

2 Objects & conditionals

2.1 Creating objects

Objects are variables that can contain many values. For instance, an animal is an object that has a name, is of a particular type and has an age.

In JavaScript you define an object as follows:



```
const animal1 = { name: "Albert", type: "duck", age: 2 }
```

As such you define an object called animal1 with name Albert, it is a duck (the type) and has an age of 2 years.

2.1.1 Exercise: Creating games



Create 4 different game objects in a new file games.js in the folder js.

A game has a name, is of a particular type and has a rating assigned.

If you are not familiar with gaming, here are some examples of popular games:

- ❖ Elden Ring, a fantasy game
- ❖ Horizon Forbidden West, an adventure game
- ❖ Pokemon Legends: Arceus, an RPG game
- ❖ GTA V, an open world game



Evaluation criteria: In game.js you should have:

- ☐ 4 different constants
- ☐ for each const: a name and a type field containing a string, and a rating field containing a float should be defined

2.2 Showing all objects

In order to show a new part of text in a new paragraph on the screen, you can use the function `addStatus` in the JavaScript file `dom.js`. The parameter `status` is the text that is added to the HTML element with id “status” and will be shown on the screen.



```
const addStatus = (status) => { ... }
```

In order to show the sentence Name: Albert – Type: duck – Age: 2 on the screen, you can use the following code:



```
addStatus(`Name: ${animal1.name} - Type: ${animal1.type}  
- Age: ${animal1.age}`)
```

2.2.1 Exercise: Showing games



Show the 4 game objects created in section 3.1 on the overview page.

The result should look as follows

My Games

These are all games in the library

Name: Elden Ring - Type: Fantasy - Rating: 4
Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5
Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3
Name: GTAV - Type: Open World - Rating: 5



Evaluation criteria: You should have defined in `game.js`:

- ☐ a function `printAllGames`
- ☐ a function `toString`
- ☐ the function `printAllGames` calls 4 times the `addStatus` method and 4 times the `toString` method
- ☐ at the end of this file you should have called only the function `printAllGames`

2.2.2 Extension: Showing more games with more information ...



```
Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5 - Favourite: false
Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3 - Favourite: true
Name: GTAV - Type: Open World - Rating: 5 - Favourite: true
Name: Gran Turismo - Type: Car - Rating: 6 - Favourite: true
```



Evaluation criteria: You should only have changed the following in your games.js

- ☐ Added a favourite field containing a boolean to each object
- ☐ added a new const for the Gran Turismo game with all its specific values
- ☐ within the function printAllGames you should only add 1 line of code to add show the new game on the screen
- ☐ within the function toString you should only add 1 little piece of code to show the value of favourite and the extra text indicating whether it is a favourite game or not

2.3 Adding more functionality

2.3.1 Exercise: Showing the average rating of all your games



Calculate and show the average rating of all the games.

The result should look as follows

My Games

These are all games in the library

```
Name: Elden Ring - Type: Fantasy - Rating: 4 - Favourite: false
Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5 - Favourite: false
Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3 - Favourite: true
Name: GTAV - Type: Open World - Rating: 5 - Favourite: true
Name: Gran Turismo - Type: Car - Rating: 6 - Favourite: true
```

Some statistics ...

Average rating: 4.3



Evaluation criteria: In the file game.js you should have defined

- ☐ a function `getAverageRating` calculating the average rating
- ☐ 1 `addStatus` call generating the title
- ☐ 1 `addStatus` call showing the average rating

The average result on the page should be

- ☐ 4.3

2.4 Conditionals

To do something in a specific situation and to do something else in another situation, you can use an if-then-else.



```
if (condition is true) {  
    do something  
} else {  
    do something else  
}
```

An example of such an if-then-else in the animals demo is shown below. If the age of the first object is greater than the age of the second object, assign the first object to result, else if the age of the second object is greater than the age of the first object, assign the second object to the result.



```
const getOldestAnimal = (first, second) => {  
    let result = null  
    if (first.age > second.age) {  
        result = first  
    } else if (first.age < second.age) {  
        result = second  
    }  
    return result;  
}
```

2.4.1 Exercise: Showing game with the highest rating



Add to the statistics the game with the highest ranking

The result should look as follows

My Games

These are all games in the library

Name: Elden Ring - Type: Fantasy - Rating: 4 - Favourite: false

Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5 - Favourite: false

Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3 - Favourite: true

Name: GTAV - Type: Open World - Rating: 5 - Favourite: true

Name: Gran Turismo - Type: Car - Rating: 6 - Favourite: true

Some statistics ...

Average rating: 4.3

Gran Turismo is game with the highest rating: 6



Evaluation criteria: In the file game.js you should have defined

- ☐ a function getHighestRating
- ☐ within this function you should have used multiple times the if
- ☐ this function returns the whole object (not only the highest rating)
- ☐ this function is only called once to show the result on the page, so not multiple times!

2.5 Conditional ternary operator

The if-then-else can also be defined shorter with the conditional ternary operator.



```
condition is true ? do something : do something else
```

An example of such a conditional ternary operator in the animals demo is shown below. If an oldest animal is found, the name of the oldest animal is shown, otherwise the sentence "Animals are equally old." is shown on the page.



```
const oldest = getOldestAnimal(animals1, animals2);
addStatus(`Oldest animal: ${oldest !== null ?
oldest.name : "Animals are equally old."}`);
```

2.5.1 Exercise: Showing your favourite games



Show the list of your favourite games, only the names should be shown on the page.

The result should look as follows

My Games

These are all games in the library

Name: Elden Ring - Type: Fantasy - Rating: 4 - Favourite: false

Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5 - Favourite: false

Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3 - Favourite: true

Name: GTAV - Type: Open World - Rating: 5 - Favourite: true

Name: Gran Turismo - Type: Car - Rating: 6 - Favourite: true

These are all the favourite games in the library

Pokémon Legends: Arceus

GTAV

Gran Turismo

Some statistics ...

Average rating: 4.3

Gran Turismo is game with the highest rating: 6



Evaluation criteria: In the file game.js you should have defined

- ☐ a function printFavouriteGames
- ☐ a function isFavourite = (game)
- ☐ the function printFavouriteGames uses the isFavourite function
- ☐ the function printFavouriteGames uses the conditional ternary operator

2.5.2 Exercise: Adding more game objects



Add 2 more games at the beginning of the games.js file.

The result should look as follows

My Games

These are all games in the library

Name: Fifa23 - Type: Football - Rating: 7 - Favourite: false

Name: AOTennis 2 - Type: Tennis - Rating: 2 - Favourite: true

Name: Elden Ring - Type: Fantasy - Rating: 4 - Favourite: false

Name: Horizon Forbidden West - Type: Adventure - Rating: 3.5 - Favourite: false

Name: Pokémon Legends: Arceus - Type: RPG - Rating: 3 - Favourite: true

Name: GTAV - Type: Open World - Rating: 5 - Favourite: true

Name: Gran Turismo - Type: Car - Rating: 6 - Favourite: true

These are all the favourite games in the library

Pokémon Legends: Arceus

GTA V

Gran Turismo

AOTennis 2

Some statistics ...

Average rating: 4.357142857142857

Fifa23 is game with the highest rating: 7



Evaluation criteria: In the file game.js you should have

- ☐ added 2 new objects at the first two lines of this file
- ☐ added 2 lines of code to the printAllGames function
- ☐ added 2 lines of code to the printFavouriteGames function
- ☐ added some code to the getHighestRating function
- ☐ added some code to the getAverageRating function