

JavaScript: Zmienne i operacje

dimon.work/kurs.html

1. Nasze plany dotyczące JavaScript

Nasze plany dotyczące JavaScript

Zmienne i operacje Rozgałęzienia (operatory warunkowe) Cykle / tablice (struktury danych) Funkcje Obiekty

Podstawy programowania



Zarządzanie dokumentami (DOM) Model zdarzeń w JavaScript Tworzenie interaktywnych widżetów.

Zastosowanie

2. Zmienne

Zadanie każdego programu - przetwarzanie danych

```
let price = 799;
let quantity = 10;
let totalCost = price * quantity;

console.log('Total Cost:', totalCost, 'UAH.');

10
11
12
```

Zmienna musi zostać **zadeklarowana**, zanim będzie można jej użyć. Powiedz przeglądarce, że chcemy utworzyć kolejne "pudełko" na wartości i nadaj mu nazwę. Zmienne są deklarowane za pomocą słowa kluczowego **let**.

JavaScript używa zmiennych do przechowywania danych (informacji). **Zmienne** można traktować jako "pudełko", które ma nazwę i przechowuje wartość. Wartością może być liczba, ciąg znaków lub inne typy danych obsługiwane przez JavaScript.

$$x=1$$
Variable Value

3. Wprowadzenie/odzyskiwanie danych

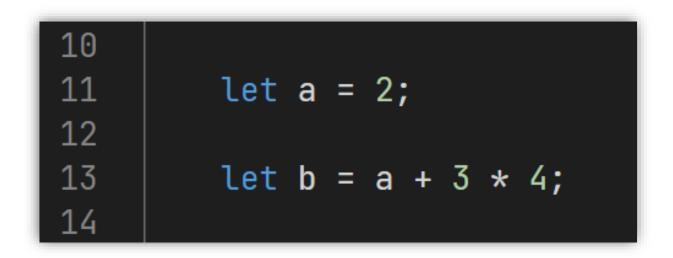
Wprowadzenie/odzyskiwanie danych

```
//Варианты ввода данных (без разметки)
       let userName = prompt('Please enter user name');
 4
 5
       let canDrive = confirm('are you can drive?');
 6
       //Варианты вывода данных (без разметки)
       alert('Hello');
 8
 9
       console.log('User name:', userName);
10
11
12
       document.write(`<h1>Hello ${userName}</h1>`);
13
14
```

Głównym sposobem wprowadzania danych do aplikacji **JavaScript** są formularze i elementy wejściowe, a dane wyjściowe znajdują się w znacznikach dokumentu. Jednocześnie do celów **szkoleniowych** (i debugowania) przydatne będą następujące metody zarządzania danymi.

4. Operacje, operatorzy, operandy i wyrażenia

Operatory przypisania





Operator przypisania służy do określenia, co ma zostać zapisane w zmiennej.

Operator przypisania pobiera to, co znajduje się po jego prawej stronie i zapisuje to w zmiennej, której nazwa znajduje się po jego lewej stronie. Jeśli po prawej stronie znajduje się wyrażenie, zostanie ono obliczone jako pierwsze, a zmienna będzie zawierać wynik obliczenia wyrażenia.

Wyrażenia

```
2
       let a = 3;
 3
 4
       let b = a + 6;
 5
 6
       let c = b + 1;
       c = a + b * c + 7;
 8
 9
       console.log('In Variable C:', c);
10
11
12
```

Po prawej stronie operatora przypisania może znajdować się konkretna wartość, a także wyrażenie - formuła, po obliczeniu której komputer otrzyma wynik, który zostanie zapisany do zmiennej, której nazwa znajduje się po lewej stronie znaku przypisania. Wyrażenie może zawierać zarówno konkretne wartości (stałe), jak i inne **zmienne**.

Operatory, operandy i operacje...

Operatory (których jest całkiem sporo) służą do wykonywania działań (**operacji**) na zmiennych (lub wartościach). Każdy zna niektóre z nich, takie jak operatory arytmetyczne.

Operatory mają **priorytety**, nie jest łatwo zapamiętać, który priorytet jest wyższy, a który niższy. Dlatego w przypadku wątpliwości, która operacja będzie pierwsza, a która druga, należy użyć nawiasów. Zasada ich użycia jest taka sama jak w matematyce - nawiasy zwiększają priorytet zapisanej w nich operacji.

Operatorzy i ich priorytety

Level	Operators
1	0 🛮 -
2	1 ~ - ++
3	* / %
4	+ -
2 3 4 5 6 7	<< >> >>>
6	< <= > >=
7	== !=
8	&
9	Λ
10	1
11	&&
12	II
13	?:
14	= += -= *= /= %= <<=
	>>= >>= &= ^= =

Operatory mają priorytety, nie jest łatwo zapamiętać, który priorytet jest wyższy, a który niższy. Dlatego w przypadku wątpliwości, która operacja będzie pierwsza, a która druga, należy użyć nawiasów. Zasada ich użycia jest taka sama jak w matematyce - nawiasy zwiększają priorytet zapisanej w nich operacji.

5. Typy danych (string & number)

Typy danych (zmiennych)

```
1
2    let a = 7;
3
4    let b = 8;
5
6    let c = a + b;
7
8    let d = a * b;
9
10    console.log('Value in C:', c); //15
11
12    console.log('Value in D:', d); //56
13
```

```
1
2    let a = '7';
3
4    let b = '8';
5
6    let c = a + b;
7
8    let d = a * b;
9
10    console.log('Value in C:', c); //'78'
11
12    console.log('Value in D:', d); //56
13
```

```
let a = '7';

let b = 8;

let c = prompt('Enter Some Number');

console.log(typeof(a), typeof(b), typeof(c)); //string, number, string
```

JavaScript nie ma sztywnego typowania danych, w którym typ zmiennej jest określany w momencie jej zadeklarowania. W JavaScript typ zmiennej jest określany po przypisaniu do niej wartości. Może się on **zmieniać** przy każdym nowym przypisaniu. Możemy sprawdzić typ **zmiennej** za pomocą funkcji typeof(...).

Typy danych (zmiennych)

Typ danych - oznaczenie dla komputera, jak traktować te lub inne dane i jakie operacje można z nimi wykonać.

Typ definiuje **możliwe wartości** i ich "**sens**", a także **operacje**, które można wykonać na tych wartościach.

undefined, number, string, boolean, function, object, symbol, bigint

6. Konwersja typów

```
3
4 let a = '7.99';
5
6 console.log(a, typeof(a)); // '7.99' string;
7
8 a = +a; // оператор + в унарном виде;
9
10 console.log(a, typeof(a)); // 7.99 number;
11
```

Konwersja typów

Funkcje parseInt()/parseFloat() umożliwiają konwersję typu zmiennej z łańcuchowego na numeryczny (jest to możliwe, jeśli łańcuch faktycznie zawiera jakiekolwiek znaki numeryczne, w przeciwnym razie wynikiem będzie wartość NaN). parseInt() - działa z liczbami całkowitymi, parseFloat() obsługuje liczby ułamkowe.

Operator + w postaci jednoargumentowej może być również używany do konwersji typów, ale w przeciwieństwie do funkcji **parse***, wszelkie znaki nienumeryczne w ciągu znaków będą skutkować wartością **NaN**.

7. NaN (Not a Number)

NaN – Not a Number

```
let a = 'hello';
 5
 6
       let b = 7;
 8
       let c = a * b; //NaN - в результате
       выполнения арефетической операции среди
       операндов оказалось значение, которое
       невозможно преобразовать к числу;
 9
10
       let d = parseFloat(a); //NaN - невозможно
       даже часть строки преобразовать к числу;
```

NaN (Not a Number) to specialna wartość typu number, która wskazuje, że jeden z operandów nie mógł zostać pomyślnie przekonwertowany na liczbę w wyniku operacji arytmetycznej (lub jawnej konwersji na liczbę). Ponieważ **JavaScript** nie jest językiem typowanym, nie mogą w nim występować błędy konwersji typów, dlatego istnieje taka konstrukcja jak NaN.

8. Math

Objekt Math

```
4
       let a = 7.6374;
 5
 6
       let b = Math.round(a);
 8
       let c = Math.floor(a);
 9
10
       let d = Math.ceil(a);
11
       console.log(a, b, c, d);
12
13
       // 7.6374 8 7 8
14
```

Obiekt **Math** zawiera dużą liczbę funkcji matematycznych, w szczególności funkcje zaokrąglania ...

9. Trochę praktyki #1

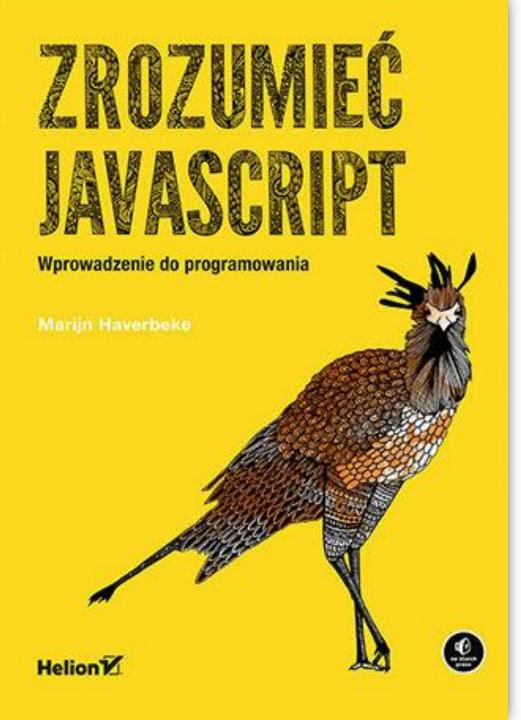
ale trochę

Zadacza: Opracuj skrypt, który obliczy kwotę podatków do zapłacenia i kwotę, która pozostanie po zapłaceniu podatków w oparciu o dochód osoby fizycznej.

10. Trochę praktyki #2

Zadacza: Opracowanie skryptu obliczającego wskaźnik masy ciała użytkownika.

Przyda się



Zrozumieć JavaScript.

Nowoczesne web-programowanie

Marjin Haverbeke

Świetna książka, zarówno jako wprowadzenie do programowania, jak i do opanowania JavaScript.

Zadanie domowe Trening!

Programowanie jest rzemiosłem i wymaga praktyki.....

- 1) Dany jest bok kwadratu. Znajdź jego obwód;
- 2) Dana jest długość krawędzi sześcianu. Znaleźć objętość sześcianu i pole jego powierzchni bocznej;
- 3) Dany jest promień okręgu. Znajdź długość okręgu i jego pole;
- 4) Podano objętość i masę substancji. Określ gęstość materiału tej substancji;
- 5) Znana jest liczba mieszkańców stanu i powierzchnia jego terytorium (w km2). Określ gęstość zaludnienia w tym stanie.
- 6). Dane są katy trójkąta prostokątnego. Znajdź jego przeciwprostokątną.
- 7) Obliczyć wartość **y** dla dowolnych wartości wejściowych **a**:



$$y = \frac{a^2 + 10}{\sqrt{a^2 + 1}}$$