

The background features a large, abstract, wavy shape in shades of green and yellow, resembling a stylized wave or a flowing ribbon, set against a white background.

JavaScript



JS Masyvai

JavaScript masyvai

...

Spread operator - išskaido elementus iš masyvo.

```
const arrValue = ['My', 'name', 'is', 'Jack'];  
  
console.log(arrValue);    // ["My", "name", "is", "Jack"]  
console.log(...arrValue); // My name is Jack
```

```
const arr1 = ['one', 'two'];  
const arr2 = [...arr1, 'three', 'four', 'five'];  
  
console.log(arr2);  
// Output:  
// ["one", "two", "three", "four", "five"]
```

JavaScript masyvai

Masyvų dubliavimas/kopijavimas

```
const sheeps = ['🐑', '🐑', '🐑'];  
  
// old way  
const cloneSheeps = sheeps.slice();  
  
// ES6 way  
const cloneSheepsES6 = [...sheeps];
```

```
const numbersOne = [1, 2, 3];  
const numbersTwo = [4, 5, 6];  
const numbersCombined = [...numbersOne, ...numbersTwo];
```

JavaScript masyvų metodai

map() - sukuria naują masyvą, su masyvo elementais atlikdamas veiksmus, aprašytus funkcija (atlieka elementų transformavimą).

Pvz. sukuria naują masyvą, padaugindamas kiekvieną elementą iš dviejų.

```
var numbers1 = [45, 4, 9, 16, 25];  
var numbers2 = numbers1.map(myFunction);  
  
function myFunction(value, index, array) {  
    return value * 2;  
}
```

Funkcija myFunction yra vadinama **callback function**.

JavaScript masyvų metodai

forEach() - atlieka funkcijoje aprašytus veiksmus su kiekvieno masyvo elementu, tame pačiame masyve nekuriant naujo, nieko negrąžina, return ignoruojamas.

Kadangi nėra return, turim su funkcijos rezultatais kažką daryti, pvz. spausdinti, nes funkcijai įvykus jos skaičiavimai bus ištrinti:

```
["Bilbo", "Gandalf", "Nazgul"].forEach((item, index, array) =>
  alert(`${item} is at index ${index} in ${array}`));
});
```

Arba rankiniu būdu priskiriame pačių sukurtam naujam masyvui:

```
const apps = ['WhatsApp', 'Instagram', 'Facebook'];
const playStore = [];

apps.forEach(function(item){
  playStore.push(item)
});

console.log(playStore);
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodai

filter() - sukuria naują masyvą iš funkcijoje aprašytą sąlygą atitinkančių elementų.

```
1 let results = arr.filter(function(item, index, array) {  
2   // if true item is pushed to results and the iteration continues  
3   // returns empty array if nothing found  
4 });
```

```
1 let users = [  
2   {id: 1, name: "John"},  
3   {id: 2, name: "Pete"},  
4   {id: 3, name: "Mary"}  
5 ];  
6  
7 // returns array of the first two users  
8 let someUsers = users.filter(item => item.id < 3);  
9  
10 alert(someUsers.length); // 2
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodai

find() - grąžina masyvo elementą, kuris pirmas atitinka paieškos funkcijoje nurodytą sąlygą.

```
1 const array1 = [5, 12, 8, 130, 44];
2
3 const found = array1.find(element => element > 10);
4
5 console.log(found);
6 // expected output: 12
7
```

Pavyzdyje parodyta paieškos funkcija callback turi vieną argumentą - masyvo elementą.

Taip pat gali turėti elemento indeksą ir patį masyvą:

(currentValue, index, arr) => {...}

Pvz.

```
1 const inventory = [
2   {name: 'apples', quantity: 2},
3   {name: 'bananas', quantity: 0},
4   {name: 'cherries', quantity: 5}
5 ];
6
7 function isCherries(fruit) {
8   return fruit.name === 'cherries';
9 }
10
11 console.log(inventory.find(isCherries));
12 // { name: 'cherries', quantity: 5 }
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodai

reduce() - šis metodas iteruoja masyvą nuo pradžios iki galo, t.y. ima po vieną masyvo reikšmę ir funkcijos pagalba atlieka skaičiavimus, tokius, kad pagal tam tikras taisykles masyvo reikšmės tampa viena reikšme. Pavyzdžiui susumuoja visas masyvo reikšmes.

```
//Implementation One  
let answer1 = array.reduce(callback, initialValue);  
  
//Implementation Two  
let answer2 = array.reduce(callback);
```

reduce metodas priima **du parametrus**:

callback - tai funkcija, kuri atlieka skaičiavimus ir grąžina rezultatą, vadinama **reducer**.

initialValue – tai pradinė reikšmė, nuo kurios pradėsime skaičiavimus. Nebūtina nurodyti. Kai ji nenurodyta pradinė reikšmė yra masyvo pirmas elementas arr[0].



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodas **reduce**

```
function callbackFn(previousValue, currentValue, currentIndex, array) { /**/ }
```

callback (reducer) priima keturis parametrus:

previousValue – tai reikšmė gauta iš ankstesnio callback kvietimo, pirmą prasisukimą ši reikšmė lygi **initialValue**, jei initialValue nenurodyta, tai masyvo pirmas elementas.

currentValue – tai einamasis masyvo elementas. Pirmu prasisukimu jei nurodyta initialValue tai currentValue yra arr[0], priešingu atveju arr[1].

currentIndex – tai einamojo masyvo elemento indeksas. Pirmu prasisukimu currentIndex yra 0, jei nurodyta initialValue ir 1 jei initialValue nenurodyta.

array – masyvas, kuriam kviečiame metodą reduce



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodas **reduce**

Lentelėje pavaizduota, kokius parametrus **reduce** metodas paduoda callback funkcijai ją kviesdamas pirmą kartą (pirmu prasisukimu), priklausomai nuo to, ar paduodama `initialValue` ar ne.

```
let answer = array.reduce(function(totalResult, currValue, currIndex, array){}, initialValue);
```

<code>initialValue</code>	<code>previousValue</code>	<code>currentValue</code>
passed	<code>initialValue</code>	<code>array[0]</code>
not passed	<code>array[0]</code>	<code>array[1]</code>



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodas **reduce**

```
//Declare an array of numbers
var array = [1,2,3,4,5];

//Declare a function that calculates and returns the product

function productArray(totalResult,currValue){
    |   |   |   return totalResult*currValue;
}

//Declare an answer variable to store the final result
//The reduce function callback is the productArray method
let answer = array.reduce(productArray);
console.log("Answer with no initial value")
console.log(answer);

console.log("Given an initial value of 2. The answer should be doubled")
let answer2 = array.reduce(productArray,2);
console.log(answer2);
```



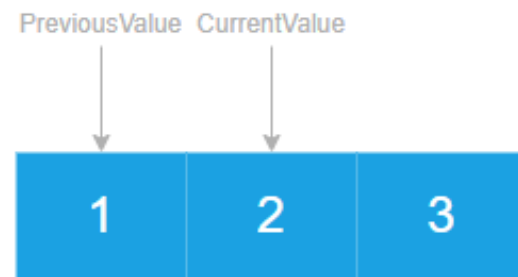
VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų metodas **reduce**

Apskaičiuosime už kokią sumą prekių yra krepšelyje.

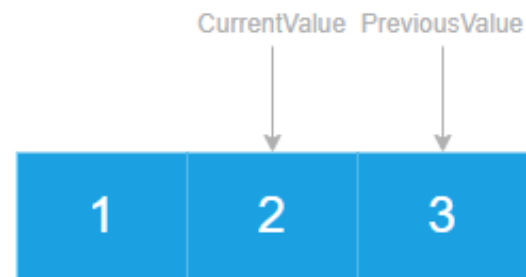
```
let shoppingCart = [  
  {  
    product: 'phone',  
    qty: 1,  
    price: 500,  
  },  
  {  
    product: 'Screen Protector',  
    qty: 1,  
    price: 10,  
  },  
  {  
    product: 'Memory Card',  
    qty: 2,  
    price: 20,  
  },  
];
```

```
let total = shoppingCart.reduce(function (previousValue, currentValue) {  
  return previousValue + currentValue.qty * currentValue.price;  
}, 0);
```



Forward

reduce()



Backward

reduceRight()



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų iteracija

for in:

```
var peoples = ["Greg", "Mary", "Devon", "James"];  
for (var index in peoples) {  
    console.log(peoples[index]);  
}
```

tradicinis for:

```
var peoples = ["Greg", "Mary", "Devon", "James"];  
for (var i=0; i<=peoples.length-1; i++) {  
    console.log(peoples[i]);  
}
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript masyvų praktinės užduotys

Projektas Knygynas

Reduce method



JS Objektai



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

Objektas, tai duomenų tipas, skirtas saugoti sudėtingesnės struktūros duomenis.

Objektai programavimo kalboje yra analogai realiame gyvenime esantiems objektams, pvz. automobilis, katė, studentas, segtuvas ir panašiai.

Objektai turi savybes ir metodus.


Savybės apibūdina koks objektas yra, pvz. objektas automobilis: koks pavadinimas, kiek metų, kas gamintojas, kokia spalva ir t.t.

Metodai reiškia veiksmą, elgseną ir aprašo, ką objektas moka daryti: judėti, keisti spalvą, kalbėti ir t.t. Metodai aprašomi funkcijomis.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

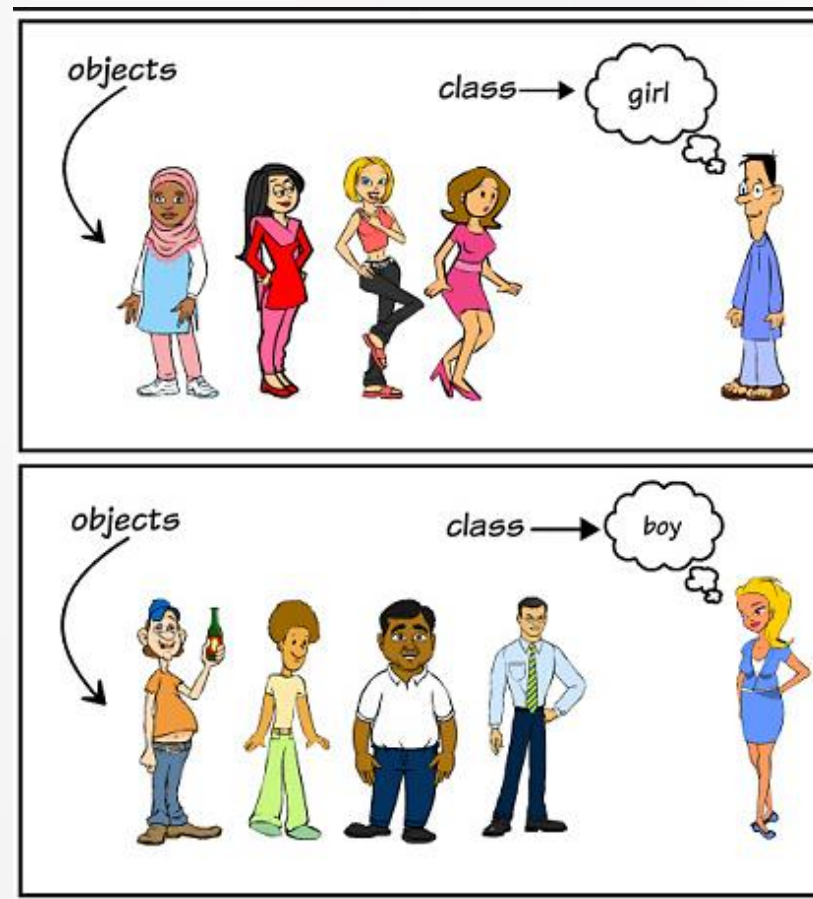
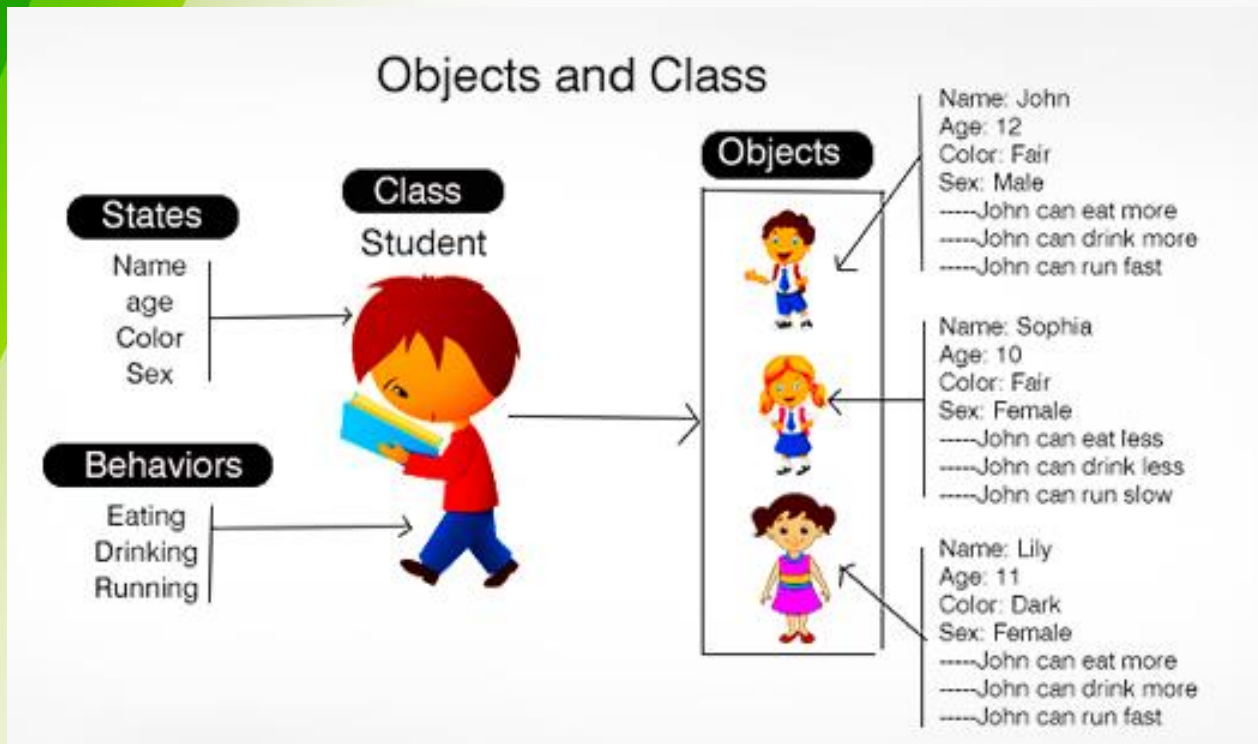
Object	Properties	Methods
	<pre>car.name = Fiat car.model = 500 car.weight = 850kg car.color = white</pre>	<pre>car.start() car.drive() car.brake() car.stop()</pre>



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

Klasė (class) yra ruošinys objektams kurti.





VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript paprasto objekto anatomija

property name
arba **key**

metodas

```
const myObject = {  
  city: "Madrid",  
  greet() {  
    console.log(`Greetings from ${this.city}`);  
  },  
};  
myObject.city; //Madrid  
myObject.greet(); // Greetings from Madrid
```

property value
arba **value**

```
▼ {city: 'Madrid', greet: f} ⓘ  
  city: "Madrid"  
  ► greet: f greet()  
  ▼ [[Prototype]]: Object  
    ► constructor: f Object()  
    ► hasOwnProperty: f hasOwnProperty()  
    ► isPrototypeOf: f isPrototypeOf()  
    ► propertyIsEnumerable: f propertyIsEnumerable()  
    ► toLocaleString: f toLocaleString()  
    ► toString: f toString()  
    ► valueOf: f valueOf()  
    ► __defineGetter__: f __defineGetter__()  
    ► __defineSetter__: f __defineSetter__()  
    ► __lookupGetter__: f __lookupGetter__()  
    ► __lookupSetter__: f __lookupSetter__()  
    ► __proto__: (...)  
    ► get __proto__: f __proto__()  
    ► set __proto__: f __proto__()
```

Objektai savybes ir metodus paveldi iš prototipų, kurie pasiekiami per `__proto__`.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

Objektus JavaScript kalboje galime kurti 4 būdais:
object literal, su raktažodžiu new, funkcijų pagalba ir klasių pagalba.

Raktinis žodis **this** rodo į metodo savininką, t.y. objektą.

1. Object Literal:

```
1  ✓ let person = {  
2    name: 'John',  
3    age: 50,  
4  
5  ✓  sayHi(){  
6    |    return `${this.name} says Hi!`  
7    |  }  
8  }  
9  
10 console.log(person.sayHi());
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

Object Literal objektą galima sukurti iš kintamųjų:

```
const userName = 'John';  
const age = 42;  
const loggedIn = true;  
  
// sukuriame objektą  
const myObj = {userName, age, loggedIn};  
  
console.log(myObj);
```

Rezultatas:

```
▼ (3) {userName: "John", age: 42, loggedIn:...}  
  userName: "John"  
  age: 42  
  loggedIn: true
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

2. Funkcijos pagalba

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Object Constructors</h2>

<p id="demo"></p>

<script>

// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.name = function() {
    return this.firstName + " " + this.lastName
  };
}

// Create a Person object
var myFather = new Person("John", "Doe", 50, "blue");

// Display full name
document.getElementById("demo").innerHTML =
"My father is " + myFather.name();

</script>

</body>
</html>
```

Funkcijos vardas rašomas iš didžiosios raidės.

Funkcijos konstruktoriaus pagalba sukuriami to paties tipo objektai.

Jų gali būti keletas.

Tai senesnis būdas kurti objektus JS kalboje.

Taip sukurtam objektui negalime pridėti nei metodo nei savybės tiesiogiai:

Person.nationality = "English";

Savybių ir metodų pridėjimui naudojami prototipai.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų prototipai

Kai objektai kuriami su funkcija, **savybėms ir metodams pridėti konstruktoriams naudojami Prototipai**. Visi objektai paveldi savybes iš prototipo. Prototipas yra objekto pirmtakas.

```
<script>
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

Person.prototype.nationality = "English";

var myFather = new Person("John", "Doe", 50, "blue");
document.getElementById("demo").innerHTML =
"The nationality of my father is " + myFather.nationality;
</script>
```

```
<script>
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

Person.prototype.name = function() {
  return this.firstName + " " + this.lastName
};

var myFather = new Person("John", "Doe", 50, "blue");
document.getElementById("demo").innerHTML =
"My father is " + myFather.name();
</script>
```

```
▼ Person {firstName: "John", LastName: "Doe", age: 50, eyeColor: "blue"} ⓘ
  firstName: "John"
  lastName: "Doe"
  age: 50
  eyeColor: "blue"
  ▼ __proto__:
    nationality: "English"
    ► constructor: f Person(first, last, age, eye)
    ► __proto__: Object
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

3. JavaScript klasės -

naujausias būdas objektams kurti

```
1  class User {  
2  
3      constructor(name) {  
4          this.name = name;  
5      }  
6  
7      sayHi() {  
8          alert(this.name);  
9      }  
10  
11 }  
12  
13 // Usage:  
14 let user = new User("John");  
15 user.sayHi();
```

Klasė tai šablonas, pagal kurį sukursime daug to paties tipo objektų.

Klasės JavaScript yra nuo ES6 versijos.

Tai šiuolaikinis, modernus būdas kurti objektus.

Klasės vardas rašomas iš didžiosios raidės.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript klasės paveldėjimas

```
1 class Animal {
2   constructor(name) {
3     this.speed = 0;
4     this.name = name;
5   }
6   run(speed) {
7     this.speed = speed;
8     alert(`${this.name} runs with speed ${this.speed}.`);
9   }
10  stop() {
11    this.speed = 0;
12    alert(`${this.name} stands still.`);
13  }
14 }
15
16 let animal = new Animal("My animal");
```

```
1 class Rabbit extends Animal {
2   hide() {
3     alert(`${this.name} hides!`);
4   }
5 }
6
7 let rabbit = new Rabbit("White Rabbit");
8
9 rabbit.run(5); // White Rabbit runs with speed 5.
10 rabbit.hide(); // White Rabbit hides!
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektai

Kai objektą kuriame iš klasės, jo sudėtį galime pamatyti išvedę į konsolę sukurta objektą.

```
console.log(hero1);
```

Konsolėje, kur matomas išvestas objektas matome: savybes, konstruktorių ir metodus (prototype).

Output

```
Hero {name: "Varg", level: 1}
```

```
__proto__:
```

```
▶ constructor: class Hero
```

```
▶ greet: f greet()
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su objektais

Objektų savybių ir metodų iškvietimas:

```
const gimli = {  
  name: "Gimli",  
  race: "dwarf",  
  weapon: "axe",  
  greet: function() {  
    return `Hi, my name is ${this.name}!`;  
  },  
};
```

Savybių iškvietimas:

```
// Retrieve the value of the weapon property  
gimli.weapon;
```

Output
"axe"

Metodų iškvietimas:

```
gimli.greet();
```

Output
"Hi, my name is Gimli!"



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su objektais

Objektų savybių ir metodų pridėjimas ir keitimas ir šalinimas:

Pridėjimas:

```
// Add new age property to gimli  
gimli.age = 139;
```

```
// Add new fight method to gimli  
gimli.fight = function() {  
    return `Gimli attacks with an ${this.weapon}.`;  
}
```

Savybių keitimas:

```
// Update weapon from axe to battle axe  
gimli.weapon = "battle axe";
```

Šiuo būdu keičiamos tik savybės.

Savybių šalinimas:

```
// Remove weapon from gimli  
delete gimli.weapon;
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su objektais

Objektų savybių iškvietimas ir pakeitimas galimas ir per laužtinius skliaustus:

```
1  const myCar={  
2      make: "BMW",  
3      model: "X1",  
4      year: 2000  
5  }
```

```
myCar['make'] = 'Ford';  
myCar['model'] = 'Mustang';  
myCar['year'] = 1969;
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su objektais

Objekto savybių pakeitimas nurodant savybes per kintamuosius.

```
1  const myCar={  
2      make: "BMW",  
3      model: "X1",  
4      year: 2000  
5  }
```

```
7  // access different properties by changing the contents of the variable  
8  
9  let propertyName = 'make';  
10 myCar[propertyName] = 'Audi';  
11  
12  
13 propertyName = 'model';  
14 myCar[propertyName] = 'Mustang';  
15  
16 propertyName='year';  
17 myCar[propertyName] = "2000";  
18  
19  
20 console.log(myCar);  
21  
22 // [Log] {make: 'Ford', model: 'Mustang'}
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų metodas

Object.keys()

Sukuria objektų savybių pavadinimų (keys) masyvą. Gali būti naudojamas išvesti savybes su pavadinimais (keys and values)

```
// Initialize an object
const employees = {
  boss: 'Michael',
  secretary: 'Pam',
  sales: 'Jim',
  accountant: 'Oscar'
};

// Get the keys of the object
const keys = Object.keys(employees);

console.log(keys);
```

Output

```
["boss", "secretary", "sales", "accountant"]
```

```
// Iterate through the keys
Object.keys(employees).forEach(key => {
  let value = employees[key];

  console.log(`${key}: ${value}`);
});
```

Output

```
boss: Michael
secretary: Pam
sales: Jim
accountant: Oscar
```

Pasinaudodami Object.keys() nustatykite objekto dydį - kiek turi savybių.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų metodas **Object.values()**

Sukuria objektų savybių (values) masyvą.

```
// Initialize an object
const session = {
  id: 1,
  time: `26-July-2018`,
  device: 'mobile',
  browser: 'Chrome'
};

// Get all values of the object
const values = Object.values(session);

console.log(values);
```

Output

```
[1, "26-July-2018", "mobile", "Chrome"]
```

Object.keys() ir Object.values() yra skirti objekto duomenims paimti.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų metodas

Object.assign()

Skirtas objekto duomenims (savybėms ir reikšmėms) nukopijuoti iš vieno objekto į kitą.

```
// Initialize an object
const name = {
  firstName: 'Philip',
  lastName: 'Fry'
};

// Initialize another object
const details = {
  job: 'Delivery Boy',
  employer: 'Planet Express'
};

// Merge the objects
const character = Object.assign(name, details);

console.log(character);
```

Output

```
{firstName: "Philip", lastName: "Fry", job: "Delivery Boy", employer: "Planet Express"}
```

```
1 // Initialize an object
2 const name = {
3   firstName: 'Philip',
4   lastName: 'Fry'
5 };
6
7 //Kopijuojamos savybės iš name į naują tuščią objektą character
8 const character = Object.assign({}, name);
9 character.nationality = "LT";
10 console.log(character);
```

Objekto kopijavimas į tuščią



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų spread operatorius

...

```
const circle = {  
  radius: 10  
};
```

```
const coloredCircle = {  
  ...circle,  
  color: 'black'  
};  
  
console.log(coloredCircle);
```

Rezultatas:

```
{  
  radius: 10,  
  color: 'black'  
}
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų spread operatorius

...

```
const circle = {  
  radius: 10  
};  
  
const style = {  
  backgroundColor: 'red'  
};  
  
const solidCircle = {  
  ...circle,  
  ...style  
};  
  
console.log(solidCircle);
```

Gali būti
naudojamas objektų
jungimui.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų spread operatorius

...

```
1  let product1={
2    title: "Car",
3    price: 25000
4  }
5
6  let product2={
7    title: "Book",
8    price: 451
9  }
10
11 let cart={
12   ...product1,
13   ...product2
14 }
15
16 console.log(cart);
```

Jungiant objektus su vienodais object keys, savybės perrašomos, t.y. lieka tik paskutinis objektas.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript spread perduodant funkcijoms parametrus

...

```
const numbers = [1, 3, 5, 7];  
  
function addNumbers(a, b, c, d) {  
  return a + b + c + d;  
}  
  
console.log(addNumbers(...numbers));
```

Rezultatas: 16



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript rest operatorius

...

```
// Use rest to enclose the rest of specific user-supplied values into an array:  
function myBio(firstName, lastName, ...otherInfo) {  
    return otherInfo;  
}  
  
// Invoke myBio function while passing five arguments to its parameters:  
myBio("Oluwatobi", "Sofela", "CodeSweetly", "Web Developer", "Male");  
  
// The invocation above will return:  
["CodeSweetly", "Web Developer", "Male"]
```

Naudojamas paduodant parametrus funkcijai.
Trys taškiukai prieš atskirus parametrus funkcijoje juos sujungia į masyvą.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų destruktorius

```
1  var employee = {    // Object we want to destructure
2      |  firstname: 'Jon',
3      |  lastname: 'Snow',
4      |  dateofbirth: '1990'
5  };
6
7  // Destructuring the object into our variables
8  var { firstname, lastname, dateofbirth } = employee;
9
10
11  console.log( firstname);
12  console.log(lastname);
13  console.log(dateofbirth)
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript objektų destruktorius su object keys pervardijimu

```
1  ✓ var employee = {    // Object we want to destructure
2    |   firstname: 'Jon',
3    |   lastname: 'Snow',
4    |   dateofbirth: '1990'
5    | };
6
7    // Destructuring the object into our variables
8    var { firstname:fn, lastname:ln, dateofbirth:db} = employee;
9
10   console.log(fn);
11   console.log(ln);
12   console.log(db)
```

The background features a series of overlapping, wavy bands in various shades of green and yellow, creating a dynamic, organic feel. The top portion is a solid, vibrant green, while the lower portion transitions into lighter, more complex wavy patterns in shades of lime green and pale yellow.

JS Moduliai



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Kai kodas tampa didelės apimties, patartina jį suskaidyti į mažesnes dalis – atskirus failus, kurie vadinami **moduliais**.

Išskaidoma pagal kodo paskirtį.

Modulio failo plėtinys: ***.mjs arba *.js**. Patartina naudoti *.js.

Taigi modulis yra atskiras nedidelis failas su kodu (skriptu).

Moduliai gali naudotis ir paleisti vienas kito skriptus - "bendrauti".

Export pažymėti kintamieji ir funkcijos gali būti pasiekiami modulio išorėje. Eksportuoti galime kintamuosius, konstantas, funkcijas ir klases, kurie yra top level. Pvz. Iš funkcijos vidaus išeksportuoti nepavyks.

Import leidžia importuoti funkcionalumą iš modulių.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai Export (named export)

Tarkime turime failą sayHi.js , kuris išeksportuoja funkciją:

```
1 // 📁 sayHi.js
2 export function sayHi(user) {
3     alert(`Hello, ${user}!`);
4 }
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Import

Kitas failas main.js gali importuoti ir naudoti sayHi.js išeksportuotą funkciją. **Tai vadinama named import.**

```
1 // 📁 main.js
2 import {sayHi} from './sayHi.js';
3
4 alert(sayHi); // function...
5 sayHi('John'); // Hello, John!
```

Funkcijos vardas, turi būti toks pat,
kaip faile sayHi.js

Kelias iki failo su išeksportuojama
funkcija - path



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Import

Jei importuojami keli elementai, jie atskiriami kableliais:

```
import { name, draw, reportArea, reportPerimeter } from './modules/square.js';
```

Importuojamus/exportuojamus elementus galime pervardinti:

```
1 // inside module.js
2 export {
3   function1 as newFunctionName,
4   function2 as anotherNewFunctionName
5 };
6
7 // inside main.js
8 import { newFunctionName, anotherNewFunctionName } from './modules/module.js';
```

```
1 // inside module.js
2 export { function1, function2 };
3
4 // inside main.js
5 import { function1 as newFunctionName,
6         function2 as anotherNewFunctionName } from './modules/module.js';
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Modulių susiejimas su HTML

Jungiant su HTML būtina naršyklei nurodyti, kad tai modulis.

```
JS test.js > ...  
1   export function sayHi(user) {  
2   |   alert(`Hi, ${user}`);  
3   }
```

```
JS test2.js  
1   import { sayHi } from "../test.js";  
2  
3   sayHi("Rita");
```

```
index.html > ...  
69   |   </div>  
70   </div>  
71  
72   <script type="module" src="test2.js"></script>  
73 </body>  
74  
75 </html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Modulių ir standartinių skriptų skirtumai

- Moduliai neveikia be serverio.
- Moduliai naudoja "strict mode" automatiškai.
- Modulo skriptai yra pasiekiami tik tame skripte, kuriame yra atliktas importas, visuose kituose pasiekiami nėra.
- Modulis suveikia vieną kartą, jį importavus.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai

Default Export

Default export yra naudojamas tada, kai norime išeksportuoti vieną elementą: funkciją, kintamąjį...

```
// file math.js
export default function square(x) {
  return x * x;
}

//while importing square function in test.js
import square from './math';
console.log(square(8)) //64
```

Funkcijos/kintamojo vardas rašomas be {} ir nebūtinai turi sutapti su išeksportuojamos funkcijos/kintamojo vardu



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai pavyzdys

```
// User.js
let userCount = 0

class User {
  constructor(name, age) {
    this.name = name
    this.age = age
    userCount++
  }
}

function printName(user) {
  console.log(`User has the name ${user.name}`)
}

function printAge(user) {
  console.log(`${user.name} is ${user.age} years old`)
}
```

```
// script.js
const user = new User("Kyle", 26)
printName(user)
// User has the name Kyle
printAge(user)
// Kyle is 26 years old
```

```
<!-- index.html -->
<script src="User.js"></script>
<script src="script.js"></script>
```

Jei nenaudosime modulių, abu js failus reiks prijungti prie html. Reikia užtikrinti, kad jungiant User.js parašytas pirmas už script.js. Dar viena problema, kad kintamieji ir funkcijos yra prieinamos kitiems skriptams ir lengva juos perrašyti.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai pavyzdys

Exporting

```
// User.js
let userCount = 0

class User {
  constructor(name, age) {
    this.name = name
    this.age = age
    userCount++
  }
}

function printName(user) {
  console.log(`User has the name ${user.name}`)
}

function printAge(user) {
  console.log(`${user.name} is ${user.age} years old`)
}

export default User
export printName
export printAge
```

Importing

```
// script.js
import User, { printAge, printName } from './User.js'

const user = new User("Kyle", 26)
printName(user)
// User has the name Kyle
printAge(user)
// Kyle is 26 years old
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript moduliai užduotis

For this assignment you should create three JavaScript files:

src/helpers.js

This file should export two array helper functions:

- `choice(items)`: returns a randomly selected item from array of items
- `remove(items, item)`: removes the first matching item from items, if item exists, and returns it. Otherwise returns undefined.

src/foods.js

This file should export this array of fruits:

```
[  
  "🍇", "🍌", "🍓", "🍊", "🍋", "🍍", "🍎", "🍓",  
  "🍇", "🍌", "🍓", "🍊", "🍋", "🍍", "🍎",  
];
```

src/index.js

This file should import the fruits and both array helpers. It should then:

- Randomly draw a fruit from the array
- Log the message "I'd like one *RANDOMFRUIT*, please."
- Log the message "Here you go: *RANDOMFRUIT*"
- Log the message "Delicious! May I have another?"
- Remove the fruit from the array of fruits
- Log the message "I'm sorry, we're all out. We have *FRUITSLEFT* left."

The background features a large, abstract shape composed of overlapping wavy bands. The top band is a vibrant green, while the bottom band is a lighter, yellowish-green. The shape flows from the left side of the frame towards the right, creating a sense of movement.

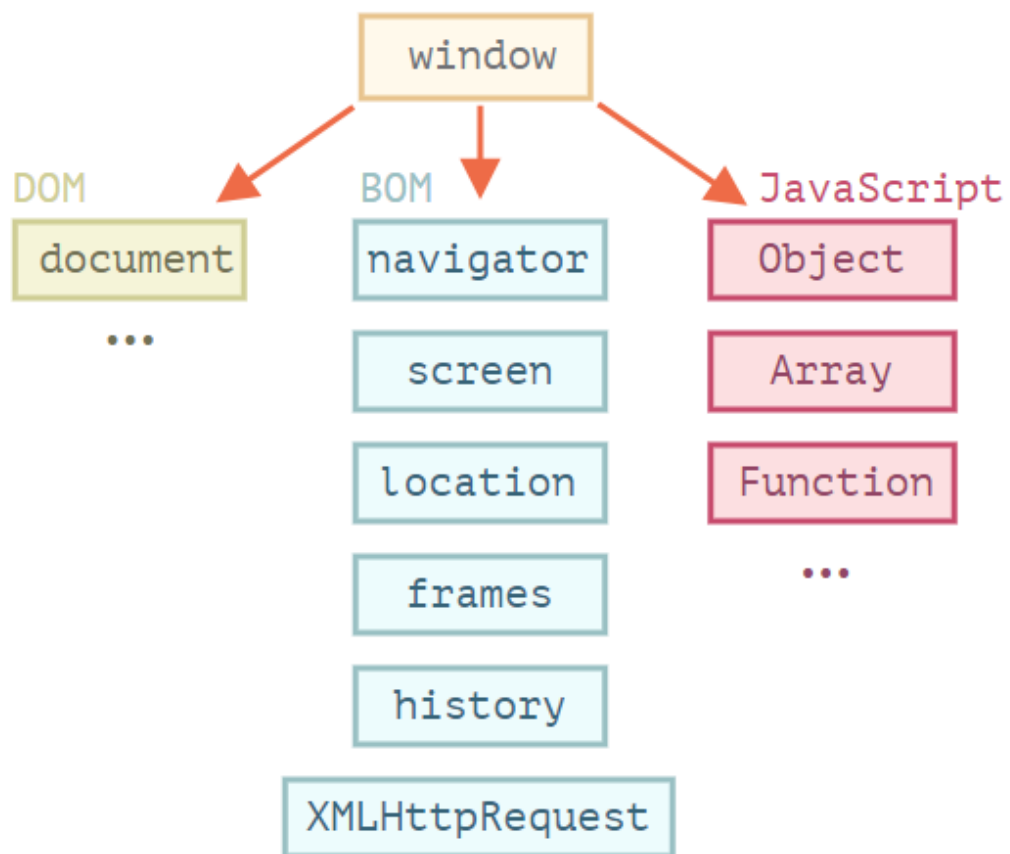
JS DOM



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

Kai JS kodą paleidžiame naršyklėje:
window – root objektas

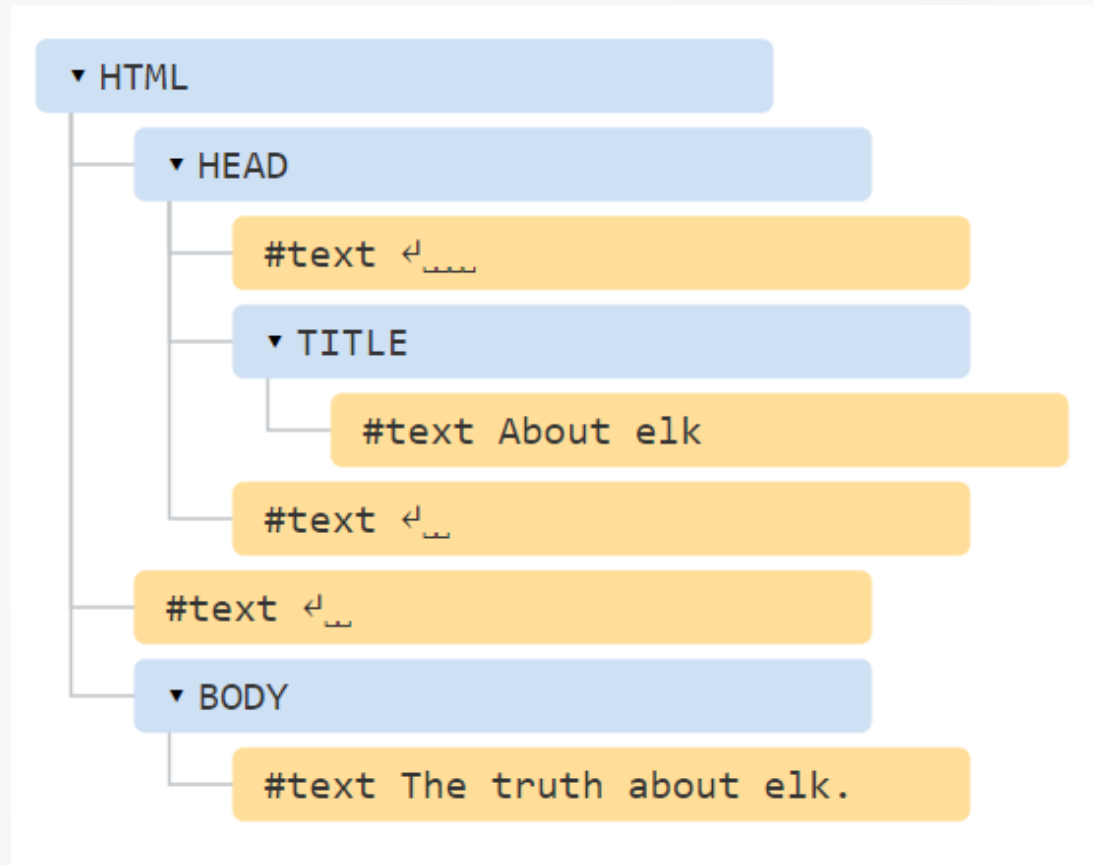




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>About elk</title>
5 </head>
6 <body>
7   The truth about elk.
8 </body>
9 </html>
```



HTML elementai, jų tekstai, tarpai ir perėjimai į kitą eilutę (tarp elementų), sudaro DOM medį. Elementai vadinami mazgais (DOM nodes).

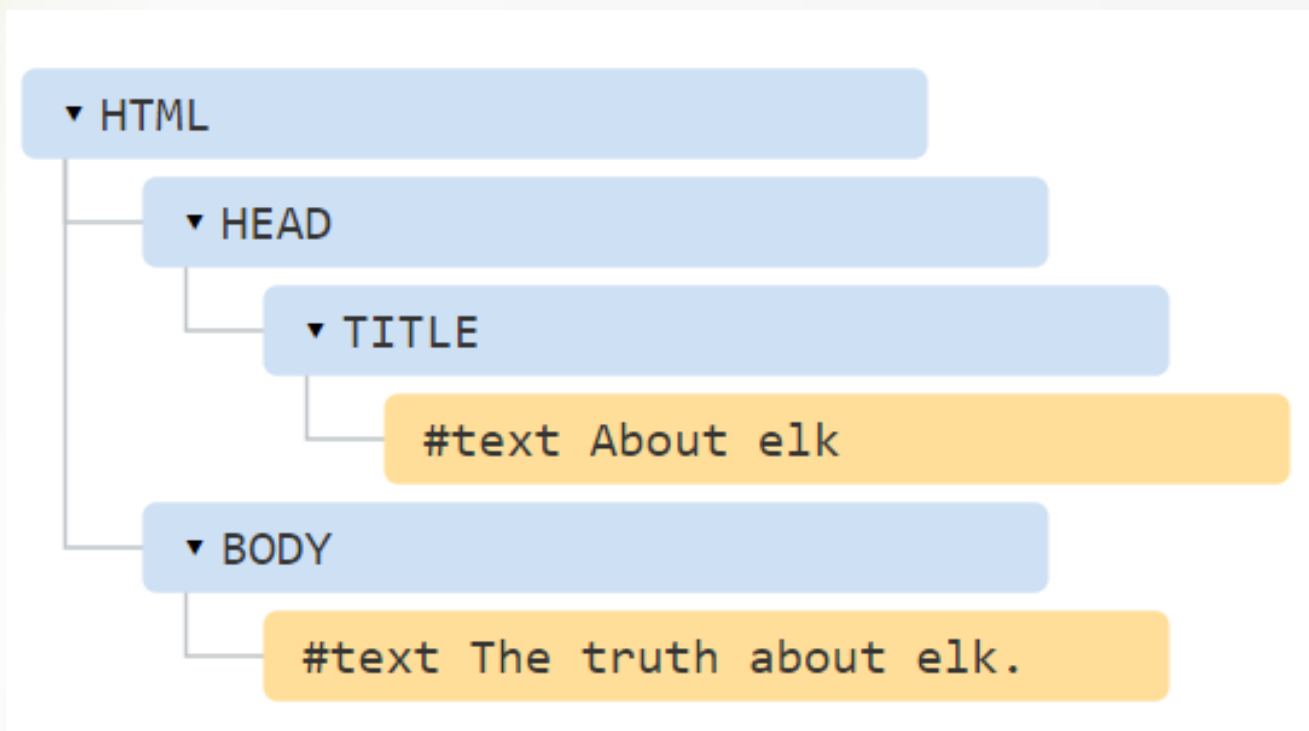


VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

Jei HTML parašysime be tarpų:

```
1 <!DOCTYPE HTML>  
2 <html><head><title>About elk</title></head><body>The truth about elk.</body></html>
```





VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

Dažniausiai dirbame su:

- document – pagrindiniu DOM mazgu.
- element nodes – HTML elementais.
- text nodes – tekstas, enter ir tarpai.
- Kartais dar su komentarais.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

Norint dirbti su HTML elementais, nustatyti ar pakeisti jų parametrus, juos reikia surasti:

HTML elementų radimas pagal jų id:

```
1  <div id="elem">
2    <div id="elem-content">Element</div>
3  </div>
4
5  <script>
6    // get the element
7    let elem = document.getElementById('elem');
8
9    // make its background red
10   elem.style.background = 'red';
11 </script>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM

HTML elementų radimas pagal css selektorių:

```
1  <ul>
2    <li>The</li>
3    <li>test</li>
4  </ul>
5  <ul>
6    <li>has</li>
7    <li>passed</li>
8  </ul>
9  <script>
10   let elements = document.querySelectorAll('ul > li:last-child');
11
12   for (let elem of elements) {
13     alert(elem.innerHTML); // "test", "passed"
14   }
15 </script>
```

Gražina kolekciją visų elementų, atitinkančių duotą css selektorių.

Jei elementų skaičius keičiasi, rezultatas neatsinaujina.

Jei reikia pirmo, atitinkančio css selektorių, naudojame **querySelector.**



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų turinio keitimas

```
<html>
<body>

<p id="p1">Hello World!</p>

<script>
document.getElementById("p1").innerHTML = "New text!";
</script>

</body>
</html>
```

Perrašo visą buvusį elemento turinį.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript can Change HTML</h2>

<p id="p1">Hello World!</p>

<script>
document.getElementById("p1").innerHTML += "New text!";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Papildo elemento turinį.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų atributai

```
const div = document.createElement('div');  
// create a new div referenced in the variable 'div'
```

```
div.setAttribute('id', 'theDiv');  
// if id exists update it to 'theDiv' else create an id  
// with value "theDiv"
```

```
div.getAttribute('id');  
// returns value of specified attribute, in this case  
// "theDiv"
```

```
div.removeAttribute('id');  
// removes specified attribute
```



JavaScript DOM elementų atributų pridėjimas

```
1  <button type="button" id="myBtn">Click Me</button>
2
3  <script>
4      // Selecting the element
5      var btn = document.getElementById("myBtn");
6
7      // Setting new attributes
8      btn.setAttribute("class", "click-btn");
9      btn.setAttribute("disabled", "");
10 </script>
```



JavaScript DOM elementų atributų gavimas

```
1  <a href="https://www.google.com/" target="_blank" id="myLink">Google</a>
2
3  <script>
4      // Selecting the element by ID attribute
5      var link = document.getElementById("myLink");
6
7      // Getting the attributes values
8      var href = link.getAttribute("href");
9      alert(href); // Outputs: https://www.google.com/
10
11     var target = link.getAttribute("target");
12     alert(target); // Outputs: _blank
13 </script>
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų atributų keitimas

```
1  <a href="#" id="myLink">Tutorial Republic</a>
2
3  <script>
4      // Selecting the element
5      var link = document.getElementById("myLink");
6
7      // Changing the href attribute value
8      link.setAttribute("href", "https://www.tutorialrepublic.com");
9  </script>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų atributų šalinimas

```
1  <a href="https://www.google.com/" id="myLink">Google</a>
2
3  <script>
4      // Selecting the element
5      var link = document.getElementById("myLink");
6
7      // Removing the href attribute
8      link.removeAttribute("href");
9  </script>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų atributų reikšmių keitimas

```
<!DOCTYPE html>
<html>
<body>



<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>

</body>
</html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų stiliaus savybių (css) keitimas

```
const div = document.createElement('div');  
// create a new div referenced in the variable 'div'
```

```
div.style.color = 'blue';  
// adds the indicated style rule
```

```
div.style.cssText = 'color: blue; background: white';  
// adds several style rules
```

```
div.setAttribute('style', 'color: blue; background: white');  
// adds several style rules
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų stiliaus savybių (css) keitimas

```
<html>
<body>

<p id="p2">Hello World!</p>

<script>
document.getElementById("p2").style.color = "blue";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Atkreipkite dėmesį, kad JS kalboje **kitaip užrašomos css savybės**: nenaudojant brūkšnio - , o sujungiant žodžius ir parašant juos camelCase.

Pvz. backgroundColor (css background-color). - nenaudojame, nes tai atimties ženklas.

Visas sąrašas: https://www.w3schools.com/jsref/dom_obj_style.asp



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų klasės keitimas - perrašymas

.className parodo, kokia elemento klasė. Galima šią savybę keisti.

```
1 <body class="main page">
2   <script>
3     alert(document.body.className); // main page
4   </script>
5 </body>
```

```
1 <!DOCTYPE html>
2 <html>
3
4 <body>
5
6   <h2>JavaScript can Change HTML</h2>
7
8   <p id="p1" class="hel">Hello World!</p>
9
10  <p>The paragraph above was changed by a script.</p>
11
12  <script>
13    let elem = document.getElementById("p1");
14    console.log(elem.className); //išvedam į konsolę klasės vardą
15    elem.className = "bell"; //pakeičiam, perrašom klasės vardą
16    console.log(elem.className); //išvedam naują klasės vardą
17  </script>
18 </body>
19 </html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų klasės keitimas - pridėjimas ir kt.

classList tai elemento klasių sąrašas (nes elementas gali turėti ne vieną klasę)

```
div.classList.add('new');
```

```
// adds class "new" to your new div
```

```
div.classList.remove('new');
```

```
// remove "new" class from div
```

```
div.classList.toggle('active');
```

```
// if div doesn't have class "active" then add it, or if
```

```
// it does, then remove it
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų klasės keitimas - pridėjimas ir kt.

classList tai elemento klasių sąrašas (nes elementas gali turėti ne vieną klasę)

```
1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6    <h2>JavaScript can Change HTML</h2>
7
8    <p id="p1" class="hell">Hello World!</p>
9
10   <p>The paragraph above was changed by a script.</p>
11
12   <script>
13     let elem = document.getElementById("p1");
14     elem.classList.add("bell"); //prideda naują klasę
15
16     console.log(elem.classList[0]); //parodo elemento klasių sąrašo 1 klasę t.y. hell
17   </script>
18 </body>
19 </html>
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai

JavaScript gali reaguoti į įvairius DOM elementų įvykius ir vykdyti komandas, t.y. atlikti tam tikrus veiksmus.

Įvykiai:

Pelės įvykiai (click, dblclick, mousemove, mouseover, mousewheel);

Lietimo įvykiai (planšetės, telefonai ir kt.) (touchstart, touchmove);

Klaviatūros įvykiai (keypress, keyup);

Formos įvykiai (focus, change, submit);

Window įvykiai (scroll, resize);



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai **onclick**

```
<!DOCTYPE html>
<html>
<body>

<p>Click "Try it" to execute the displayDate() function.</p>

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>
document.getElementById("myBtn").onclick = displayDate;

function displayDate() {
    document.getElementById("demo").innerHTML = Date();
}
</script>

</body>
</html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai **onload**

```
<!DOCTYPE html>
<html>
<body onload="checkCookies()">

<p id="demo"></p>

<script>
function checkCookies() {
    var text = "";
    if (navigator.cookieEnabled == true) {
        text = "Cookies are enabled.";
    } else {
        text = "Cookies are not enabled.";
    }
    document.getElementById("demo").innerHTML = text;
}
</script>

</body>
</html>
```

Objektas
navigator
saugo informaciją
apie naršyklę.

Kodas patikrina, ar
cookies naršyklėje
aktyvuoti.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai **onchange**

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
```

Enter your name: `<input type="text" id="fname" onchange="myFunction()">`

`<p>`When you leave the input field, a function is triggered which transforms the input text to upper case.`</p>`

```
</body>
</html>
```

Atkreipkite dėmesį, kaip paimama formos tekstinio lauko reikšmė **`document.getElementById("fname").value`**



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai

onmouseover, onmouseout

```
<!DOCTYPE html>
<html>
<body>

<div onmouseover="mOver(this)" onmouseout="mOut(this)"
style="background-color:#D94A38;width:120px;height:20px;padding:40px;">
Mouse Over Me</div>

<script>
function mOver(obj) {
  obj.innerHTML = "Thank You"
}

function mOut(obj) {
  obj.innerHTML = "Mouse Over Me"
}
</script>

</body>
</html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykiai **onfocus**

(kai į formos teksto įvedimo lauką padedama pelė)

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction(x) {
  x.style.background = "yellow";
}
</script>
</head>
<body>
```

Enter your name: `<input type="text" onfocus="myFunction(this)">`

`<p>`When the input field gets focus, a function is triggered which changes the background-color.`</p>`

```
</body>
</html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM įvykių klausymas

Bet kuriam DOM elementui galima parašyti įvykių klausymo metodą.

Tai šiuolaikinis būdas dirbti su įvykiais.

```
<button id="myBtn">Try it</button>

<script>
document.getElementById("myBtn").addEventListener("click", myFunction);

function myFunction() {
  alert ("Hello World!");
}
</script>
```

Įvykis gali būti bet kuris iš sąrašo:

https://www.w3schools.com/jsref/dom_obj_event.asp



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų kūrimas

1. Teksto kūrimas:

```
var node = document.createTextNode("This is new.");
```

2. HTML elemento kūrimas:

```
var para = document.createElement("p");
```

3. Teksto ir elemento sujungimas

(analogiškai vyksta ir dviejų elementų sujungimas, naujas elementas pridedamas tėvinio elemento gale)

```
para.appendChild(node);
```



prie kurio elemento



kurį elementą jungiame



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų kūrimas

```
<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>  
  
<script>  
var para = document.createElement("p");  
var node = document.createTextNode("This is new.");  
para.appendChild(node);  
var element = document.getElementById("div1");  
element.appendChild(para);  
</script>
```

!!! Naujai sukurtas elementas nematomas puslapyje, kol jis nepridedamas prie jau esamo elemento (appendChild)

This is a paragraph.

This is another paragraph.

This is new.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų kūrimas

Įterpiant naują elementą, galima nurodyti jo vietą:

```
element.insertBefore(para, child);
```

↓
Į kurį
elementą
įterpiam

↓
Kurį
elementą
įterpiam

↓
Prieš kurį
elementą
įterpiam



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų kūrimas

```
<div id="div1">  
<p id="p1">This is a paragraph.</p>  
<p id="p2">This is another paragraph.</p>  
</div>
```

```
<script>  
var para = document.createElement("p");  
var node = document.createTextNode("This is new.");  
para.appendChild(node);
```

```
var element = document.getElementById("div1");  
var child = document.getElementById("p1");  
element.insertBefore(para, child);  
</script>
```



VILNIAUS TECHNOLOGIJŲ
MOKYTOJŲ CENTRAS

JavaScript DOM elementų šalinimas

```
<div>
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
</div>

<button onclick="myFunction()">Remove Element</button>

<script>
function myFunction() {
    var elmnt = document.getElementById("p1");
    elmnt.remove();
}
</script>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM elementų pakeitimas

```
<div id="div1">
  <p id="p1">This is a paragraph.</p>
  <p id="p2">This is another paragraph.</p>
</div>

<script>
var parent = document.getElementById("div1");
var child = document.getElementById("p1");
var para = document.createElement("p");
var node = document.createTextNode("This is new.");
para.appendChild(node);
parent.replaceChild(para, child);
</script>
```

Kuriame elem. vyks
pokyčiai

Kuo keisim
(naujas elem.)

Kurį elem. Keisim
(senas elem.)



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript DOM pavyzdys

```
<!-- your html file: -->
<body>
  <h1>
    THE TITLE OF YOUR WEBPAGE
  </h1>
  <div id="container"></div>
</body>
```

```
// your javascript file
const container = document.querySelector('#container');

const content = document.createElement('div');
content.classList.add('content');
content.textContent = 'This is the glorious text-content!';

container.appendChild(content);
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

```
<form action="/signup" method="post" id="signup">  
</form>
```

Formos elementas **HTMLFormElement** turi du svarbiausius atributus:

action – nurodo URL, kuris bus užkrautas kai patvirtinsime formą, t.y. nuspausime mygtuką submit, tai gali būti failas su skriptu formos duomenims apdoroti.

method – nurodo HTTP metodą, kuriuo bus patvirtinta forma (GET, POST...)



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

```
<form action ="/signup" method = "post" id="signup" name="signup"></form>
```

Formą galime pasiekti **per ID ir per vardą**:

```
const form = document.getElementById('signup');
```

```
const form = document.forms['login'];
```

document.forms grąžina kolekciją visų dokumente esančių formų



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Formos laukų išvalymas (reset):

```
<form id="form">  
  First name: <input type="text" name="firstname"><br>  
  Last name: <input type="text" name="lastname"><br><br>  
  <input type="button" onclick="resetForm()" value="Reset form">  
</form>
```

```
function resetForm() {  
  document.getElementById("form").reset();  
}
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Kai norime ką nors daryti pvz. atlikti formos duomenų validaciją po to, kai nuspaudžiame mygtuką submit, reikia pridėti formai įvykių klausymą ir kai norimas įvykis įvyksta iškviečiama funkcija.

```
const form = document.getElementById('signup');  
  
form.addEventListener('submit', (event) => {  
    // handle the form data  
});
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Kad paspaudus submit mygtuką puslapis nepersikrautų, reikia pridėti **event.preventDefault()** metodą.

```
form.addEventListener('submit', (event) => {  
    // stop form submission  
    event.preventDefault();  
});
```



VILNIAUS TECHNOLOGIJŲ
MOKYTOJŲ CENTRAS

JavaScript darbas su formomis

Formos elementų pasiekimas galimas su bet kuriuo iš DOM metodų:

getElementsByName(),
getElementById(),
querySelector()

arba:

```
const form = document.getElementById('signup');
```

```
form.elements[1]; // by index  
form.elements['email']; // by name  
form.elements['email']; // by id
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Dar vienas būdas, kaip paimti į formos laukus įrašytus duomenis:

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}
</script>
</head>
<body>

<form name="myForm" onsubmit="return validateForm()">
    Name: <input type="text" name="fname">
    <input type="submit" value="Submit">
</form>

</body>
</html>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Formos elementų reikšmės pasiekiamos su savybe **value**:

```
const form = document.getElementById('signup');  
const name = form.elements['name'];  
const email = form.elements['email'];  
  
// getting the element's value  
let fullName = name.value;  
let emailAddress = email.value;
```

arba

```
var age = document.getElementById("age").value;
```

Formų elementų reikšmės visada yra string tipo, norint atlikti skaičiavimus, reikia versti skaičiumi.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Duomenų paėmimas iš formos laukų text, number ir date:

```
1  const form = document.querySelector("#register");
2
3  form.addEventListener('submit', (e) => {
4      e.preventDefault();
5      let name = form.elements['fname'].value;
6      let lastName = form.elements['lname'].value;
7      let age = form.elements['age'].valueAsNumber;
8      let birthday = form.elements['birthday'].value;
9
10     let user = { name, lastName, age, birthday };
11     console.log(user);
12 })
13
```

```
<form id="register">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>

  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br>

  <label for="quantity">Age (between 1 and 100):</label>
  <input type="number" id="age" name="age" min="1" max="100" value="20"><br>

  <label for="start">Birthday:</label>
  <input type="date" id="birthday" name="birthday" value="2000-07-22"><br>

  <input type="submit" value="Submit">
</form>
```

Galima datos reikšmę iš karto paimti kaip datos objektą, naudojant ne value, bet valueAsDate.



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Duomenų paėmimas iš formos laukų checkbox:

```
1  const form = document.querySelector("#register");
2  const skills = document.querySelectorAll('input[name = "fav_language"]');
3
4  form.addEventListener('submit', (e) => {
5    e.preventDefault();
6    let skillsarray = [];
7
8    skills.forEach((skill) => {
9      if (skill.checked) {
10        skillsarray.push(skill.value);
11      }
12    })
13    console.log(skillsarray);
14  })
```

```
<form id="register">
  <p>Skills:</p>
  <input type="checkbox" id="html" name="fav_language" value="HTML"
checked>
  <label for="html">HTML</label><br>
  <input type="checkbox" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="checkbox" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label><br>

  <input type="submit" value="Submit">
</form>
```




VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Duomenų paėmimas iš formos laukų radio:

JS letterGrade.js > ...

```
1 let form = document.getElementById("login");
2 let radioButtons = document.querySelectorAll('input[name="fav_language"]');
3
4 form.addEventListener("submit", (e) => {
5   // atspausdinam visus radio mygtukus, kad pamatyti savybę checked
6   console.log(radioButtons);
7
8   e.preventDefault();
9
10  radioButtons.forEach((radio) => {
11    if (radio.checked) {
12      console.log(radio.value);
13    }
14  });
15 });
```

```
<form id="login">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
  <input type="submit" value="Submit">
</form>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Duomenų paėmimas iš formos lauko select:

```
<label>
  Choose an ice cream flavor:
  <select class="ice-cream" name="ice-cream">
    <option value="">Select One ...</option>
    <option value="chocolate">Chocolate</option>
    <option value="sardine">Sardine</option>
    <option value="vanilla">Vanilla</option>
  </select>
</label>

<div class="result"></div>
```

```
const selectElement = document.querySelector('.ice-cream');

selectElement.addEventListener('change', (event) => {
  const result = document.querySelector('.result');
  result.textContent = `You like ${event.target.value}`;
});
```

Choose an ice cream flavor:

You like chocolate



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript darbas su formomis

Duomenų paėmimas iš formos lauko select:

```
<body>

<p>Select a new car from the list.</p>

<select id="mySelect" onchange="myFunction()">
  <option value="Audi">Audi</option>
  <option value="BMW">BMW</option>
  <option value="Mercedes">Mercedes</option>
  <option value="Volvo">Volvo</option>
</select>

<p>When you select a new car, a function is triggered which
outputs the value of the selected car.</p>

<p id="demo"></p>

<script>
function myFunction() {
  var x = document.getElementById("mySelect").value;
  document.getElementById("demo").innerHTML = "You selected: " +
x;
}
</script>

</body>
```



VILNIAUS TECHNOLOGIJŲ
MOKYMO CENTRAS

JavaScript formos validacija

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>JavaScript Form Demo</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <link rel="stylesheet" href="css/style.css" />
7  </head>
8  <body>
9      <div class="container">
10         <form action="signup.html" method="post" id="signup">
11             <h1>Sign Up</h1>
12             <div class="field">
13                 <label for="name">Name:</label>
14                 <input type="text" id="name" name="name" placeholder="Enter your fullname" />
15                 <small></small>
16             </div>
17             <div class="field">
18                 <label for="email">Email:</label>
19                 <input type="text" id="email" name="email" placeholder="Enter your email address" />
20                 <small></small>
21             </div>
22             <div class="field">
23                 <button type="submit" class="full">Subscribe</button>
24             </div>
25         </form>
26     </div>
27     <script src="js/app.js"></script>
28 </body>
29 </html>
```

<https://www.javascripttutorial.net/sample/dom/form/css/style.css>

<https://www.javascripttutorial.net/sample/dom/form/js/app.js>

Paaiškinimai: <https://www.javascripttutorial.net/javascript-dom/javascript-form/>