# script src

* gdzie należy umieszczać <script>

# Problem z task runnerem

* Node-sass

# Funkcja w addEventListener

buttonScissors.addEventListener ('click', *function*() {

    playGame (3)

});

# Argument funkcji

*function* printMessage(*msg*){

*let* div = document.createElement('div');

    div.innerHTML = *msg*;

    document.getElementById('messages').appendChild(div);

}

# [Metody obiektu](https://kodilla.com/pl/docs-js-es6#/objects?id=metody-obiektu)

Do tej pory mówiliśmy o właściwościach obiektu. Przyjęło się, że właściwością nazywamy tylko taką parę klucz-wartość, w której wartość nie jest funkcją.

Dla odmiany, jeśli wartość jest funkcją, to taką parę klucz-funkcja nazywamy metodą. Przykładem metody może być np.

const calculate = {

add: function(a, b){

return a + b;

}

};

const sumOfNumbers = calculate.add(3, 7);

console.log('sumOfNumbers:', sumOfNumbers); // sumOfNumbers: 10

Metodami są również np. document.querySelector, czy Math.random – są one wbudowane w silnik JS przeglądarki i nie musimy ich tworzyć, ale również je nazywamy metodami. Możesz założyć, że metodą jest każda funkcja, której nazwę piszemy po kropce

# [Obiekty wielowymiarowe](https://kodilla.com/pl/docs-js-es6#/objects?id=obiekty-wielowymiarowe)

*const* objTwo = {

    'id1' : 'val1', 'id2' : {

        'name' : 'eve',

        'surname' : 'smith',

    },

    'id3' : [

        't1',

        {

            'o1' : 'v1',

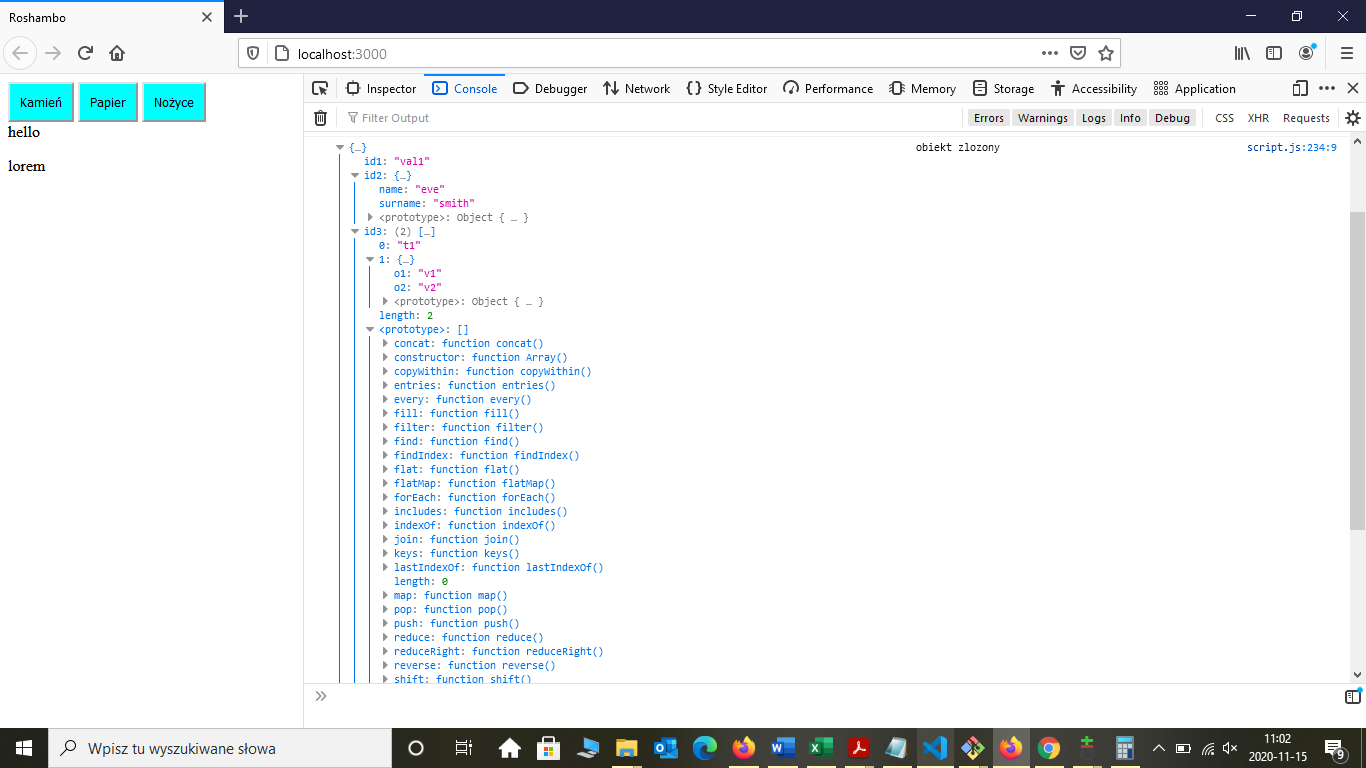
            'o2' : 'v2'

        },

    ],

};

console.log(objTwo, ' obiekt zlozony');



# swith & if

# [Pętla for...in](https://kodilla.com/pl/docs-js-es6#/loops?id=p%c4%99tla-forin)

Pętla for...in umożliwia wykonywanie operacji dla każdego elementu w obiekcie.

**Przykład**

const person = {

name: 'John',

surname: 'Doe',

age: 35

};

for( let key in person ){

console.log('Value at key "' + key + '" in person: ' + person[key]);

}

const elementsList = ['lorem', 123, 3+'c', '999'];

for( let i = 0; i < elementsList.length; i++ ){

if(typeof(elementsList[i]) == 'string'){

console.log('Text from an element from elementsList at index ' + i + ' is: ' + elementsList[i]);

}

}

# Zmienne lokalne i globalne