

Funkcje w JavaScript

📅 Created	
👤 Created by	
🏷️ Tags	

Wstęp

W końcu coś na co chyba wszyscy czekaliśmy, prawda? Do tej pory zagłębialiśmy się jedynie w teorię, teraz nadszedł jednak czas na część praktyczną, którą są oczywiście funkcje. Zgaduję, że może w Twojej głowie rodzić się pytanie - czym są owe funkcje i jak je się wykorzystuje. Chyba najprostszym przykładem są wyskakujące błędy w aplikacjach webowych - niepoprawne logowanie - błąd, źle wypełniony formularz - błąd. Za wyświetlanie tego typu informacji odpowiada właśnie funkcja, kiedy zostanie uruchomiona. Co najważniejsze - może być to ta sama funkcja, ale zawierające różne argumenty dostosowujące treść do sytuacji, w której została wywołana. Otrzymujemy więc potężne narzędzie, które, jeżeli nauczymy się odpowiednio wykorzystywać, zaoszczędzi Nasz czas oraz skróci pisany kod o "kilka" ładnych linijek.



W celu łatwiejszego zrozumienia co jest czym, podzielimy sobie ten artykuł na kilka etapów:

1. Deklaracja funkcji

```
function nazwaFunkcji(argumenty) {  
    // ciało funkcji (instrukcje do  
    wykonania)  
};
```

- `function` - musimy wskazać JavaScriptowi co będziemy robić, dlatego definiujemy obszar Naszych działań jako "function",
- `nazwaFunkcji` - jak sama nazwa wskazuje, będziemy tutaj umieszczać nazwę funkcji. Chodzi o to, żeby naprowadzić pozostałe osoby korzystające z kodu za co ta funkcja odpowiada, np. nazwa "getData" będzie w jakimś stopniu odpowiadać za pobieranie danych. Jak pokazuje przykład nazewnictwo funkcji w JavaScript odbywa się głównie za pomocą camelCase,
- `argumenty` - są to argumenty przekazywane do funkcji w celu wykonania na nich jakichś działań, np. dodawanie; ilość argumentów nie jest z góry zdefiniowana - może to być 0,1 lub wiele parametrów,
- `ciało funkcji` - w tym miejscu wykonywane są wszelkie działania jakie programista chce, aby zadziały się po wywołaniu funkcji,

Przykład:

```
function saySomething(text) {  
  console.log(text);  
}
```

2. Wywołanie funkcji

Zdziwiony, że powyższy kod nic nie zdziałał w konsoli? Hola, hola, powoli! To trochę tak jak z pomysłem wyjścia na piwo ze znajomym - na razie zrodził się w Twojej głowie pomysł (w naszym przypadku deklaracja), ale nic z tego nie wyjdzie, jeżeli nie powiesz swojemu znajomemu o tym pomysle (musi wyjść z Twojej głowy na zewnątrz, musisz go 'wywołać'), aby otrzymać odpowiedź 😊

W celu wywołania funkcji wystarczy podać jej nazwę oraz argumenty w nawiasach (jeśli występują):

`nazwaFunkcji(argumenty);`

Spróbujmy zatem odwołać się do naszej funkcji saySomething:

```
saySomething('Dzień dobry!');
```

I jak, jest efekt?

3. Zwracanie wartości z funkcji

Każda funkcja bezwzględnie zwróci **undefined**:

```
function saySomething(text) {  
  console.log(text)  
}  
  
let res = saySomething('Dzień dobry!');  
console.log ('Result of res: ', res);  
  
// w odpowiedzi otrzymamy:  
  
"Dzień dobry!"  
"Result of res: ", undefined
```

Chyba że jednoznacznie określimy jakie wyrażenie lub wartość funkcja ma zwrócić za pomocą **return**:

```
function multiply(x,y) {  
  return x*y;  
}  
  
let res = multiply(2,7);  
console.log ('Result of res: ', res);  
  
// w odpowiedzi zostanie zwrócone:  
  
"Result of res: ", 14
```

4. Użycie funkcji przed zadeklarowaniem (ang. hoisting)

JavaScript pozwala 'podciągnąć' deklarację do wyżej użytego wywołania. Dzieje się tak dlatego, że silnik tego języka ustawia wszystkie deklaracje domyślnie na początku kodu zanim zacznie je wykonywać.

```
hoistMeUp();  
  
function hoistMeUp() {  
  console.log('To podciąganie naprawdę działa!')  
}
```

WAŻNA INFORMACJA DO ZADAŃ!!!

!! Do każdego artykułu utwórzcie sobie na komputerze folder i w nim pliki z rozwiązaniami zadań. Zawsze, nawet jeśli pytanie jest otwarte i czysto teoretyczne to warto napisać na nie odpowiedź by nasz mózg lepiej zapamiętał to co piszemy.

Legenda:

bez gwiazdek - zadania podstawowe z artykułów oraz filmu,

* - na podstawie artykułu, ale nie znajdziesz tego na filmie,

** - nie ma tego w artykule ani w filmiku ale dostajesz link gdzie znajduje się rozwiązanie,

*** - brak informacji w artykule, filmie oraz brak linku do rozwiązania. Są to najbardziej rozwijające zadania, które również uczą samodzielności :)

1. Napisz funkcję o nazwie **sum**, która będzie przyjmować dwa argumenty i zwracać ich sumę. Wywołaj funkcję i sprawdź czy działa poprawnie.
2. ** Napisz funkcję, która jako argument przyjmować będzie Twoje imię. Następnie w ciele funkcji wywołaj **alert**, który wyświetli tekst 'Witaj, Andrzej' (jeżeli masz na imię Andrzej :))
<https://developer.mozilla.org/pl/docs/Web/API/Window/alert>
3. Napisz funkcję, która jako argument przyjmować będzie wiek, natomiast powinna zwracać wiek przeliczony na dni. Przyjmijmy, że rok ma 365dni i nie zwracamy uwagi na lata przestępne. Postaraj zabezpieczyć ciało funkcji, aby przyjmowało jedynie parametry dodatnie. Jeżeli jednak ktoś wprowadzi liczbę nie spełniającą warunku, to powinien otrzymać **alert** wskazujący na to, co jest nie tak.
4. Napisz funkcję, która będzie przyjmować dwa parametry będące liczbami całkowitymi. Powinna zwracać wartość true, jeżeli jeden z argumentów jest liczbą 12, lub suma obu argumentów wynosi 12.
5. ** Zadeklaruj tablicę z czterema imionami. Następnie napisz funkcję, która jako argument będzie przyjmować tę tablicę oraz drugi argument - literę. Następnie sprawdź czy dana litera występuje w każdym z elementów tablicy i zwróć informację na ten temat w konsoli. Sprawdź czy na wynik ma wpływ wielkość szukanej litery. Zapoznaj się z dokumentacją:
https://developer.mozilla.org/pl/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase
6. Napisz funkcję, która przyjmuje jako parametr tablicę i nowy element do tablicy. Funkcja ta powinna przeiterować tablicę sprawdzając czy już taki element w niej jest. W przypadku jeśli go nie ma, to powinna dodać go do tablicy, a jeżeli już tam się znajduje, to wyświetlić odpowiedni komunikat.
7. Stwórz funkcję, która przyjmuje tablicę z numberami i z tego tworzy dwie nowe tablice z liczbami parzystymi i nieparzystymi.

8. Stwórz funkcję, która określi czy barman powinien zaserwować napój alkoholowy osobie, która jest nieletnia. Na początek zdefiniuj sobie tablicę z obiektami:

```
const people = [  
  {  
    name: 'Andrzej',  
    age: 19,  
  },  
  {  
    name: 'Jasiek',  
    age: 16,  
  }  
]
```

Nasza funkcja przyjmie dwa parametry - pierwszym z nich będzie oczywiście powyższa tablica, kolejnym zaś boolean określający czy barman nie jest czasami na przerwie. Jeżeli barman jest na przerwie, to wyskakuje w konsoli powiadomienie - "Bar nieczynny, barman ma przerwę". Jeżeli jednak jest przy barze, to należy sprawdzić, czy Nasi dzielni chłopcy są pełnoletni za pomocą pętli i logiki. W przypadku, jeżeli chłopiec jest pełnoletni, powinna się wyświetlić informacja : "Spokojnie (imię chłopaka), barman zaraz Cię obsłuży". Jeżeli jednak ta druga osoba to małolat, to powinniśmy otrzymać informację "(imię osoby), uciekaj do domu, bo matce powiem!".