



VILNIUSCODINGSCHOOL

JS



JS



JS įvadas. Kas yra JS?

- WEB programavimo kalba
- Kliento pusės programavimo kalba
- Suteikia puslapiui interaktyvumo, gali paimti duomenis, atlieka skaičiavimus
- JS rašomas atskirame faile su plėtiniu .js
 - `<script type="text/javascript" src="scripts/manokodas.js"></script>`
- JS rašomas html dokumente
 - `<script type="text/javascript"><!-- kodas --></script>`
- JS rašomas elemente
 - `<input type="button" onclick="alert('ups');" />`
- Rašome `<head>` arba pabaigoje prieš `<body>` elementą.
- Sveikas pasauli



JS įvadas. Kintamųjų tipai

- Tipai suskirstyti į:
 - Objektas
 - Primityvi reikšmė
 - undefined (neapibrėžta)
 - Boolean: true/false
 - Number: 1,2,3, 1.5 ...
 - String: 'tekstas' arba "tekstas" arba "tekste yra kitas 'tekstas'"
 - Funkcija

```
function Add(x, y){  
  • return x+y;  
  }  
}
```

```
var addFunction = function(x, y){  
  return x + y;  
};
```

```
Add(5, 10);
```

```
addFunction(5, 10);
```

JS



JS įvadas. Kintamųjų rašymas

- Pradedami žodeliu var
- Kintamųjų pavadinimai susideda iš raidžių ir skaičių, negali prasidėti skaičiumi, pavadinime negali būti matematinių ar loginių operacijų. Teisingi pvz. x1, y2, gift4you. Taip pat gali prasidėti kintamasis simboliais _ ir \$.
- Kintamųjų vardai negali būti JS kalboje naudojami pavadinimai pvz. class, function, char...
- Kintamieji yra case sensitive. Pvz. kintamieji x1 ir X1 yra traktuojami kaip skirtingi.
- Kintamojo vardas turi būti paaiškinantis jo reikšmę, kad ir koks ilgas būtų
 - var lastAccessedDate;
 - var currentVehicle;
 - var vehicleMakeWasChanged;

JS



Keletas standartinių funkcijų

- `alert`
- `prompt`
- `confirm`
- `isNaN` (`if(isNaN(amzius))`)
- `Number` `if(Number(skaicius) == 5) Number(skaicius)`
- `String` `String(raide)`
- `Console.log`

JS



Sąlygos sakiniai

- **If** (sąlyga){ veiksmas jei sąlyga tenkinama }
- **If** (sąlyga) { veiksmas jei sąlyga tenkinama } **else** { veiksmas jei sąlyga netenkinama }
- **If** (sąlyga) { veiksmas jei sąlyga tenkinama } **else if** (kita sąlyga){ veiksmas jei sąlyga tenkinama }
- **Switch**

JS



While ciklas

- `var x = "";`
- `while(x != 'qwerty123') {`
- - `x = prompt('iveskite slaptazodi', "");`
- `}`
- `alert("Jūs patekote į sistemą");`



do ciklas

- `var retries = 0;`
- `do{`
- `retries++;`
- `var slaptazodis = prompt('Iveskite slaptazodi. Tai yra jūsų ' + retries + ' bandymas.', "");`
- `}`
- `while(slaptazodis != 'qwerty123' && retries < 3);`
- `}`
- `if(retries < 3){`
- `alert('Jūs patekote į sistemą.');`
- `}`
- `else{`
- `alert('Jūs užblokuotas amžiams!');`
- `}`

JS

for ciklas

- `for (var counter = 0; counter < 10; counter++){`
 - `alert('The counter is now set to ' + counter);`
- `}`

JS



Išėjimas iš ciklo

Klaidos atveju arba atitikus tam tikrą sąlygą.

- Break;
- Continue;

**Deklaravimas ir priskyrimas:**

```
var picosSudetis = new Array();  
picosSudetis[0] = 'sūris';  
picosSudetis[1] = 'svogūnas';  
picosSudetis[2] = 'kumpis';
```

```
var picosSudetis = new Array('sūris',  
'svogūnas', 'kumpis');
```

```
var picosSudetis = ['sūris',  
'svogūnas', 'kumpis'];
```

Skaitymas:

```
var treciojiSudetine = picosSudetis[2];
```

Keitimas:

```
picosSudetis[2] = 'žuvis';
```

JS



Masyvo metodai

https://www.w3schools.com/js/js_arrays.asp

- **concat** – sujungia du ar daugiau masyvų ir gražina naują masyvą;
- **indexOf** – suranda masyve elementą ir gražina jo indeksą;
- **join** – sukuria eilutę iš masyvo elementų, elementai eilutėje atskirti kableliai pagal nutylėjimą
- **lastIndexOf** – masyvas gali turėti kelis vienodus elementus, šis metodas suranda paskutinį ir gražina indeksą;
- **pop** – pašalina ir gražina paskutinį masyvo elementą;
- **push** – prideda naują elementą ir gražina naują masyvo ilgį;
- **reverse** – sukeičia masyvo elementus vietomis (reversuoja). Priekis tampa galu.
- **shift** – pašalina ir gražina pirmą masyvo elementą;
- **slice** – gražina naują masyvą iš egzistuojančio. Turi 2 parametrus, pirmas – naujo masyvo pradžia, antras – naujo masyvo pabaiga. Pabaigą žymintis indeksas neįtraukiamas;
- **sort** – išrikiuoja pagal abėcėlę;
- **splice** – prideda ir pašalina masyvo elementus. Gražina pašalintus. Pirmas parametras nurodo nuo kurio elemento, antras kiek elementų, trečias – kokį elementą įterpti;
- **toString** – sujungia į eilutę, pagal nutylėjimą atskirtus kableliais
- **unshift** – prideda naują elementą į pradžią ir gražina naują masyvo ilgį

JS



Priėjimas prie DOM elementų

- **document**
 - **getElementById** – grąžina nuorodą į elementą pagal nurydtą id;
 - **getElementsByTagName** – grąžina masyvą elementų kurių elemento pavadinimas atitinka metodo parametro reikšmę;
 - **getElementsByClass** – grąžina elementų masyvų kurių klasė atitinka metodo parametro reikšmę;
 - **querySelector** – grąžina pirmą pagal užklausą (užklausai naudojama css sintaksė)
 - **querySelectorAll** – grąžina visus elementus pagal užklausą (užklausai naudojama css sintaksė)

JS



- **Window event reference - lango**
- **Form event reference - formos**
- **Keyboard event reference - klaviatūros**
- **Mouse event reference - pelės**
- **Media event reference - media**

https://www.w3schools.com/tags/ref_eventattributes.asp

JS



Objektai

```
var car1 = {  
  year: 2000,  
  make: 'Ford',  
  model: 'Fusion',  
  getInfo: function () {  
    return 'Vehicle: ' + this.year + ' ' + this.make + ' ' + this.model;  
  }  
};
```

https://www.w3schools.com/tags/ref_eventattributes.asp

JS



- <https://jquery.com/>
- **\$('#vardas').veiksmas** – selektorius lygiai toks pat kaip css
- palyginimui **document.getElementById('vardas').addEventListener('click', zinute)**
- su
- **\$('#vardas').on('click', zinute)**
- Reikšmių priskyrimas elementams
- Animation
- Dinaminis elemento kūrimas

JS



Klausimai

