

Funkcje w JavaScript

Funkcja w języku JavaScript to jedno lub więcej poleceń zgrupowanych w całość przy pomocy nawiasów klamrowych `{ }`. Tak zdefiniowana funkcja może zawierać listę argumentów umieszczonych w nawiasach okrągłych `()` oddzielonych przecinkami — oraz może zwracać wartość.

Definiowanie funkcji

W języku JavaScript można definiować własne funkcje. Definicja funkcji musi zawierać słowo kluczowe `function`, po którym następuje nazwa funkcji, następnie w nawiasach okrągłych powinny zostać wymienione parametry funkcji oddzielone przecinkami. W nawiasach klamrowych jest zapisywana treść funkcji, na której końcu powinna zostać wstawiona instrukcja `return`, określająca zwracaną przez funkcję wartość. Ponieważ funkcja zawsze zwraca jakąś wartość, to jeżeli nie zostanie to podane w jawny sposób z wykorzystaniem instrukcji `return`, zwracana jest automatycznie wartość `undefined`.

Instrukcja `return` powoduje przerwanie wykonywania poleceń funkcji i powrót do miejsca, z którego funkcja została wywołana. Definicja funkcji ma postać:

```
function nazwa_funkcji (argumenty_funkcji) {instrukcje}
```

Można definiować funkcje bezparametrowe:

```
function nazwa_funkcji () {instrukcje}
```

Przykład:

```
function suma(a,b)
{
  var c=a+b;
  return c;
}
```

Wywołanie funkcji

Wywołanie funkcji nastąpi po podaniu w kodzie skryptu jej nazwy wraz z parametrami umieszczonymi w nawiasach.

Przykład

```
<script type="text/javascript">
<!--
function trzy(z) {
for(var i=0; i<=z; i++){
if((i%3!=0) continue;
document.write(i+"");
}
document.write("<br>");
}
trzy(90);
trzy(120);
//-->
</script>
```

W podanym przykładzie została zdefiniowana funkcja `trzy(z)`, która wyświetla ciąg liczb podzielnych przez 3. Argument `z` określa zakres wyświetlania liczb. Funkcja `trzy(z)` została wywołana dwukrotnie, raz z argumentem `z=90`, drugi raz z argumentem `z=120`.

Argumenty

Przy wywołaniu funkcji należy podać wartości jej argumentów. Jeżeli zostaną one pominięte, JavaScript przypisze argumentom wartość `undefined`. Jeżeli otrzyma ich więcej, niż jest wymagane, zignoruje niepotrzebne argumenty. Możliwe jest tworzenie funkcji, które będą miały zmienną liczbę argumentów. Wykorzystywana jest do tego tablica `arguments`, która automatycznie jest tworzona dla każdej funkcji.

```
function lista_arg()
{
return arguments;
}
```

Wywołanie funkcji bez podania parametrów nie zwróci żadnej wartości.

```
lista_arg();
```

Wynik:

```
[ ]
```

Wywołanie funkcji z parametrami zwróci listę wartości.


```
lista_arg(1,4,5,9,"A","koło");
```

Wynik:

```
[1,4,5,9,"A","koło"]
```

Wykorzystując tablicę `arguments`, można zdefiniować funkcję, która będzie sumowała dowolną liczbę parametrów.

```
<script type="text/javascript">
<!--
function suma_dow()
{
var i, wynik = 0;
var l_param = arguments.length;
for (i = 0; i < l_param; i++)
{
    wynik += arguments[i];
}
return wynik;
}
// -->
</script>
```



Użyta w definicji funkcji właściwość `arguments.length` zwróci liczbę parametrów podanych podczas wywołania funkcji.

Wywołanie funkcji z różną liczbą parametrów powinno dać zawsze poprawny wynik, np.

```
suma_dow(3,5,7);
```

da wynik 15,

```
suma_dow(1,2,3,4,5,6,7,8,9);
```

da wynik 45.