



Di Mon

JavaScript: Funkcje

dimon.work/kurs.html

1. Funkcje

Funkcje w JavaScript

Funkcja to fragment kodu, który zwykle ma nazwę i może być wywołany z dowolnego miejsca w programie. **Funkcje** redukują powielanie kodu w programie, kod funkcji jest pisany raz - jest używany wielokrotnie.



Funkcje w JavaScript

Spotkaliśmy się już z funkcjami: **alert()**, **prompt()**, **parseInt()** itp. Dla nas są to tylko nazwy, nie wiemy jak są zorganizowane, ale wiemy, że przekazując do nich określone **parametry** otrzymamy oczekiwany **rezultat**.

Funkcje w JavaScript

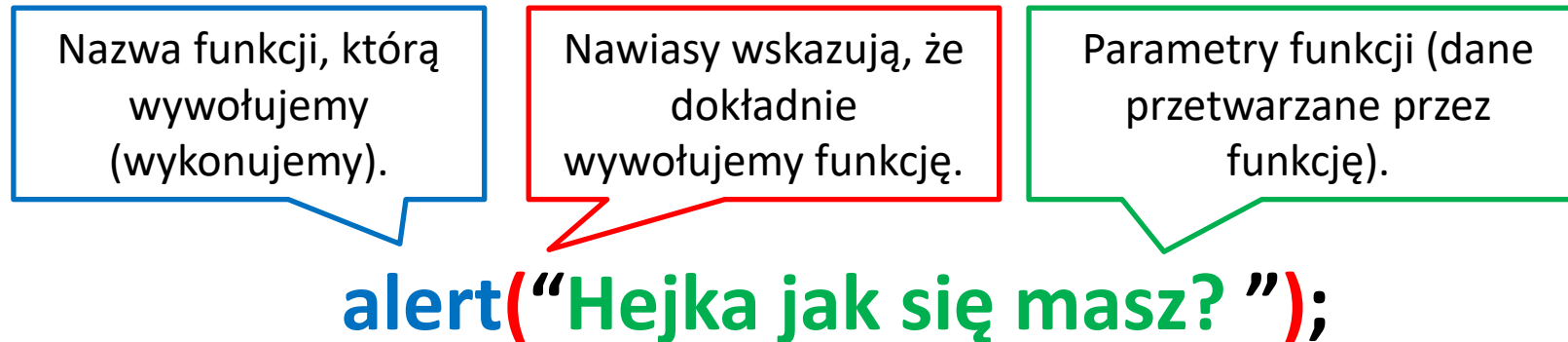
Funkcja to fragment kodu, który zwykle ma nazwę i może być wywołany z dowolnego miejsca w programie. **Funkcje** redukują powielanie kodu w programie, kod funkcji jest pisany raz - jest używany wielokrotnie

```
2  
3     function sum(a, b, c){  
4         |  
5         let result = a + b + c;  
6         |  
7         return result;  
8         |  
9     }  
10
```

Функции в JavaScript

Głównym wyróżnikiem **funkcji** są nawiasy:

np. **alert()** Nawiasy robią dwie rzeczy jednocześnie: informują przeglądarkę, że chcemy **wykonać funkcję**, której nazwa znajduje się przed nawiasami, i pozwalają nam przekazać **parametry do funkcji** (jeśli są one niezbędne do działania funkcji):



Parametry funkcji i wartość zwracana

Parametry funkcji (argumenty) – takie „zmiennie”, które są przekazywane do funkcji przy każdym wywołaniu i mogą wpływać na wynik jej pracy.

Wartość zwracana – wartość którą funkcja może zwrócić jeśli użyjemy odpowiedniego operatora **return**.

```
2  
3 → function imt(height, mass){  
4  
5     let index = mass / (height ** 2);  
6  
7     return index; ←  
8  
9     }  
10
```

Więcej informacji o parametrach funkcji

Parametry wewnątrz **funkcji** zachowują się jak zmienne tymczasowe, które działają tylko podczas **wykonywania funkcji**, po uruchomieniu przypisywane są im wartości **przekazane do funkcji** podczas jej wywołania. Nie jest konieczne przekazywanie do **funkcji** wszystkich **parametrów**, które są dostarczane przez jej nagłówek, ALE w tym przypadku ich wartość będzie **undefined**.

```
2  
3     function imt(height, mass){  
4         let index = mass / (height ** 2);  
5         return index;  
6  
7     }  
8  
9  
10
```


Praktyczne znaczenie funkcji

1. Zmniejszenie **duplikacji** kodu (powtarzania);
2. Łatwiej jest wprowadzać **zmiany**;
3. **Abstrahowanie** od szczegółów..

```
2  
3     function imt(height, mass){  
4  
5         let index = mass / (height ** 2);  
6  
7         return index;  
8  
9     }  
10
```

2. Funkcje i wydarzenia

Funkcje i wydarzenia

```
10
11 <h1 id='special' onclick="action()">Hello World!</h1>
12
13 <script>
14
15   function action(){
16     special.style.color = 'red';
17   }
18
19 </script>
20
```

Funkcje są używane w **zdarzeniowym modelu** sterowania. Ponieważ funkcja jest przede wszystkim „planem” kodu, wygodnie jest używać ich do **subskrybowania zdarzeń**. Definiujemy funkcję i „mówimy” przeglądarce, kiedy powinna zostać wywołana (np. za pomocą atrybutów znaczników: **onclick**, **onmouseenter** itp.).

https://html5css.ru/jsref/dom_obj_event.php

mousedown

mouseup

click

mousedown

mouseup

click

dblclick

3. Odwołanie do tagu

Odwołanie do tagu, praca z jego danymi

```
10
11     <h1 id='special' onclick="action()">Hello World!</h1>
12     <script>
13
14         function action(){
15             special.innerHTML = 'Some New Content...';
16         }
17
18     </script>
19
```

Najprostszym sposobem uzyskania dostępu do tagu jest jego **id**. Dla tagów z **atrybutem id** przeglądarka tworzy zmienną, poprzez którą mamy dostęp do obiektu tagu. Aby uzyskać dostęp do zawartości znacznika, możemy użyć jego właściwości **.innerHTML**. W przypadku elementów wejściowych (**input**) używana jest właściwość **.value** (w przypadku elementów wejściowych typu **checkbox** używana jest właściwość **.checked**).

4. Trochę praktyki

Kalkulator depozytów z formularzem

Dane: Istnieje kwota depozytu, roczna stopa procentowa i okres depozytu w miesiącach. Konieczne jest **obliczenie**, ile pieniędzy otrzyma deponent na koniec (kwota **depozytu** + **odsetki**). **Należy** wziąć pod uwagę, że obliczanie odsetek odbywa się w ujęciu miesięcznym (i ich kapitalizacja) bez wpływu na liczbę dni w miesiącu, a także nie uwzględnia podatków.

Formuła kapitalizacji:

$$\mathbf{Result} = \mathbf{Sum} * \left(\mathbf{1} + \frac{\mathbf{Rate}}{\mathbf{100\%}} \right)^{\mathbf{Term}}$$

*gdzie **Rate** to stopa procentowa dla okresu kapitalizacji (w naszym przypadku dla miesiąca).*

Kalkulator depozytów z formularzem

Депозитный калькулятор

Сумма вклада (грн.)

Срок вклада (мес.)

Ставка (% годовых)

Капитализация %

Рассчитать

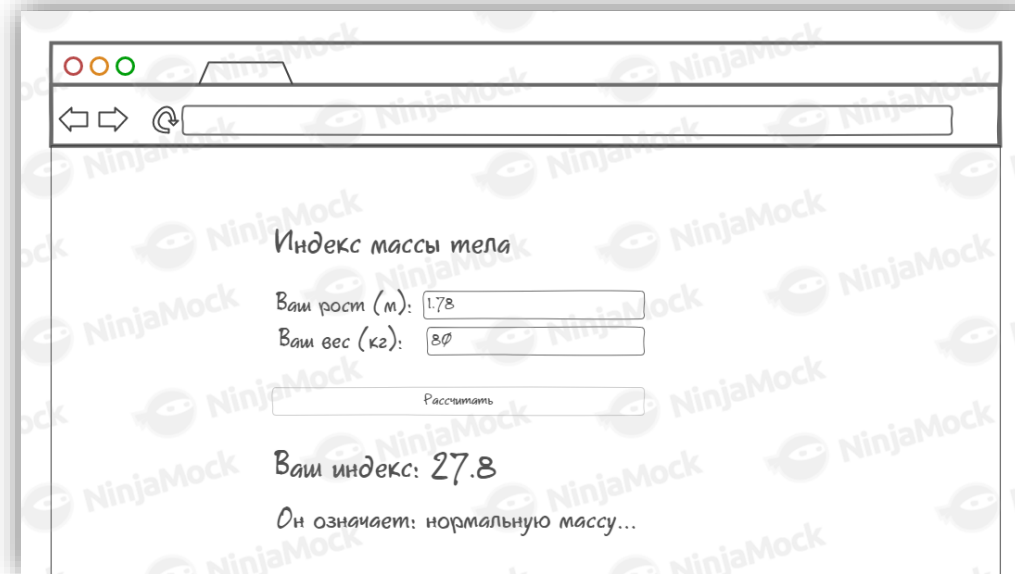
По завершению депозита вы получите: - грн.

Плиki do pobrania: dimon.work/kurs/kapital22.zip

Zadanie domowe

Praca domowa #C1

Wskaźnik masy ciała, z formularzem



Индекс массы тела

Ваш рост (м):

Ваш вес (кг):

Ваш индекс: 27.8

Он означает: нормальную массу...

Wdrożenie **obliczania** i **wyprowadzania wskaźnika masy ciała** (i diagnozy) **do znaczników** dokumentu HTML. Znaczniki powinny być przygotowane **wcześniej** (**Bootstrap** może pomóc). Należy pamiętać o możliwości pomyłkowego wprowadzenia centymetrów przez użytkownika.

JS: obiekty

```
let person = {  
  firstName: 'John',  
  lastName: 'Doe'  
};
```

