setTimeout и setInterval

- Что это за функции?
- Чем отличаются?
- Применение на практике

setTimeout

Функция setTimeout() вызывает функцию или выполняет переданный код после указанной задержки.

!!! вызывается на исполнение только один раз

Синтаксис setTimeout

var timerId = setTimeout(func / code, delay[, arg1, arg2...])

func: Имя функции, которая будет вызвана после указанной задержки.

code: Строка, содержащая JavaScript-код, который будет выполнен после указанной задержки. Строка поддерживается для совместимости, использовать её не рекомендуется.

delay Задержка в миллисекундах. 1000 миллисекунд равны 1 секунде.

arg1, arg2... Аргументы, которые нужно передать функции.
!!!Не поддерживаются в IE9-.

Пример функции setTimeout

```
setTimeout("alert('Привет')", 1000);

setTimeout(function() { alert('Привет') }, 1000);
```

setInterval

Функция setInterval() неоднократно вызывает функцию или выполняет переданный код с указанной временной задержкой между каждым вызовом.

Синтаксис setInterval

var timerId = setInterval(func / code, delay[, arg1, arg2...])

func: Имя функции, которая будет вызываться после каждой задержки.

code: Строка, содержащая JavaScript-код, который будет выполняться после каждой задержки.

delay: Задержка, указываемая в миллисекундах, по истечении которой каждый раз будет выполняться вызов функции. !!!Если указано значение меньше 10, то используется значение 10.

arg1, arg2... : Параметры, которые будут переданы в качестве аргументов указанной функции.

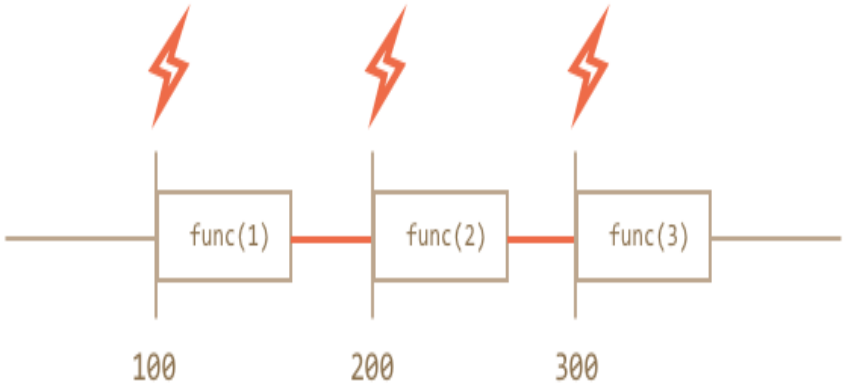
Пример функции setInterval

```
setInterval(function() { alert('Привет тебе!') }, 3000);
```

Рекурсивный setTimeout

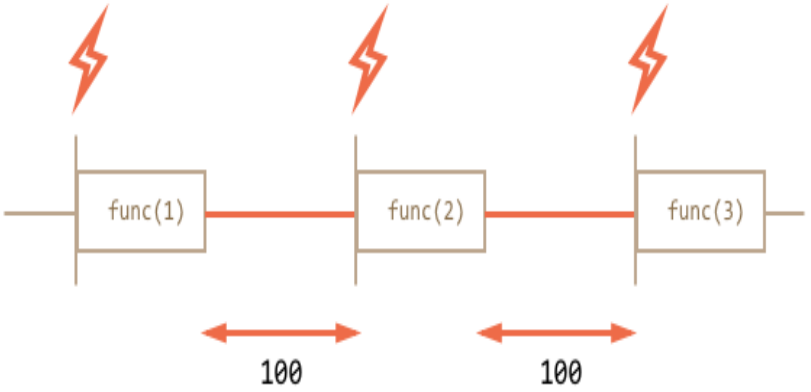
Рекурсивный setTimeout – более гибкий метод тайминга, чем setInterval, так как время до следующего выполнения можно запланировать по-разному, в зависимости от результатов текущего.

Рекурсивный setTimeout и setInterval



setInterval

```
var i = 1; setInterval(function() {
func(i); }, 100);
```



Рекурсивный setTimeout

```
var i = 1; setTimeout(function run() {
func(i); setTimeout(run, 100); }, 100);
```

clearInterval и clearTimeout

Оба метода возвращают идентификатор таймера.
Его используют для остановки выполнения вызовом clearInterval/clearTimeout.

```
IntervalID = setInterval(...)
...
clearInterval (IntervalID)
```

```
timeoutID = setTimeout(...)
...
clearTimeout(timeoutID)
```

Перейдем к практике!