

## 04 OBJEKT, POLE (array), CYKLUS (loop)

povedzme, že vyrábaš eshop ako <https://knihibot.cz/a/667> otvor si túto stránku. v dobe, keď ja píšem tento text, je na nej 12 kníh.

o **každej jednej knihe** potrebuješ evidovať **viac údajov**:

- názov **title**
- autora **author**
- cenu **price**
- ISBN **ISBN**
- popis **about**

(budem tu popisovať aj veci, ktoré som zatiaľ na hodinách neukazoval 😊 , každopádne) ak o jednej veci potrebuješ evidovať viac údajov, v javascript kóde použiješ...

### OBJEKT (Object)

```
let book = {
  title: "Levá ruka tmy",
  author: "Ursula K. Le Guin",
  price: 123,
  ISBN: "9788090182936",
  about: 'Genry Aii je jediným zástupcem lidstva v tomto románu. Jako vyslanec Ekumenu (což je jakési společenství planet) se dostává na Gethen, zimní planetu. Cílem jeho mise je smlouva o připojení Gethenu k Ekumenu. A jak už to tak při střetu dvou různých kultur bývá, nic není tak jednoduché, jak se na první pohled může zdát. Getheňané jsou sice morfologicky podobní lidem, ale jedna významná odlišnost tu je – jsou oboupohlavní a každý měsíc se z nich na pár dní stává muž nebo žena...'
};

// konkrétnu vlastnost získas cez bodku/tečku
console.log( book.title ); // "Levá ruka tmy"
console.log( book.price ); // 123
```

**objekt** sú zložené zátvorky `{}` v nich vymenuješ dvojice `{ vlastnost: hodnota, vlastnost: hodnota, ... }`

**konkrétnu hodnotu získaš cez bodku/tečku:**

```
book.author // "Ursula K. Le Guin"
```

## PRÁCA S OBJEKTOM V REALITE

občas budeš v kóde vytvárať nové objekty. príklad máš hore. ale možno častejšie *budeš pracovať s objektami*, ktoré prídu z externých zdrojov. napríklad z tzv. servera, z iného webu alebo z databázy.

znova si predstav, že vyrábaš eshop. tu je fiktívny kód, ktorý si vyžiada údaje z databázy:

```
// do premennej premennej data sa mi vrátia udaje z databazy
const data = '...kod ktorý si vyžiada udaje z databazy...';
console.log( data );
```

**neustále opakujem, že 100% času máš v prehliadači otvorenú konzolu.** ty nevieš, čo presne sa ti vráti z databázy. ale takmer určite to bude objekt. alebo pole objektov.

ak vo výpise v konzole teraz uvidíš `{}`, tak vieš:

- že sa jedná o objekt
- že objekt drží viac hodnôt
- že ku konkrétnej hodnote sa dostaneš cez tečku:

```
// ak v konzole vidim podobny vypis, vďaka {} viem, že sa jedna o objekt
{ title: "Levá ruka tmy", price: 123 }

// ak chcem získať konkrétne hodnotu jeho vlastnosti price
data.price; // 123
```

## POLE, Array()

**ten istý príklad:** eshop predáva knihy. údaje o každej knihe sú v databáze. ty ako programátor požiadaš databázu o knihy. z databázy sa ti vráti **zoznam 12 kníh**. a ty každú z tých 12 kníh chceš zobrazíť na stránke

keď máš **zoznam vecí**, keď máš kolekciu viacerých vecí, v programovaní tomu hovoríme **pole**. po anglicky **array**. akože... po správnosti by som mal povedať niečo na štýl "*pole je dátová štruktúra schopná držať kolekciu hodnôt alebo premenných*" ale keď s programovaním začíname, pre mňa nie je dôležitý exaktný jazyk. pre mňa je dôležité, aby sme chápali, čo pole je. a pointa je:

**máš viac vecí? máš pole!**

prišlo ti z databázy 12 kníh? to znamená, že to **není jedna kniha** ❌. je to **viac kníh!** ✅ je to kolekcia kníh! je to **zoznam kníh!** je ich niekoľko!

je. to. **POLE!**

máš pole plné kníh.

```
let books = [  
  "Levá ruka tmy",  
  "Zločin a trest",  
  "Hamlet"  
];
```

pole začína hranatými zátvorkami `[]`, v nich sú čiarkou oddelené hodnoty. pole `books` tu má 3 hodnoty typu `String`. pole vie držať akékoľvek hodnoty. môžem tam mať 5 čísel, môžu to byť 3 stringy, môže to byť 12 objektov (napríklad 12 kníh).

k jednotlivým položkám sa dostanem cez tzv. `[index]`. index je číslo. v programovaní číslovať **začínáme od nuly**. takže:

```
books[0]; // "Levá ruka tmy"  
books[2]; // "Hamlet"  
books[1]; // "Zločin a trest"  
books[3]; // undefined
```

(s výnimkou krásneho programovacieho jazyka Lua, ktorý začína od jednotky, jak člověk)

takže ak potrebuješ, vieš nájsť konkrétnu položku poľa. **ale najčastejšie budeš chcieť prebehnúť cez všetky hodnoty poľa**, jednu po druhej, a spacovať ich. napríklad vypísať ich do stránky. alebo do konzole. cez:

## CYKLUS, LOOP (forEach, ...)

toto som na hodinách zatiaľ neukazoval, ale smerujeme k tomu 😊 **cyklus ti dovolí zopakovať riadky kódu x-krát**. existuje mnoho rôznych zápisov pre cyklus, v javascripte ten najpriamočiarejší je `forEach`. čo doslova znamená **preKaždý**.

čiže ak chcem povedať "pre každý prvok poľa `books` sprav výpis do konzole", napíšem:

```
books.forEach(kniha => {  
  // riadky tu vo vnútri sa zopakujú pre každú položku poľa books  
  console.log(kniha);  
});  
  
// v konzole uvidíš  
"Levá ruka tmy"  
"Zločin a trest"  
"Hamlet"
```

keďže v poli `books` mám 3 hodnoty, `console.log` sa zopakuje 3x. každé opakovanie sa nazýva **ITERÁCIA**.

v prvom opakovaní (iterácii) sa do premennej `kniha` uloží "Levá ruka tmy", pretože to je prvá hodnota poľa. `console.log` to vypíše do konzole. v druhej iterácii sa do `kniha` uloží "Zločin a trest", pretože to je druhá hodnota poľa. `console.log` to vypíše do konzole. a takto sa to opakuje pre každú položku poľa.

👉 ďalšie príklady použitia cyklov (ako `for..of`, `for`) máš v #soubory, konkrétne v `03a-pole-cyklus-jsdoc.zip`

*detail:* premennú som si tu nazval `kniha`, za normálnych okolností by som použil anglicky `book`, ale:

- chcem demonštrovať, že je to môj vlastný názov, ktorý som si sám vymyslel
- keďže tam mám premennú `books` s `s` na konci, nechcem, aby sa vám plietlo v hlavne `books` a `book`

čiže ak máš pole `books`, môžeš povedať `books.forEach()` a do vnútra napíšeš funkciu, ktorá sa zopakuje pre každú položku poľa.

## PRÍKLAD S ČÍSLAMI

```
// v poli mozu byt akekolvek typy hodnot, napriklad cisla
let numbers = [420, 69, 0, 11];
let sum = 0;

// pre kazde cislo z pola zopakujeme
numbers.forEach(cislo => {
  // do premennej sum sa pripocita kazde cislo z pola
  sum = sum + cislo;
});

console.log(sum); // 500
```

v poli `numbers` mám 4 čísla. premenná `sum` začína na `0`. v prvej iterácii cyklu bude v premennej `cislo` hodnota `420`. tá sa pripočíta k nule, takže v `sum` je teraz `420`. v druhej iterácii bude v `cislo` hodnota `69`. takže `sum = 420 + 69` a toto pokračuje pre každé číslo v poli.

**pole je zoznam hodnôt.** pole je keď máš viac vecí 😊 a cyklom (ako napríklad `forEach`) vieš prebehnúť každú hodnotu poľa. pole má v sebe viac vecí. a `forEach` vie spustiť kód pre každú z nich. príklad:

## POLE OBJEKTOV [ {}, {}, {}, ... ]

**stále ten istý príklad:** eshop predáva knihy. údaje o každej knihe sú v databáze. ty ako programátor požiadáš databázu o knihy. z databázy sa ti vráti **zoznam 12 kníh**. čiže pole. každá z tých kníh bude objekt. takže máš pole objektov. tvoja úloha je zobraziť ich na stránke.

pozri na tú našu žltú appku, s ktorou na hodinách pracujeme. máme na nej 3 boxy. každý má nadpis a text. v javascript kóde by sme ich mohli reprezentovať takto:

```
const data = [  
  {  
    title: 'Playdate SDK',  
    content: 'Our full development environment, available for Mac, PC,  
and Linux. Includes Lua and C APIs, as well as a Simulator for local  
development, with profiling and more.'  
  },  
  {  
    title: 'Inside Playdate',  
    content: 'Reference documentation for programming Playdate games.  
Covers both <strong>Lua</strong> and <strong>C</strong> programming.'  
  },  
  {  
    title: 'Caps',  
    content: 'Bitmap font editor for Playdate fonts—draw your own  
characters, import desktop fonts, kern and preview them.'  
  }  
];  
  
console.log(data);
```

vidíš `[]` takže máš **po**le.

v poli vidíš 3 `{}` hodnoty oddelené čiarkou a vieš, že `{}` je objekt.

takže máš pole s 3 objektami.

smerujeme k tomu, že cez takéto pole prebehneme v cykle a každú jednu položku poľa nakreslíme na obrazovku a **TOTO JE ONO!** je základ celého webu ako takého. základ väčšiny aplikácií. máš pole plné objektov a v cykle prebehneš cez každý objekt. spracuješ hodnoty v prvom objekte, potom druhom, potom treťom, ...až kým sa neminú.

```
data.forEach(card => {  
  console.log(card);  
});
```

zatiaľ len výpis do konzole, ale onedlho ich spracujeme a vyrobíme z nich HTML kód a vznikne reálna vec  
❤️

**znova:** ja som si tu tie objekty vypísal priamo do kódu. v reálnejšej appke by prišli napríklad z databázy. z databázy by mi prišlo pole objektov. cyklom by som spravil každý z nich. presne ako tomto príklade. a v pondelok ideme na to:

polia. a objekty. a polia plné objektov. ❤️

## ČO ĎALEJ?

je tu veľa slov. ale keď prižmúriš oči, medzi nimi sú aj kusy kódu! ak začínaš, pokús sa skopírať / prepísať tieto kusy kódu do tvojho programu. hrať s nimi. robiť si výpisy. snažiť sa upravovať tie polia a objekty. pridávať do nich nové údaje a robiť si nonstop `console.log` výpisy a sledovať, ako sa menia. ako výpisy reagujú na tvoje zmeny kódu.

polia objektov sú kolekcie `{ }` zátvoriek medzi `[ ]` zátvorkami, oddelené čiarkami, do toho sú tam rôzne apostrofy a úvodzovky a ďalšie hovadiny. keď začínaš, treba si to precvičovať. treba naučiť oči vidieť tie štruktúry a naučiť prsty písať tie znaky a nezblázniť sa z toho. precvičovať si to. cvič cvič cvič 🏃 postupne popriďavam aj úlohy pre lepšie cvičenie 😊