Скрипт 60 секунд

Скрипт работает по системе постоянной отправки целей каждые 30 секунд в Яндекс Метрику с уникальным идентификатором. При этом вся активность на сайте запоминается в КУКАХ браузера пользователя.

Это значит, что все посещения сохраняются и при следующем посещении сайта пользователем его активность продолжит идти в плюс без разрыва и повторно отправленных целей 60 секунд от одного пользователя.

Инструкция

Копируем скрипт целиком ниже. В IDmetrika подставьте свой номер счетчика.

```
<script>
var settings = {
   /* Настройки которые надо менять */
   need: 30, // Сюда вписываем через какое количество секунд повторно отправлять цель, по умолчанию стоит 60, но можно указать и 30
   checkTime: 10, // секунды. период проверки.
   //желательно, чтобы need было кратно checkTime
        IDmetrika: 123456789, // Сюда вписываем ИД Счетчика Яндекс Метрики
        /* Настройки которые надо менять */
}
/*
## Инструкция по добавлению целей в Яндекс.Метрику ##
Если вы указываете в функции NEED шаг в 30 секунд, то в Метрике вы можете отслеживать цели с шагом в 30 секунд и цели событий добавляйте такие:
30sec
60sec
90sec
```

120sec

150sec и т.д.

Если вы указываете в функции NEED шаг в 60 секунд, то в Метрике вы можете отслеживать цели с шагом в 30 секунд и цели событий добавляйте такие:

60sec

120sec

180sec

240sec и т.д.

*/

/* БОЛЬШЕ В КОДЕ НИЧЕГО НЕ ТРОГАЕМ, ЦЕЛИ ТОЖЕ НЕ ПРАВИМ */

```
var metricsFn = function () {
    console.log(ActiveScore.timer);
    console.log(ActiveScore.need);
    var c1 = this.getCookie(this.cookieName);
    console.log(c1);
    if (ActiveScore.timer >= ActiveScore.need) {
        console.log("событие отправилось");
        /* Тут перечислять все что нужно будет вызвать по достижению цели */
        ym(settings.IDmetrika, "reachGoal", this.cookieName.slice(0, -3));

        /* Тут перечислять все что нужно будет вызвать по достижению цели */
    }
};

var ActiveScore = {
    need: settings.need,
    checkTime: settings.checkTime,
```

```
loop: true,
counter: 0,
cookieName: "60sec_ap",
sendFn: null,
parts: 0,
active_parts: 0,
timer: 0,
events: [
    "touchmove",
    "blur",
    "focus",
    "focusin",
    "focusout",
    "load",
    "resize",
    "scroll",
    "unload",
    "click",
    "dblclick",
    "mousedown",
    "mouseup",
    "mousemove",
    "mouseover",
    "mouseout",
    "mouseenter",
    "mouseleave",
    "change",
    "select",
    "submit",
    "keydown",
```


```
    "keypress",
    "keyup",
    "error",
],

setEvents: function () {
    for (var index = 0; index < this.events.length; index++) {
        var eName = this.events[index];
        window.addEventListener(eName, function (e) {
            if (e.isTrusted && ActiveScore.period.events == false) {
                ActiveScore.period.events = true;
            }
        });
    }
},

period: {
    start: 0,
    end: 0,
    events: false,
},

init: function (fn) {
    this.calcParts();
    this.setEvents();
    this.setStartCounter();
    if (this.checkCookie()) {
        this.sendFn = fn;
        this.start();
    }
```


```
},

readLastCookie: function () {
    var absurdlyLarge = 100000;
    for (var i = 1; i < absurdlyLarge; i++) {
        var cookie = this.getCookie(i * this.need + 'sec_ap');
        if (cookie != this.parts * this.parts) return { i: i, cookie: cookie };
    }
    return { i: 1, cookie: 0 };
},

setStartCounter: function () {
    var lastCookie = this.readLastCookie();
    this.counter = lastCookie.i - 1;
    this.active_parts = Number(lastCookie.cookie);
    this.cookieName = (this.counter + 1) * this.need + "sec_ap";
},

calcParts: function () {
    this.parts = Math.ceil(this.need / this.checkTime);
},

setPeriod: function () {
    this.period.start = this.microtime();
    this.period.end = this.period.start + this.checkTime;
    this.period.events = false;
},

microtime: function () {
    var now = new Date().getTime() / 1000;
```


```
    var s = parseInt(now);
    return s;
},

start: function () {
    this.setPeriod();
    this.runPeriod();
},

timeoutId: null,

checkPeriod: function () {
    if (this.period.events == true) {
        this.active_parts = this.active_parts + 1;
        // console.log('В этой секции были действия');
    } else {
        // console.log('В этой секции НЕБЫЛО действия');
    }
    this.timer = this.active_parts * this.checkTime;
    console.log(
        this.active_parts + " / " + this.parts + " [" + this.timer + "]"
    );

    if (this.checkSecs()) {
    } else {
        this.start();
    }
    this.setCookie(this.cookieName, this.active_parts);
},
```


```
checkSecs: function () {
    if (this.timer >= this.need) {
        this.send();
        if (this.loop == true) {
            this.counter++;
            this.timer = 0;
            this.active_parts = 0;
            this.cookieName = (this.counter + 1) * this.need + "sec_ap";
            return false;
        } else {
            // console.log('Завершили проверку активности');
            return true;
        }
    }
    return false;
},

timeoutFn: function () {
    ActiveScore.checkPeriod();
},

runPeriod: function () {
    this.timeoutId = setTimeout(this.timeoutFn, this.checkTime * 1000);
},

send: function () {
    if (this.getCookie(this.cookieName) == this.parts * this.parts) {
        this.setStartCounter();
    } else {
        this.setCookie(this.cookieName, this.active_parts * this.active_parts);
```


```
    }
    this.sendFn();
},

checkCookie: function () {
    var c = this.getCookie(this.cookieName);
    if (c == null) {
        return true;
    } else {
        if (c == '') return true;
        c = parseInt(c);
        if (c >= this.parts) {
            // console.log('Скрипт даже не запустился...');
            if (this.loop == true) {
                return true;
            }
            return false;
        } else {
            this.active_parts = c;
            return true;
        }
    }
},

setCookie: function (name, value, days) {
    var expires = "";
    if (days) {
        var date = new Date();
        date.setTime(date.getTime() + days * 24 * 60 * 60 * 1000);
        expires = "; expires=" + date.toUTCString();
```


```
        }
        document.cookie = name + "=" + (value || "") + expires + "; path=/";
    },
    getCookie: function (name) {
        var nameEQ = name + "=";
        var ca = document.cookie.split(";");
        for (var i = 0; i < ca.length; i++) {
            var c = ca[i];
            while (c.charAt(0) == " ") c = c.substring(1, c.length);
            if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
        }
        return null;
    },
    eraseCookie: function (name) {
        document.cookie =
            name + "=; Path=/; Expires=Thu, 01 Jan 1970 00:00:01 GMT;";
    },
};

ActiveScore.init(metricsFn);
</script>
```