



Подключение

1. Перед началом установки, требуется разместить папку `moonplayer_ce` в корне вашего сайта.

(... если нужен только пример, то он находится ниже, под заголовком «Пример страницы с плеером»)

2. Для работы плеера, внутри тега `<head>` должен находиться следующий список скриптов:

```
<link rel="stylesheet" href="./moonplayer_ce/css/style.css">
<link rel="stylesheet" href="./moonplayer_ce/css/style.php">
<script src="./moonplayer_ce/js/clappr.min.js"></script>
<script src="./moonplayer_ce/js/main.js"></script>
```

Создаем блок плеера

3. В теге `<body>` создаем блок `<div>` с уникальным идентификатором `myvideo`. Идентификатор может принимать любое уникальное значение.

```
<div id="myvideo"></div>
```

4. Что бы сразу задать размеры плеера, в параметре `style` того же блока `<div>` укажем ширину (`width`) и высоту (`height`) как на примере ниже:

```
<div id="myvideo" style="width:640px; height:360px;"></div>
```

Конфигурация плеера

5. Перед закрывающим тегом `</body>` разместим тег `<script>`, где создадим переменную `myvideo` (`var myvideo`). В эту переменную запишем экземпляр нашего, пока не сконфигурированного плеера (`new MPCE`), а в скобках укажем тот самый идентификатор блока `<div>` – `myvideo`.

```
<script>
  var myvideo = new MPCE('myvideo');
</script>
```

Выше был создан экземпляр класса `MPCE` и записан в переменную `myvideo`. Так же, тут стоит отметить, что вторым параметром мы могли бы указать адрес постера, как на примере ниже:

```
<script>
  var myvideo = new MPCE('myvideo', './moonplayer_ce/assets/poster.jpg');
</script>
```

6. У экземпляра есть два метода `add` и `init`. Метод `add` добавляет к плееру видео-файл и мета-данные. Добавим новые строки в том же блоке `<script>`:

```
<script>
  var myvideo = new MPCE('myvideo');

  myvideo.add({
    link: './moonplayer_ce/assets/source0.mp4',
    season: 'Сезон 1',
    episode: 'Эпизод 1',
    release: 'Lostfilm',
    bitrate: '320p'
  });
</script>
```

Как можно догадаться, используя метод `add` мы передали плееру `myvideo` ссылку на видео и мета-данные, а именно сезон (`season`), эпизод (`episode`), перевод (`release`) и битрейт (`bitrate`).

7. Используя второй метод `init` укажем плееру, что передача данных и конфигурация завершена.

```
<script>
  var myvideo = new MPCE('myvideo');

  myvideo.add({
    link: './moonplayer_ce/assets/source0.mp4',
    season: 'Сезон 1',
    episode: 'Эпизод 1',
    release: 'Lostfilm',
    bitrate: '320p'
  });

  myvideo.init();
</script>
```

Теперь, если запустить страничку мы сможем увидеть плеер. Но в нем не будет сезона, серии, перевода и битрейта, так как плеер сконфигурирован всего лишь на один видео-ролик, а выбирать из одного сезона/серии или другой сущности не имеет смысла.

8. Что бы пользователю было из чего выбирать, нужно добавить вариации данных. Добавим еще один эпизод. А так же уберем все то, что находится в единственном экземпляре.

```
<script>
  var myvideo = new MPCE('myvideo');

  myvideo.add({
    link: './moonplayer_ce/assets/source0.mp4',
    episode: 'Эпизод 1'
  });

  myvideo.add({
    link: './moonplayer_ce/assets/source1.mp4',
    episode: 'Эпизод 2'
  });

  myvideo.init();
</script>
```

Те мета-данные, которые не имеют вариаций плееру можно не сообщать, так как использоваться они не будут. Следуя этому принципу, в примере выше, мы убрали сезон, перевод и битрейт.

Пример страницы с плеером

```
<!DOCTYPE html>

<html>
<head>
  <title>Page Title</title>
  <meta charset="utf-8">

  <!-- подключаем Moonplayer CE -->
  <link rel="stylesheet" href="./moonplayer_ce/css/style.css">
  <link rel="stylesheet" href="./moonplayer_ce/css/style.php">
  <script src="./moonplayer_ce/js/clappr.min.js"></script>
  <script src="./moonplayer_ce/js/main.js"></script>
</head>

<body>

  <div id="player" style="width:640px; height:360px;"></div>

  <script>
    // создаем экземпляр плеера
    var player = new MPCE('player');

    // первый эпизод
    player.add({ link: './moonplayer_ce/assets/source0.mp4',
      season: 'Сезон 1',
      episode: 'Эпизод 1',
      release: 'Lostfilm',
      bitrate: '320p'
    });
    // тот же эпизод и озвучка, но уже другой битрейт
    player.add({ link: './moonplayer_ce/assets/source1.mp4',
      season: 'Сезон 1',
      episode: 'Эпизод 1',
      release: 'Lostfilm',
      bitrate: '480p'
    });
    // второй эпизод
    player.add({ link: './moonplayer_ce/assets/source2.mp4',
      season: 'Сезон 1',
      episode: 'Эпизод 2',
      release: 'Lostfilm'
    });
    // второй эпизод, только другая озвучка
    player.add({ link: './moonplayer_ce/assets/source3.mp4',
      season: 'Сезон 1',
      episode: 'Эпизод 2',
      release: 'Coldfilm'
    });
    // единственное видео второго сезона
    player.add({ link: './moonplayer_ce/assets/source4.mp4',
      season: 'Сезон 2'
    });

    // применяем конфигурацию
    player.init();
  </script>
</body>
</html>
```

Настройка цветовых схем плеера

В папке `moonplayer_ce` находится файл `style.ini`, который содержит параметры со значениями цветов в формате `HEX/RGB/RGVA/HSI/HSLA`. Данные параметры устанавливают цвета для каждого элемента плеера, что позволяет подстроить плеер под дизайн сайта.

Позиции баннера

Установить `Popup`-баннер можно в три позиции:

1. Преролл (on: 'play')
2. На паузу (on: 'pause')
3. Оверролл (on: 'end')

Ниже пример кода подключения баннера в момент конфигурации плеера. Размещение необходимо, между созданием экземпляра `MPCE` и вызовом метода `init()`.

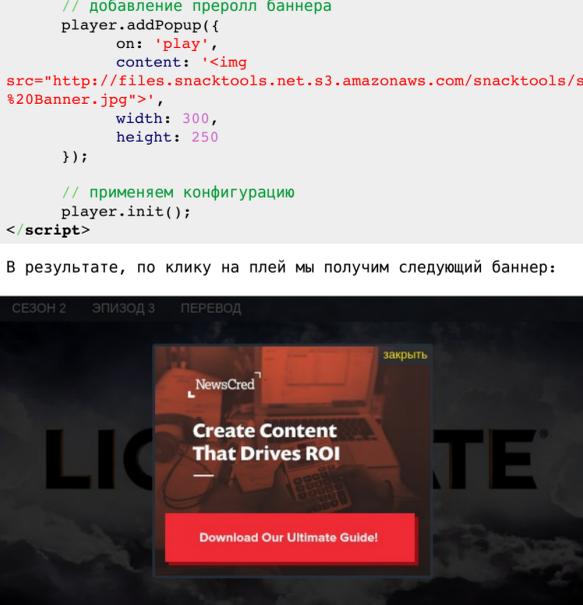
```
<script>
  // создание экземпляра MPCE
  var player = new MPCE();

  // ... тут добавление видео ...

  // добавление преролл баннера
  player.addPopup({
    on: 'play',
    content: '',
    width: 300,
    height: 250
  });

  // применяем конфигурацию
  player.init();
</script>
```

В результате, по клику на плей мы получим следующий баннер:



Настройка баннера

В предыдущем примере, баннер был подключен используя только обязательные параметры, такие как `on`, `content`, `width`, `height`. Без этих параметров баннер создан не будет. Ниже расположен полный список параметров, которые можно настроить:

Параметры метода `addPopup`:

***on** (строка) – позиция баннера. Является обязательным параметром и принимает три значения: 'play', 'pause', 'end'.

delay (число) – если параметр установлен, закрыть баннер можно только по истечению указанного количества секунд с момента его появления.

delayText (строка) – если установлен параметр `delay`, то в параметре `delayText` можно указать сообщение с счетчиком, которое выводится перед показом закрывающей кнопки. Значение по умолчанию 'time% сек' пример: 'Через time% сек', где time% будет содержать секунды, меняющиеся в реальном времени.

close (строка) – содержимое закрывающей кнопки. Если параметр не указан, то 'закрыть' будет являться значением по умолчанию. Что бы указать крестик, можно использовать его html код: '✖'

message (строка) – если присутствует данный параметр, под баннером появится панель с указанным в значении содержимым.

***content** (строка) – содержимое баннера. Параметр может принимать значение в виде строки, а так же идентификатор элемента '#id', содержимое которого будет использовано в `Popup`-баннере. Значение нужно указывать в одну строку.

***width** (число) – ширина баннера в пикселях.

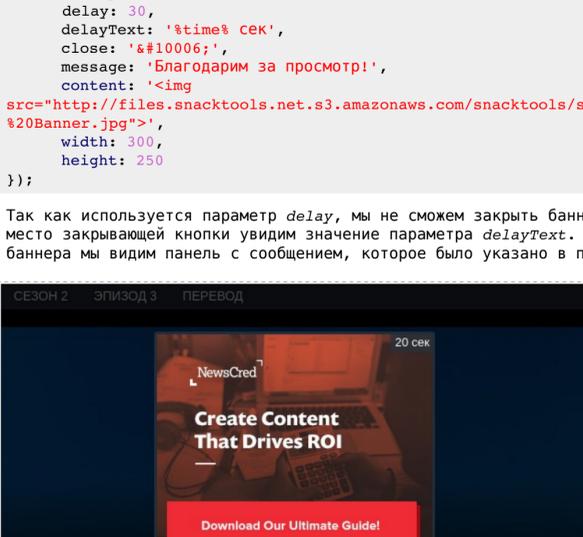
***height** (число) – высота баннера в пикселях.

* Обязательные параметры выделены звездочкой.

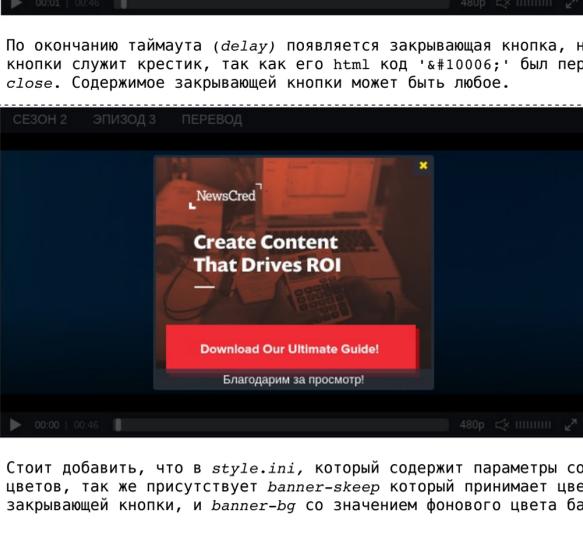
Пример с указанием всех параметров:

```
player.addPopup({
  on: 'pause',
  delay: 30,
  delayText: 'time% СЕК',
  close: '&#10006;',
  message: 'Благодарим за просмотр!',
  content: '',
  width: 300,
  height: 250
});
```

Так как используется параметр `delay`, мы не сможем закрыть баннер сразу, а в место закрывающей кнопки увидим значение параметра `delayText`. Так же, внизу баннера мы видим панель с сообщением, которое было указано в параметре `message`.



По окончании таймута (`delay`) появляется закрывающая кнопка, но содержимое кнопки служит крестик, так как его html код '✖'; был передан в параметр `close`. Содержимое закрывающей кнопки может быть любое.



Стоит добавить, что в `style.ini`, который содержит параметры со значениями цветов, так же присутствует `banner-skeer` который принимает цвет текста активной закрывающей кнопки, и `banner-bg` со значением фонового цвета баннера.