

# Podstawy JavaScript

## 1. Wyprowadzanie / wprowadzanie danych

1. Utwórz dokument HTML zawierający następujący kod:

```
<html>
  <body>
    <div>Treść dokumentu HTML przed skrypcem</div>
    <script>
      console.log('Tekst 1');
      window.alert('Tekst 2');
      document.write('Tekst 3');
    </script>
    <div>Treść dokumentu HTML po skrypcie</div>
  </body>
</html>
```

2. Uruchom przeglądarkę WWW, a następnie [otwórz jej konsolę](#)
3. Załaduj powyższy dokument w bieżącej zakładce przeglądarki WWW
4. Spróbuj zlokalizować miejsce pojawiania się tekstów: Tekst 1, Tekst 2 oraz Tekst 3
5. Utwórz alternatywną wersję dokumentu HTML, w której to skrypt będzie wywoływany po zakończeniu renderowania dokumentu HTML, a nie w trakcie (renderowania) — utwórz, a następnie załaduj, następujący dokument HTML:

```
<html>
  <body onload="
    console.log('Tekst 1');
    window.alert('Tekst 2');
    document.write('Tekst 3');
  ">
    <div>Treść dokumentu HTML</div>
    <div>Treść dokumentu HTML</div>
    <div>Treść dokumentu HTML</div>
  </body>
</html>
```

6. Jak myślisz, dlaczego w tej wersji, treść dokumentu HTML nie jest widoczna w przeglądarce — usuń linię, która jest tego przyczyną
7. Dopisz w ciele dokumentu:

```
<script>
  window.prompt("Tekst1", "Tekst2");
</script>
```

8. Zbadaj, jakie znaczenie mają poszczególne argumenty metody `window.prompt()` i czy są one obowiązkowe
9. Zbadaj, co jest zwracane (wartość, [typ danych](#)) w przypadku:
- Wprowadzeniu wartości będącej liczbą i naciśnięciu klawisza 'Enter' lub przycisku 'OK'
  - Wprowadzeniu wartości będącej napisem i naciśnięciu klawisza 'Enter' lub przycisku 'OK'
  - Niewprowadzeniu wartości i naciśnięciu powyższego klawisza / przycisku
  - Wprowadzeniu wartości i naciśnięciu przycisku 'Anuluj'
10. Dopisz w obrębie elementu "body"
- ```
<form>
  <input name="pole_tekstowe" type="text">
  <input name="pole_liczbowe" type="number">
  <input type="button" value="Wypisz">
</form>
```
11. Korzystając z odpowiedniego [obserwatora zdarzeń](#) postaci '`onNazwaZdarzenia`', kolekcji [DOM 0](#), takich jak: `document.forms[]` oraz `document.forms[].elements[]`, spowoduj, aby po kliknięciu przycisku "Wypisz" wyświetliła się zawartość wprowadzona w polu tekstowym oraz liczbowym formularza
12. Zbadaj, co jest zwracane (wartość, typ danych) w przypadku:
- Wprowadzeniu wartości będącej liczbą i naciśnięciu powyższego przycisku
  - Wprowadzeniu wartości będącej napisem i naciśnięciu w/w przycisku
  - Niewprowadzeniu wartości i naciśnięciu przycisku "Wypisz"
13. Przenieś całą zawartość (kod JS) elementu "script" do osobnego pliku "script.js" (utwórz zewnętrzny skrypt JS), a następnie załaduj ten skrypt z poziomu dokumentu HTML

## 2. Testy

1. Utwórz dokument HTML o nazwie 'test.html' i poniższej zawartości

```
<!-- Źródło / Source: https://mochajs.org/#running-mocha-in-the-browser -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      Mocha tests
    </title>
    <meta name="viewport"
      content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
      href="https://unpkg.com/mocha/mocha.css">
  </head>
  <body>
    <div id="mocha">
    </div>
    <script src="https://unpkg.com/chai/chai.js"></script>
    <script src="https://unpkg.com/mocha/mocha.js"></script>
    <script class="mocha-init">
```

```

    mocha.setup('bdd');
    mocha.checkLeaks();
  </script>
  <script src="script.js"></script>
  <script class="mocha-exec">

    mocha.run();
  </script>
</body>
</html>

```

## 2. Zastąp zawartość pliku 'script.js' następującą:

```

var expect = chai.expect;

function sum(x,y) {
  return x+y;
}

describe('The sum() function', function() {
  it('Returns 4 for 2+2', function() {
    expect(sum(2,2)).to.equal(4);
  });
  it('Returns 0 for -2+2', function() {
    expect(sum(-2,2)).to.equal(0);
  });
});

```

## 3. Otwórz dokument 'test.html' w przeglądarce WWW

## 4. Przeczytaj [opis](#) tworzenia testów w oparciu o [Mocha](#) oraz [Chai](#)

## 5. Włącz [tryb ścisły](#)

## 6. Napisz skrypt, który za pomocą metody `window.prompt()` wczytuje dane (napis), przekazuje je do poniższych funkcji, a następnie wypisuje na bieżąco (po każdym wczytaniu danych):

```
\twynik_działania_funkcji_cyfry\twynik_działania_funkcji_litery\twynik_działania_funkcji_suma
```

Przykład działania:

```

111
 3 0 111 ← Suma = 111
11aa
 2 2 122 ← Suma = 111 + 11
b3345a
15 2 122 ← Suma = 111 + 11 + 0

```

Uwagi i informacje:

- o Funkcja `cyfry(napis)`, dla wczytanego napisu, oblicza, a następnie zwraca **sumę** zawartych w nim cyfr
- o Funkcja `litery(napis)`, oblicza, a następnie zwraca **ilość** zawartych w nim liter
- o Funkcja `suma(napis)`, oblicza, na bieżąco, **sumę wszystkich** wczytanych liczb, o ile napis wygląda jak liczba, tzn. rozpoczyna się od ciągu cyfr (patrz linia 3 przykładu) lub zawiera same cyfry (patrz linia 1 przykładu)

- Wczytywanie danych ma się odbywać do momentu naciśnięcia przycisku 'Anuluj'
7. Zaimplementuj test, w oparciu o Mocha, weryfikujący poprawność działania powyższych funkcji, dla napisów zawierających:
- Same cyfry
  - Same litery
  - Litery, a po nich cyfry
  - Cyfry, a po nich litery
  - Pusty napis

### 3. Tworzenie dynamicznych grafik

1. Przeczytaj [HTML5 Canvas – kurs od podstaw](#) lub [Kurs HTML5 – Canvas – Wprowadzenie](#)
2. Utwórz dokument HTML o nazwie 'rysunek.html' i poniższej zawartości

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <meta charset="UTF-8">
    <title>
      Page title
    </title>
    <script>
      var canvas = document.getElementById('canvas');
      var ctx = canvas.getContext('2d');
      ctx.fillText("Hello World", 10, 50); //Wykreślenie podanego tekstu na płótnie / Drawing given text on canvas
    </script>
  </head>
  <body>
    <main>
      <h1>
        Płótno
      </h1>

      <h1>
        Canvas
      </h1>

      <canvas id="canvas"
        width="200"
        height="300"
        style="border:1px solid #000000;">
        Wygląda na to, że twoja przeglądarka nie obsługuje elementu "canvas" / It looks like your browser does not support the "canvas" element
      </canvas>
    </main>
  </body>
</html>
```

3. Dlaczego na powierzchni płótna nie pojawił się napis "Hello World"? — zobacz jaki komunikat wyświetla się w konsoli przeglądarki WWW (Ctrl+Shift+I); spróbuj wprowadzić odpowiednie modyfikacje, tak aby powyższe instrukcje zadziałały

4. Opracuj, a następnie narysuj (w elemencie 'canvas') logo dla aplikacji, która jest treścią ćwiczeń — "biblioteka" / "wypożyczalnia" / "dealer samochodowy" — logo powinno się składać, z co najmniej, trzech figur geometrycznych

## 4. Zadania

Dokument HTML zawiera formularz. Napisz skrypt oraz towarzyszący mu test Mocha. Założenia dla skryptu:

- Po kliknięciu [przycisku formularza](#) skrypt wczytuje dane zawarte w obszarze tekstowym ([textarea](#)), korzystając z obiektów/kolekcji 'document.forms' oraz 'elements'
- Używa kolekcji [indeksowanych](#) lub kluczowanych ([Map / Set](#), [SessionStorage / localStorage](#)) przechowujących dane typów [prostych](#) lub [obiektowych](#)
- Realizuje funkcjonalność podaną na **początku ćwiczeń**
- Wypisuje wynik w konsoli lub w elemencie 'canvas'