

Раздел

[Основы JavaScript](#)

Навигация по уроку

Многострочные стрелочные функции

Итого

Задачи (1)

Комментарии

Поделиться

[Редактировать на GitHub](#)[🏠 → Язык программирования JavaScript](#)
[→ Основы JavaScript](#) 1-го октября 2020

Функции-стрелки, основы

Существует ещё более простой и краткий синтаксис для создания функций, который часто лучше, чем синтаксис Function Expression.

Он называется «функции-стрелки» или «стрелочные функции» (arrow functions), т.к. выглядит следующим образом:

```
1 let func = (arg1, arg2, ...argN) => expression
```

...Такой код создаёт функцию `func` с аргументами `arg1..argN` и вычисляет `expression` с правой стороны с их использованием, возвращая результат.

Другими словами, это более короткий вариант такой записи:

```
1 let func = function(arg1, arg2, ...argN) {  
2   return expression;  
3 };
```

Давайте взглянем на конкретный пример:

```
1 let sum = (a, b) => a + b;  
2  
3 /* Более короткая форма для:  
4  
5 let sum = function(a, b) {  
6   return a + b;  
7 };  
8 */  
9  
10 alert( sum(1, 2) ); // 3
```

То есть, `(a, b) => a + b` задаёт функцию с двумя аргументами `a` и `b`, которая при запуске вычисляет выражение справа `a + b` и возвращает его результат.

- Если у нас только один аргумент, то круглые скобки вокруг параметров можно опустить, сделав запись ещё короче:

```
1 // тоже что и  
2 // let double = function(n) { return n * 2 }  
3 let double = n => n * 2;  
4  
5 alert( double(3) ); // 6
```

- Если нет аргументов, указываются пустые круглые скобки:

```
1 let sayHi = () => alert("Hello!");  
2  
3 sayHi();
```

Функции-стрелки могут быть использованы так же, как и Function Expression.

Например, для динамического создания функции:

Раздел

Основы JavaScript

Навигация по уроку

Многострочные стрелочные функции

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



```
1 let age = prompt("Сколько Вам лет?", 18);
2
3 let welcome = (age < 18) ?
4   () => alert('Привет') :
5   () => alert("Здравствуй!");
6
7 welcome(); // теперь всё в порядке
```

Поначалу функции-стрелки могут показаться необычными и трудночитаемыми, но это быстро пройдёт, как только глаза привыкнут к этим конструкциям.

Они очень удобны для простых однострочных действий, когда лень писать много букв.

Многострочные стрелочные функции

В примерах выше аргументы использовались слева от `=>`, а справа вычислялось выражение с их значениями.

Порой нам нужно что-то посложнее, например, выполнить несколько инструкций. Это также возможно, нужно лишь заключить инструкции в фигурные скобки. И использовать `return` внутри них, как в обычной функции.

Например:

```
1 let sum = (a, b) => { // фигурная скобка, открывающая
2   let result = a + b;
3   return result; // при фигурных скобках для возврата з
4 };
5
6 alert( sum(1, 2) ); // 3
```

Дальше будет ещё информация

Здесь мы рассмотрели функции-стрелки как способ писать меньше букв. Но это далеко не всё!

Стрелочные функции обладают другими интересными особенностями. Их изучение требует знания некоторых других возможностей языка JavaScript, поэтому мы вернёмся к стрелочным функциям позже, в главе [Повторяем стрелочные функции](#).

А пока мы можем использовать их для простых однострочных действий и колбэков.

Итого

Функции-стрелки очень удобны для однострочных действий. Они бывают двух типов:

1. Без фигурных скобок: `(...args) => expression` – правая сторона выражение: функция выполняет его и возвращает результат.
2. С фигурными скобками: `(...args) => { body }` – скобки позволяют нам писать многострочные инструкции внутри функции, но при этом необходимо указывать директиву `return`, чтобы вернуть какое-либо значение.

Задачи

Перепишите с использованием функции-стрелки

Замените код Function Expression стрелочной функцией:

```
1 function ask(question, yes, no) {
2   if (confirm(question)) yes()
```

Раздел

[Основы JavaScript](#)

Навигация по уроку

Многострочные стрелочные функции

Итого

Задачи (1)

Комментарии

Поделиться



Редактировать на GitHub



```
3   else no();
4 }
5
6 ask(
7   "Вы согласны?",
8   function() { alert("Вы согласились."); },
9   function() { alert("Вы отменили выполнение."); }
10 );
```

решение

Проводим [курсы по JavaScript и фреймворкам](#). ✕



Комментарии

перед тем как писать...

