, sztring...stb.) Az eredmény függvényében true/false értékkel tér vissza. input | output { "1": 123 } | true -> hiszen ez egy object [1, 2, 3] | false -> hiszen ez egy tömb "egykettő' | false -> hiszen ez egy sztring \*/*  **function** **isItAnObject** (obj) {  *// ide írhatod a kódodat* }  Segítség: [link](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/typeof) |